

# **TÍMOVÝ PROJEKT**

## **TraSpi**

Dokumentácia k inžinierskemu dielu

Bc. Hajdu Daniela  
Bc. Marták Lukáš  
Bc. Mäsiar Aleš  
Bc. Miškovský Lukáš  
Bc. Moravčíková Zora  
Bc. Šandor Filip

**Tím č. 20:** TraSpi  
**Vedúci projektu:** Ing. Eduard Kuric  
**Predmet:** Tímový projekt 1  
**Ročník:** 2015/2016  
**Mailový kontakt:** [traspi-team@googlegroups.com](mailto:traspi-team@googlegroups.com)

# Obsah

---

|       |                                       |    |
|-------|---------------------------------------|----|
| 1     | Big picture .....                     | 1  |
| 1.1   | Úvod .....                            | 1  |
| 1.2   | Globálne ciele zimného semestra ..... | 1  |
| 1.3   | Globálne ciele letného semestra ..... | 2  |
| 1.4   | Celkový pohľad na systém .....        | 3  |
| 2     | Moduly systému .....                  | 4  |
| 2.1   | Analýza .....                         | 4  |
| 2.2   | Návrh .....                           | 4  |
| 2.2.1 | Scenár .....                          | 5  |
| 2.2.2 | Grafika a interakcia .....            | 5  |
| 2.2.3 | Model hry .....                       | 6  |
| 2.2.4 | Model scény .....                     | 7  |
| 2.2.5 | Model interakcií .....                | 8  |
| 2.3   | Stavový priestor hry .....            | 9  |
| 2.3.1 | Pojmy .....                           | 9  |
| 2.3.2 | Objekty .....                         | 9  |
| 2.3.3 | Tvorba stavového priestoru .....      | 10 |
| 2.3.4 | Vytvorené diagramy .....              | 14 |
| 2.4   | Reprezentácia interakcií .....        | 17 |
| 2.5   | Reprezentácia dialógov .....          | 22 |
| 2.6   | Reprezentácia konverzácií .....       | 24 |
| 2.7   | Implementácia .....                   | 27 |
| 2.7.1 | Pohyb po 2D scéne .....               | 28 |
| 2.7.2 | Animácia postavy .....                | 29 |
| 2.7.3 | Interakcia s objektami .....          | 29 |
| 2.7.4 | Tvorba grafiky .....                  | 30 |
| 2.7.5 | Výpis na obrazovku .....              | 41 |
| 2.7.1 | Konzola pre ladenie ASR modulu .....  | 42 |
| 2.7.2 | Inventár .....                        | 43 |
| 2.7.3 | Obchod .....                          | 44 |
| 2.7.4 | Ukončenie hry .....                   | 44 |
| 2.7.5 | Animácia dverí .....                  | 44 |
| 2.7.6 | Hlasové moduly SAV .....              | 45 |

|        |  |    |
|--------|--|----|
| 2.7.7  | Čítanie prísahy pre zistenie základných parametrov reči..... | 47 |
| 2.7.8  | Správanie vtipkárky.....                                     | 47 |
| 2.7.9  | ChatBot – Dialógový manažér .....                            | 48 |
| 2.7.10 | MiniHra – Flappy Harry.....                                  | 50 |
| 2.7.11 | Logovanie.....   | 52 |
| 2.7.12 | Nastavenia .....   | 53 |
| 2.8    | Testovanie.....  | 54 |
| 3      | Príručky .....   | 54 |
| 3.1    | Inštalčná príručka.....                                      | 54 |
| 3.2    | Používateľská príručka .....                                 | 54 |
| 3.2.1  | Nastavenia .....   | 55 |
| 3.2.2  | Ovládanie hry .....  | 55 |
| 3.3    | Príručka k súboru s logmi.....                               | 55 |
| 4      | Technická dokumentácia.....                                  | 57 |

# 1 Big picture

---

## 1.1 Úvod

Tento dokument predstavuje časť projektovej dokumentácie nášho produktu - inžinierskeho diela, ktoré je primárnym výstupným parametrom nášho projektu. V dokumente rozoberieme tvorbu tohto softvérového produktu a zameriame sa na popísanie jednotlivých častí a etáp vývoja.

Cieľom nášho projektu je návrh, implementácia, testovanie a nasadenie počítačovej hry, ktorá bude schopná získať od hráčov údaje použiteľné pre výskum zadávateľa projektu, ktorý je zameraný na ovplyvňovanie dôvery ľudí v stroje. Cieľom tohto výskumu je overiť a preskúmať možnú závislosť dôvery človeka v umelú inteligenciu od konkrétnych parametrov jej verbálnej komunikácie s daným človekom.

Pre vytvorenie hry, ktorá spĺňa parametre použiteľnosti pre uvedený výskum tiež vyplýva množstvo ohraničení.

Počas hry musí hráča sprevádzať jedna, alebo dve entity - avatari, ktorí priebežne hráčovi prostredníctvom verbálnej komunikácie dávajú rady k rozhodnutiam, ktorými hráč ovplyvňuje priebeh hry. Hra nesmie dať hráčovi jasne najavo, či konkrétne rozhodnutie bolo výhodné, alebo nevýhodné, aby hráč nemohol obviňovať avatara na základe poskytnutej rady. Rady avatarov musia byť komplementárne, teda hráč nesmie mať možnosť vyskúšať viacero rozhodnutí. Avatari sa nesmú okrem hlasu ničím iným odlišovať.

Úspech hráča v hre musí závisieť iba od rozhodnutí, ktoré počas hry urobil, nesmie byť ovplyvnený schopnosťami hráča. Hra musí byť po slovensky, minimálne dialógy a komunikácia s avatarom taktiež.

Hra bude komunikovať s hlasovým modulom, ktorý tomuto projektu zabezpečuje Slovenská Akadémia Vied (SAV) prostredníctvom REST rozhrania.

## 1.2 Globálne ciele zimného semestra

V našom projekte neboli vopred jasné požiadavky zákazníka. Z tohto dôvodu bolo našim prvým cieľom zistenie detailov prebiehajúceho výskumu a zakomponovanie všetkých týchto aspektov do hry, ktorá by okrem zbierania dát mohla poskytnúť hráčovi istú mieru zábavy.

Dôležitým krokom v každom projekte je takisto výber softvérových nástrojov, ktoré pomôžu pri vývoji. Potrebné je aj určenie typu hry vzhľadom na požiadavky a obmedzené zdroje.

Po objasnení požiadaviek a určení typu hry je hlavná priorita vymyslenie scenára hry. Treba dopredu určiť zápletku a cieľ hry, prostredie v ktorom sa bude odohrávať a priebežne vytvárať scenár jednotlivých častí.

Dôležitý aspekt hry je aj reprezentácia hráča, ktorý bude na obrazovke interagovať s jednotlivými hernými objektami. Jedným z našich prvých cieľov je vytvoriť vhodnú postavu a umožniť jej pohyb v hre.

Nakoniec sa budeme venovať návrhom prvých scén, vymyslíme ich architektúru a budeme implementovať prvé prototypy.

Na konci zimného semestra by sme chceli mať hotový prototyp hry, v ktorom bude hráč schopný

- pohybovať sa po jednej scéne,
- interagovať s objektami nachádzajúcimi sa v scéne,
- interagovať s postavou nachádzajúcou sa v scéne a
- interagovať s avatarami.

Hlavná postava hry by mala byť nakreslená a animovaná v nami zvolenom štýle.

Zároveň je našim cieľom mať v prototypu pripravenú infraštruktúru potrebnú pre komunikáciu s hlasovým modulom od SAV a v prípade možnosti túto komunikáciu aj uskutočniť.

Ak ostane čas, radi by sme tiež vymýšľali a implementovali doplnujúce zábavné prvky do hry. Zatiaľ sme vymysleli a špecifikovali ako zábavný prvok hrací automat s minihrami, v ktorom sa dajú hraním minihier vyhrať mince, za ktoré sa dajú kúpiť predmety, ktoré hráčovi v hre pridávajú nejakú schopnosť, alebo efekt.

### 1.3 Globálne ciele letného semestra

V tomto semestri je našim cieľom hru naďalej rozširovať o nové príbehové prvky, a tiež neustále vylepšovanie existujúcich častí hry. Keďže základy ako architektúra, ovládanie, grafický štýl a náčrt príbehu boli položené v zimnom semestri je nutné na nich stavať a primárne sa sústrediť na implementovanie nových rozhodnutí, ktorým bude hráč čeliť.

Pridanie čo najväčšieho počtu rozhodnutí je dôležité hlavne preto, že práve z nich sú produkované dáta pre výskum, ktorý podnietil vznik samotného projektu. Vzhľadom na vysoký počet požadovaných rozhodnutí ich nie je možné pridávať ich len rozširovaním príbehu o ďalšie kapitoly, no nutné je vymyslieť aj spôsoby ako implementovať tieto rozhodnutia do existujúcich častí hry. Okrem počtu týchto rozhodnutí je potrebné prihliadať aj na ich vážnosť, a teda vymyslieť a realizovať aj implementáciu rozhodnutí, ktoré majú väčší dopad na dej hry ako ostatné. Z tohto vyplývajú základné ciele:

- implementácia ďalších kapitol príbehu,
- dopĺňanie existujúcej kapitoly príbehu,
- vytvorenie závažných rozhodnutí, ktoré na hráča vytvárajú nátlak.

V rámci pridania ďalších kapitol je potrebné vytvorenie nových scén, či miestností, a tiež postáv, ich grafickej reprezentácie aj správania. Ďalej napísanie scenáru a prislúchajúcich dialógov založených na hrubom náčrte príbehu z minulého semestra.

Pri dopĺňaní existujúcej kapitoly sa zameriame hlavne na vkladanie a vytváranie nových rozhodnutí do existujúceho deja za použitia existujúcich grafických prvkov, postáv a scén. Taktiež je dôležité odstraňovanie chýb, ktoré spozorujeme pri hraní hry. Potrebné je aj rozšírenie správania existujúcich postáv, vylepšovanie grafiky a možností interakcie hráča s hrou formou rozšírenia dialógov a komunikácie s modulom poskytnutým SAV.

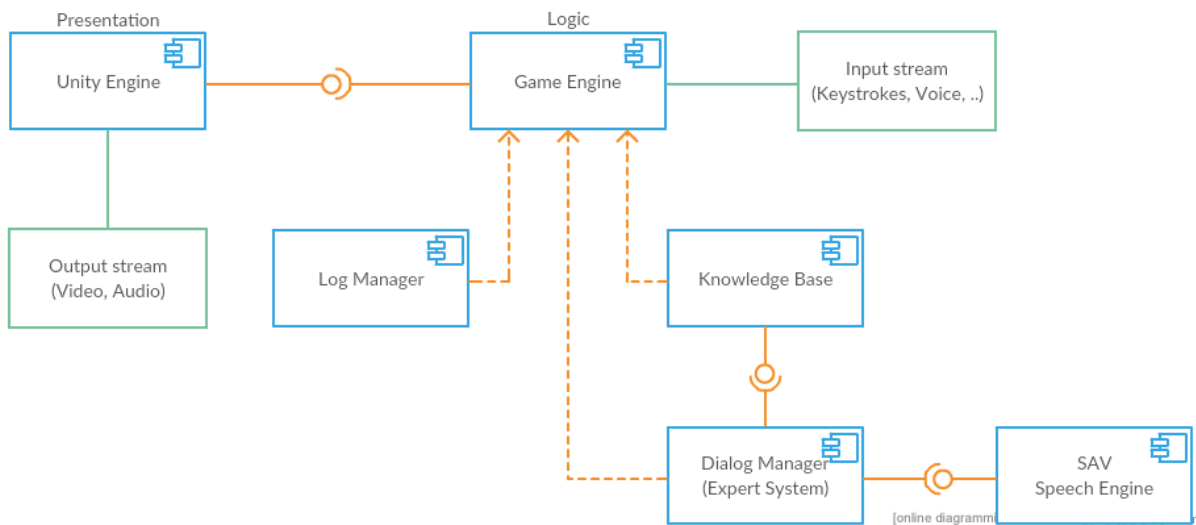
V rámci verbálnej interakcie hráča s postavami avatarov je kľúčové sa zamerať na to aby sme hráčovi navodili pocit, že s nimi komunikuje voľne, jeho slová sú vnímané a odpovedi poskytované na základe rozvíjajúcej sa konverzácie. Pre výskum je kľúčový zber dát, a teda aby korektne fungovalo prispôbovanie reči jedného z avatarov. V tomto ohľade je nutné popracovať na nastavení parametrov prispôbovanej reči a zapracovať do hry dostatočné množstvo situácií, ktoré budú vyžadovať od hráča hlasový vstup, a to z toho dôvodu, že čím viac vzoriek hráčovho hlasu získame tým bude prispôbenie avatarovho hlasu lepšie.

Medzi nové prvky, ktoré plánujeme implementovať do aktuálnej aj nastávajúcich kapitol, patrí hranie minihier prostredníctvom automatu lokalizovaného v istých scénach a diskiet, ktoré hráč získava postupovaním v príbehu. Toto nám tiež umožní vnieť do hry fiktívnu platobnú menu, ktorou bude hráč odmeňovaný počas hrania minihier. Zároveň nám to umožní vytvorenie obchodu, ktorý poskytuje ďalšie možnosti aj pre implementáciu rozhodnutí.

Za hlavný cieľ si ukladáme aby výstup tohto semestra predstavoval uzavretý hrateľný celok, ktorý obsahuje škálu rôzne dôležitých rozhodnutí a všetky prvky pridanej funkcionality, ako minihra a obchod.

## 1.4 Celkový pohľad na systém

Pri návrhu architektúry systému sme vytvorili diagram komponentov a pokúsili sa definovať jednotlivé komponenty systému a ich vzťahy.



Obrázok 1: Diagram zobrazujúci architektúru systému

Prezentačnú vrstvu systému – vykresľovanie grafiky a prehrávanie zvukov hry - nám zabezpečí Unity Engine. Ten je navrhnutý tak, že nám poskytuje rozhranie, ktorým vieme jednoducho pripájať skripty na herné komponenty a tak programovať logiku a správanie hry – Game Engine.

Súčasťou nášho Game Engine bude aj komponent, ktorý bude ukladať všetky údaje z priebehu hry potrebné pre výskum – Log Manager. Ďalším podstatným komponentom je báza znalostí o stave scény a priebehu hry – Knowledge Base.

Znalostnú bázu o stave hry bude bezpodmienečne využívať predovšetkým komponent pre správu dialógov - Dialog Manager. Tento komponent bude mať za úlohu zabezpečiť schopnosť avatarov v hre zmysluplne komunikovať s hráčom vzhľadom na súčasnú situáciu v hre.

Napokon externým komponentom nášho systému je hlasový modul, ktorý nám SAV sprístupní pomocou REST rozhrania. Tento modul bude pre nás konvertovať reč hráča na text a takisto bude pre nás syntetizovať vybraný text na reč, ktorú použijeme ako reč avatarov.

## 2 Moduly systému

---

### 2.1 Analýza

Konzultáciami so zákazníkom sme dospeli k viacerým ohraničeniam nášho projektu, ktoré sú zhrnuté v nasledujúcej časti.

1. Avatari – predstavujú kľúčovú časť hry, prostriedok na získanie experimentálnych dát požadovaných od zákazníka.
  - hra obsahuje jednu/dve entity, ktoré hráčovi radia
  - čo sa správnosti týka ich rady sú v konečnom dôsledku rovnocenné (obaja avatari poskytujú rovnaký pomer dobrých a zlých rád)
  - či bola rada avatara dobrá alebo nie hráč nesmie zistiť okamžite (moment náhody)
  - ak si už hráč vybral jednu cestu, a teda využil radu jedného z avatarov nesmie mu byť umožnené vrátiť sa a preskúmať druhú z možností
  - pokiaľ budú avatari dvaja nesmú byť vizuálne odlišní, aby sa hráč medzi ich radami nerozhodoval na základe iných vnemov ako zvukových
  - mená avatarov predstavujú problém, treba ich rozlíšiť inak ako hlasovo
  - hráč si rady od avatarov pýta alebo ich bude dostávať automaticky vždy na určitých checkpointoch, poprípade nejaká kombinácia týchto dvoch možností
  - hráč si môže vypočúť rady oboch avatarov pre každé rozhodnutie alebo, ak si bude rady pýtať môže si vypočúť len jedného z avatarov
2. Ďalšie požiadavky
  - úspech hráča nesmie závisieť od jeho schopností ale iba od toho aké rozhodnutia vykonal (ktoré rady poslúchol)
  - treba v úvodnej časti hry získať od hráča čo najdlhšiu zvukovú vzorku jeho hlasu
  - hra je po slovensky
  - neskôr sa budú robiť jazykové mutácie hry (asi nepoužívať informácie, ktorých znalosť sa viaže na lokalitu)
  - možno by bolo vhodné do hry zakomponovať nejaký ekonomický aspekt na štýl kasína, kde hráč môže získať/prísť o peniaze, pričom ani profit ani strata by opäť nemali byť evidentné okamžite keďže aj pri tomto by mu radili avatari
  - hra rozdelená do levelov
3. Technické ohraničenia
  - hra by mala byť nezávislá od platformy, ideálne webová hra
  - komunikácia s hlasovým modulom SAV pomocou REST
  - čím dlhšie zvukové nahrávky budeme posielat', tým dlhšie bude trvať jej spracovanie a poslanie odpovede
  - hra bude počúvať či hráč niečo nehovorí alebo sa mikrofón zapne po stlačení tlačidla

### 2.2 Návrh

Pri návrhu druhu hry sme sa snažili zohľadniť najmä potrebu, aby bol priebeh hry riadený rozhodnutiami hráča. Preto sme vymýšľali taký príbeh, v ktorom by dej mohol byť rozvetvovaný do alternatívnych dejových línií.

V súlade s ohraničeniami výskumnej časti projektu sme sa rozhodli za druh hry zvoliť adventúru, keďže spomedzi rôznych druhov hier najviac vyhovuje našim požiadavkám na

použitelnosť radiacích avatarov. Navyše tento druh hry nám bol zadávateľom projektu predložený ako preferovaná možnosť.

### 2.2.1 Scenár

Navrhli sme množstvo zaujímavých scenárov. V každom z nich nám však chýbalo celkové porozumenie a stotožnenie sa celého tímu s hlavnou myšlienkou - zápletkou daného scenára. Keďže vo väčšine našich námetov sa prejavovali prvky žánru sci-fi, nakoniec sme sa rozhodli vymyslieť príbeh s použitím adopcie črt existujúcich sci-fi diel, ktoré všetci už poznáme. S cieľom pobaviť hráča sme sa taktiež zhodli na tom, že ak to bude možné, pokúsime sa skombinovať črty rôznych sci-fi diel a použiť v našej hre kontrast medzi nimi.

Vymysleli sme preto príbeh a zápletku postavenú na kombinácií reálneho sveta a čarodejníckeho sveta tak, ako je predstavený v príbehu Harryho Pottera. Samozrejme sme si nechali otvorený priestor pre pridávanie rôznych prvkov z ďalších sci-fi svetov. Príbeh sa odohráva vo svete, kde čarodejníci nie sú schopní používať techniku, pretože v ich bezprostrednom okolí zlyháva. Tým pádom sú v bežnom svete nepoužiteľní a spoločensky utláčaní, čo sa im samozrejme nepáči.

Hlavnou motiváciou hráča v našom príbehu je pomôcť vyriešiť konflikt medzi dvoma svetmi s použitím vedeckého prístupu. Mal by sa podieľať na výskume a vývoji zariadenia ktoré umožní čarodejníkom fungovať vo svete nečarodejníkov. Aby sa dostal k tomuto výskumu, ide študovať na čarodejnícku školu, kde sa zariadenie vyvíja a tam sa musí popasovať s rôznymi nástrahami a prekážkami a eliminovať záhadné sabotérske indivíduá.

Pred počiatkom prvej kapitoly je hráč uvedený do deja prostredníctvom krátkeho textu, kde sú stručne vysvetlené pomery vo fiktívnom svete, v ktorom sa dej hry odohráva, kde sa hráč nachádza a taktiež udalosti, ktoré k tejto situácií viedli.

Cieľom prvej kapitoly hry je aby hráč pomohol Jozefovi, hlavnej postave nášho príbehu, ubytovať sa na internáte. Keďže Jozef je novým študentom na magickej univerzite je pre neho prostredie rovnako neznáme ako pre hráča. Kapitola slúži aj na oboznámenie hráča s ovládaním hry a funkciami akými sú napríklad hranie minihier a obchod. Počas tejto kapitoly sú hráčovi predstavené niektoré kľúčové aj vedľajšie postavy príbehu. Hráč sa tu po prvý krát stretne s Ančou, ktorá predstavuje jeden zo zábavných prvkov hry, Zuzou, od ktorej získava disketu obsahujúcu prvú minihru a hlavne spozná avatarov, ktorí mu budú po zvyšok hry radiť. Okrem interakcií s týmito postavami je hráč nútený prekonávať prekážky, napríklad vo forme zamknutých magických dverí, ktoré od neho požadujú heslo inak ho nenechajú prejsť. Kapitola sa končí po tom čo Jozef úspešne nájde internáty, vyberie si fakultu, ktorej bude počas svojich štúdií súčasťou a je mu pridelená jeho izba.

V druhej kapitole sa Jozef, a teda aj hráč, stretáva s ďalšími postavami ako napríklad s jedným z jeho profesorov, ale hlavne so zápornou postavou Vladom, ktorý mu už od začiatku začne spôsobovať problémy. Vďaka nemu bude Jozef meškať na svoju prvú hodinu. V rámci tejto kapitoly stále pracujeme na dialógoch a správaní profesora, ktorý spomínanú hodinu vyučuje. Táto kapitola bude taktiež obsahovať zásadné rozhodnutie, kde má chybný výber akcie vážne dôsledky a môže pre hráča predstavovať ukončenie celej hry.

### 2.2.2 Grafika a interakcia

Hra bude graficky reprezentovaná s použitím 2D obrázkov. Objekty na scénach sa však budú vykresľovať na rôznych vrstvách, čo by malo evokovať priestorovosť scén. V scénach sa teda bude dať pohybovať v 3D.

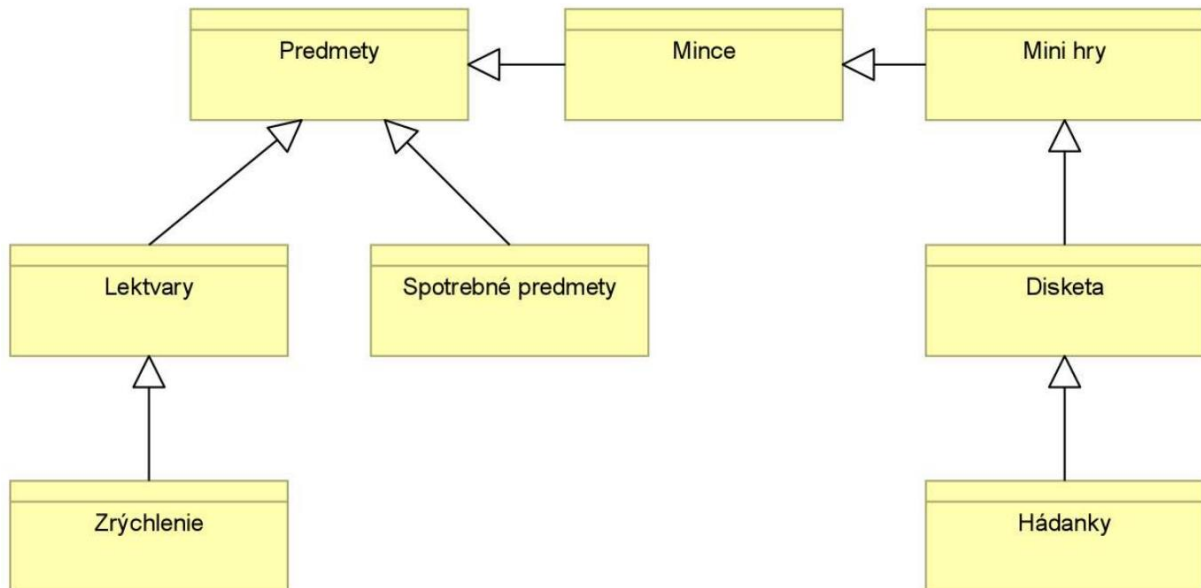


Hráč bude s hrou interagovať najmä s pomocou myši a klávesnice. Pri komunikácii s avatarmi alebo inej použitej verbálnej komunikácii bude s hrou interagovať pomocou mikrofónu.

### 2.2.3 Model hry

Rozpracovali sme taktiež model architektúry herného mechanizmu – Game Engine – pre vnútornú reprezentáciu scén, objektov a stavov hry. Tento model predstavuje high level pohľad na hernú logiku.

Na obrázku nižšie je herný model našej hry, reprezentujúci vzájomné vzťahy herných elementov, ktoré budú hráča motivovať a baviť.

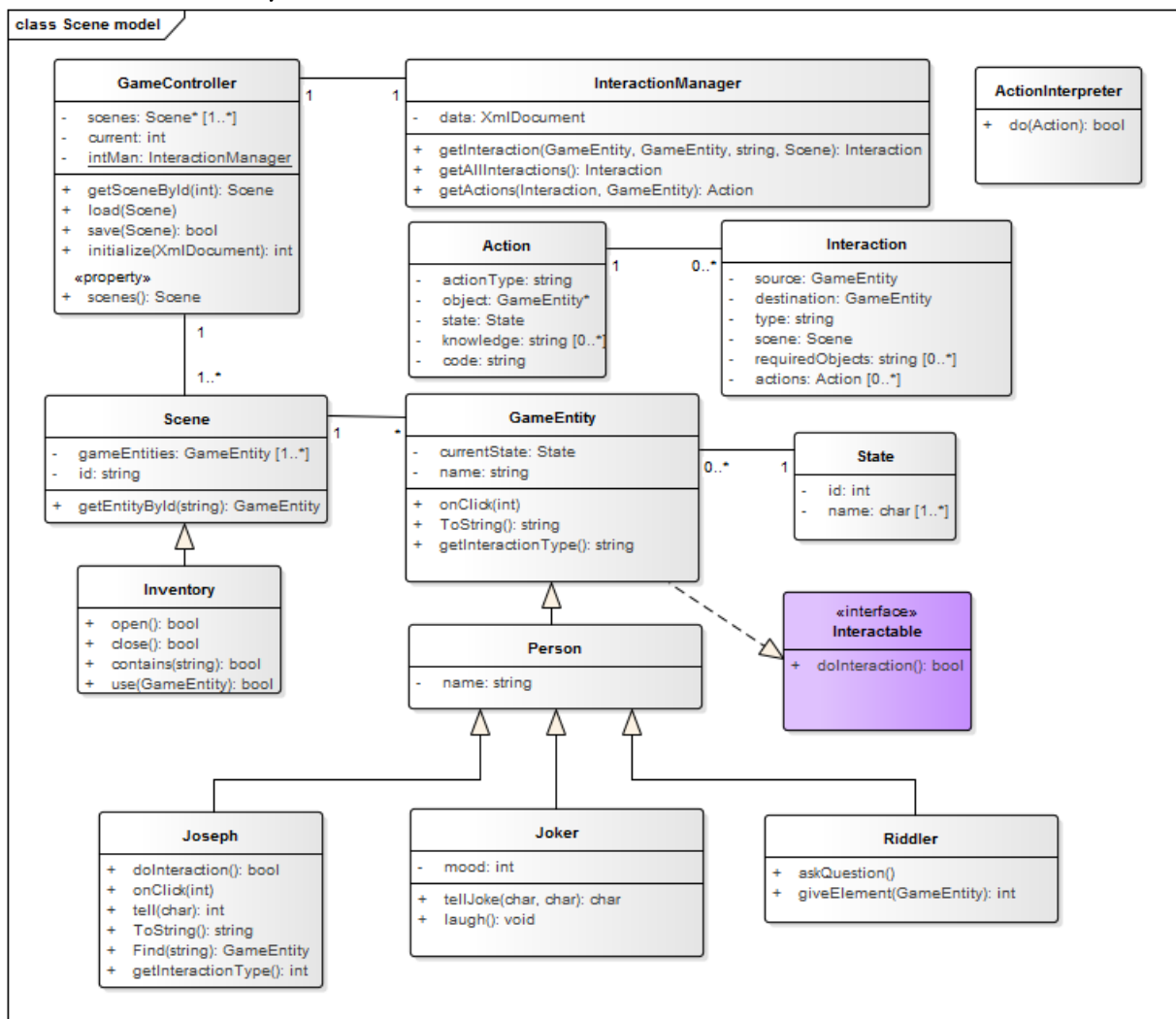


Obrázok 2: Diagram herného modelu

Hráč má možnosť pri konverzácií so špecifickou osobou vyriešiť hádanku, za čo dostane disketu do hracieho automatu, ktorá mu v ňom odomkne nové mini hry. Za hranie minihry môže hráč získať mince, za ktoré si môže kúpiť predmety v obchode v hlavnej hre.

Predmety sú dvoch typov: lektvary a spotrebné predmety. Lektvary pridajú hráčovi dočasne nejakú schopnosť(napr. zrýchlenie). Spotrebné predmety budú napríklad čokoláda ktorou môže uplácať iné postavy, lampáš, ktorý sa vybije po použití.

## 2.2.4 Model scény



Obrázok 3: Diagram tried znázorňujúci model scény

Na vrchu hierarchie nášho modelu scény sa globálna statická trieda GameController stará o inicializáciu, načítavanie a ukladanie scén a o prenos údajov medzi scénami. Obsahuje zoznam všetkých scén hry a zároveň aktuálne načítanú scénu.

V epicentre tohto modelu je trieda GameEntity. Každá inštancia tejto triedy reprezentuje jeden z interakcie schopných objektov na scéne. Každá scéna (trieda Scene) preto v sebe nesie zoznam týchto entít. Inventár je špeciálny druh scény, v ktorom sa dá manipulovať s nesenými predmetmi.

Každý objekt má nejaký stav. Stavy sú preddefinované v externom XML súbore a sú použité pri riadení interakcií objektov.

Trieda GameEntity implementuje rozhranie Interactable, preto každý objekt musí mať definovanú odpoveď na pokus o interakciu s ním.

Trieda Interaction reprezentuje zoznam informácií o interakciách a je použitá ako objekt, pomocou ktorého si interakciu vykonávajúce metódy vymieňajú informácie o interakciách. Trieda Action je typ, ktorý nesie informácie o akciách, ktorá sa má na úrovni kódu v hre vykonať ako dôsledok priebehu konkrétnej interakcie. Na reprezentáciu interakcie môže byť potrebné vykonanie viacerých akcií.

Statická trieda *InteractionManager* poskytuje metódy, ktoré pracujú s definíciami interakcií v externom XML súbore a zabezpečuje správny výber inštalácie interakcie, prípadne adekvátnu reakciu pri neplatnej požiadavke o interakciu. Pracuje s objektom *Interaction*.

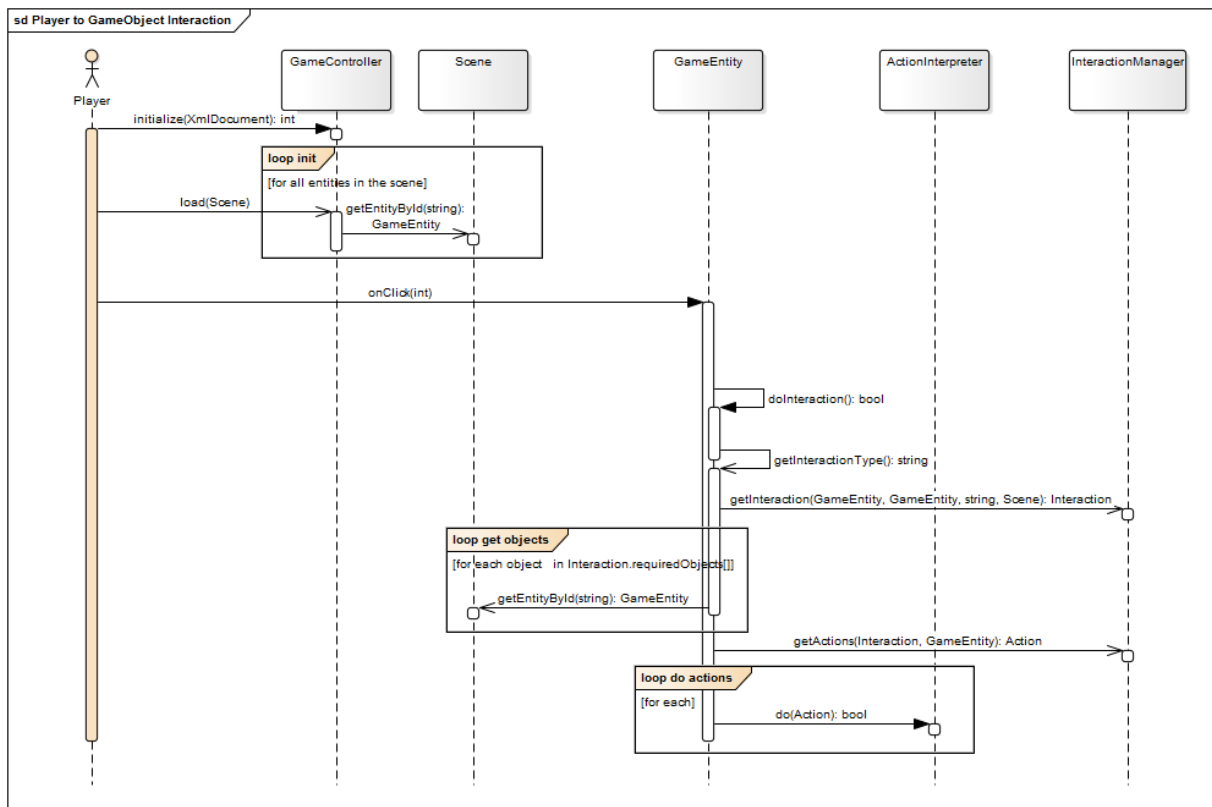
Statická trieda *ActionInterpreter* je trieda, ktorá poskytuje funkcionality pre vykonanie akcie definovanej objektom *Action*.

*Person* je špecifický typ herného objektu, ktorého inštalácie reprezentujú generické osoby.

*Joseph*, *Joker* a *Riddler* sú špecifikácie generickej osoby, teda reprezentujú modely špecifických osôb použitých v hre.

## 2.2.5 Model interakcií

Poznámka: Lifeline-y reprezentujú objekty - inštalácie uvedených tried.



Obrázok 4: Diagram interakcií znázorňujúci priebeh interakcie v hre

Hráč pri spustení aplikácie hry **inicializuje zoznam scén** hry nad globálnym objektom *GameController*.

Následne na začiatku novej hry, alebo pri načítaní uloženej sa načíta pole objektov nachádzajúcich sa v načítavanej scéne.

Počas hry hráč kliknutím na objekt v scéne spustí požiadavku o interakciu typu *clickTo*, čo spustí metódu *doInteraction*. V rámci nej si entita načíta typ interakcie na základe jej aktuálneho stavu a vypýta si inštaláciu interakcie od inštalácie triedy *InteractionManager*.

Na základe objektov potrebných na výber inštalácie interakcie ktoré si *InteractionManager* obratom vypýta si metóda ďalej vypýta od scény ich súčasné stavy a obohatí o ne dátový objekt *Interaction*. Následne ho pošle ako argument požiadavky o akcie, ktoré treba vykonať objektu *InteractionManager*.

Pre získané akcie napokon cyklicky požiadava *ActionInterpreter* o ich vykonanie.

## 2.3 Stavový priestor hry

Tento dokument obsahuje vysvetlenie diagramov, ktoré slúžia na opis jednotlivých stavov objektov v hre a možností získania znalostí. Diagramy popisujú aké interakcie a za akých okolností je možné vykonávať medzi objektami v scéne.


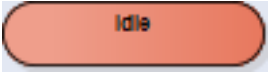
### 2.3.1 Pojmy

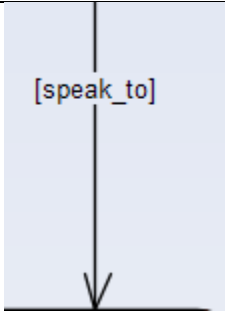
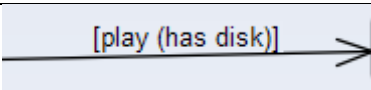
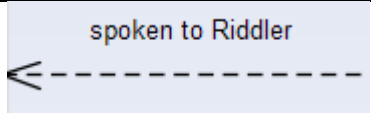

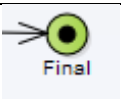
- **Scéna**
  - Jedna ucelenú časť príbehu v kapitole. Má svoje vstupné/ výstupné podmienky. Väčšinou reprezentovaná jednou miestnosťou prípadne dlhšou chodbou kde je úlohou hráča túto miestnosť opustiť
- **Objekt**
  - V scéne sa vyskytujú objekty, s ktorými je možné interagovať. Hráč (hlavná postava Jozef) môže hovoriť s inými osobami, klikať na predmety, rozprávať sa s avatarmi – to všetko sú objekty v scéne
- **Stav**
  - Situácia, v ktorej sa nachádza objekt. Zmena stavu môže predstavovať aj zmenu vzhl'adu objektu v hre. Pr. Jozef svieti lampášom, je v stave *green\_light*.
- **Znalosť**
  - Objekt má svoje vedomosti o sebe a hraním hry získava ďalšie. Objekt si pamätá svoje doterajšie akcie ak je to nutné. Pr. Hádankárka vie, že Jozefovi dala otázku 1.
- **Interakcia**
  - Akcia vykonaná medzi objektami v scéne sa nazýva interakcia. Stav jednotlivých objektov sa menia na základe vyvolanej interakcie. Výsledkom interakcie môže byť aj získanie znalosti. Pr. Jozef sa rozpráva s Vtipkárkou – interakcia *speak\_to* medzi Jozefom a Vtipkárkou, kde sa mení stav Jozefa aj Vtipkárky na *speaking* a Vtipkárka získava znalosť, že Jozefovi povedala vtip číslo 1.

### 2.3.2 Objekty

V diagrame používame nasledovné značenia:

Tabuľka 1: Tabuľka značiek používaných v stavových diagramoch

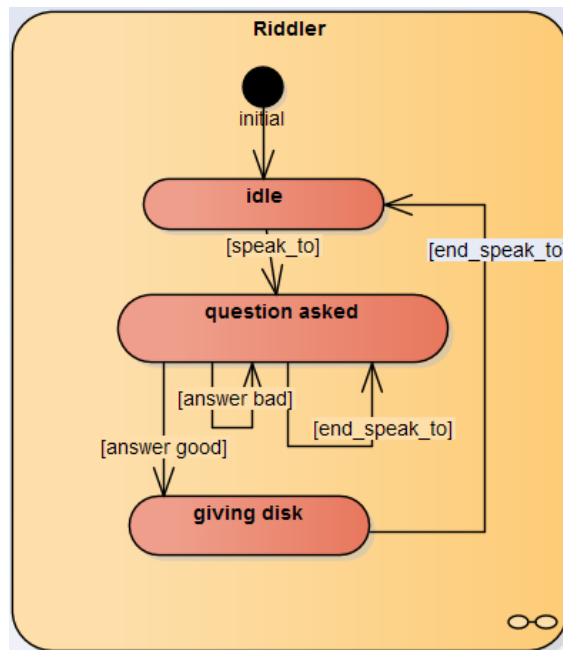
| Značka  | Názov  | Konvencie  |
|---|--------|--|
|  | Objekt | Veľké písmeno<br>Jedno slovo   |
|  | Stav   | Malé písmeno<br>Aj viac slov oddelených medzerou<br>Umiestňuje sa vo vnútri objektu<br>a tak vyjadruje jeho stav |

|   |  |   |
|---|--|---|
|    | <p>Interakcia-nepodmiernená</p> <p>Môže sa vykonať bez ohľadu na obsah znalostí</p>            | <p>Malé písmeno</p> <p>Viac slov oddelených _</p> <p>Uzatvorené v [ ] zátvorkách</p> <p>Smeruje od objektu, ktorý interakciu vyvoláva do objektu, ktorý ju prijíma</p> <p>Vykonanie interakcie je závislé od stavu objektu, v ktorom je vyvolané a smeruje do iného stavu alebo do toho istého.</p> |
|    | <p>Interakcia – podmienená</p> <p>Vykoná sa len ak má vyvolávajúci objekt potrebnú znalosť</p> | <p>Malé písmeno</p> <p>Viac slov oddelených _</p> <p>Uzatvorené v [ ] zátvorkách</p> <p>Znalosť/i, ktorá/é je/sú potrebná/é na vykonanie akcie je/sú vypísaná/é v ( )</p>   |
|    | <p>Znalosť</p>   | <p>Malým písmenom</p> <p>Viac slov oddelených medzerou</p> <p>Smeruje od stavu, ktorý znalosť vytvára do objektu, ktorý túto znalosť získava</p>  |
|  | <p>Začiatok</p>  | <p>Malým písmenom</p> <p>Initial – začiatok scény, objekt sa vytvorí</p> <p>Vyskytuje sa v stavovom diagrame objektu</p>  |
|  | <p>Koniec</p>  | <p>Final - Koniec celej scény</p> <p>Vyskytuje sa len jeden v celom diagrame</p>  |

### 2.3.3 Tvorba stavového priestoru

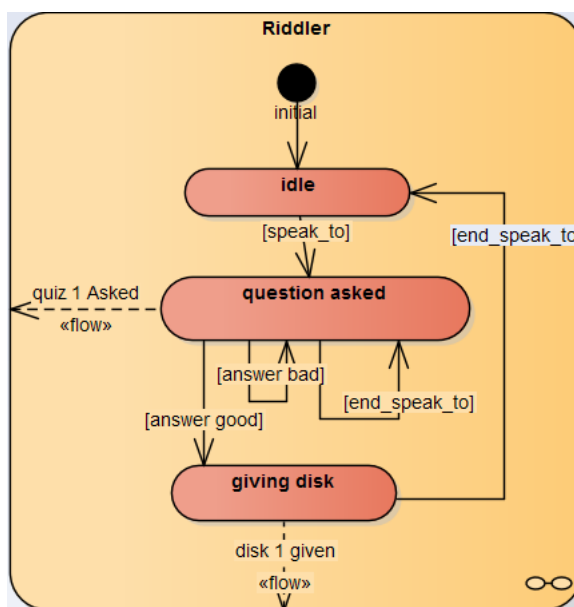
Jeden diagram sa vytvára pre jednu scénu v hre.

- Najprv si identifikujeme všetky objekty v scéne, ktoré sa tu budú vyskytovať a budú mať vplyv na vývoj deja
- Pre každý objekt si vytvoríme jeho vnútorný stavový diagram, určíme si stavy, v ktorých sa môže nachádzať a zdefinujeme jednotlivé prechody medzi týmito stavmi



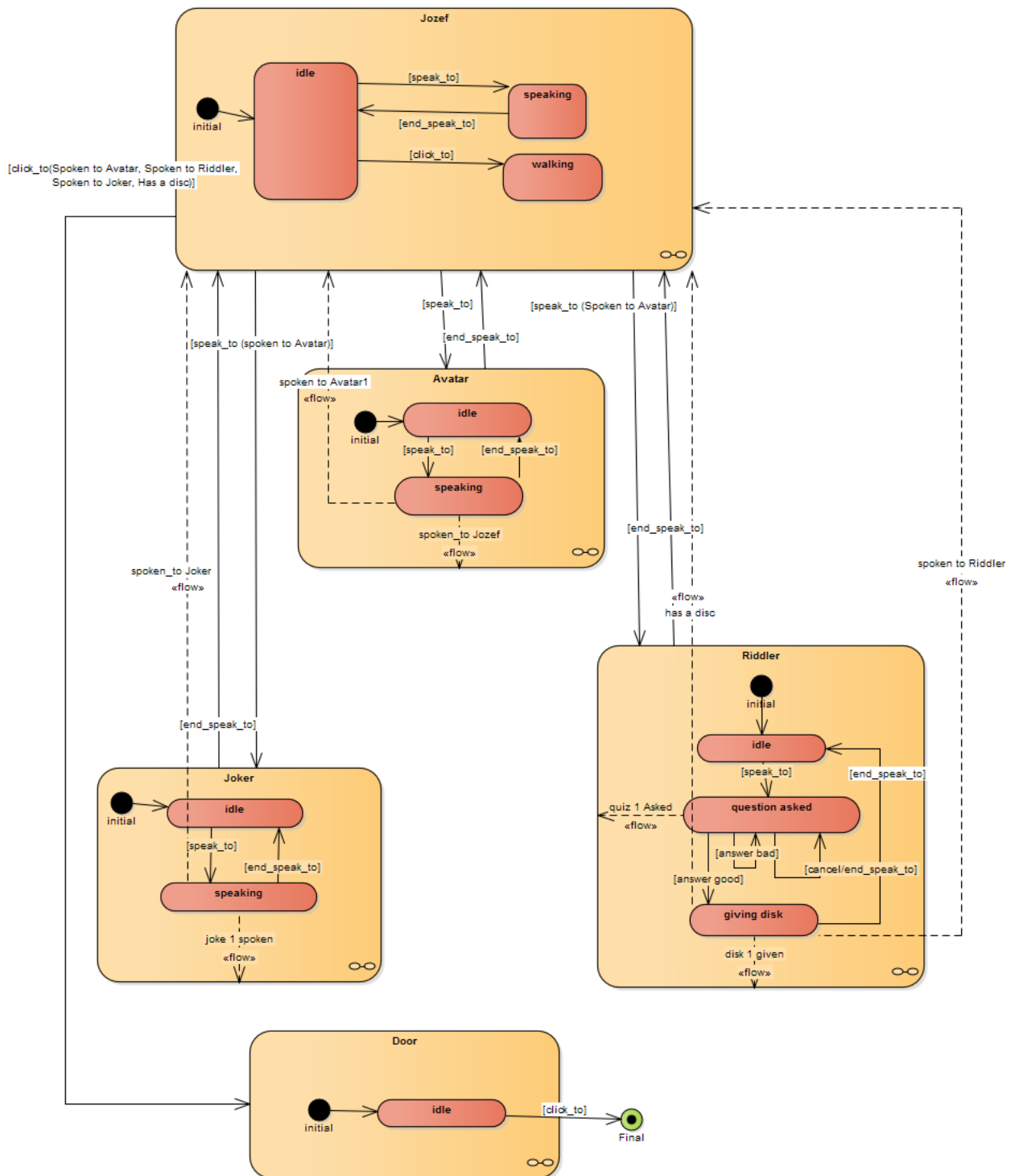
Obrázok 5: Objekt s popísanými stavmi

- V prípade, že v niektorých stavoch získava objekt znalosť sám od seba z označíme



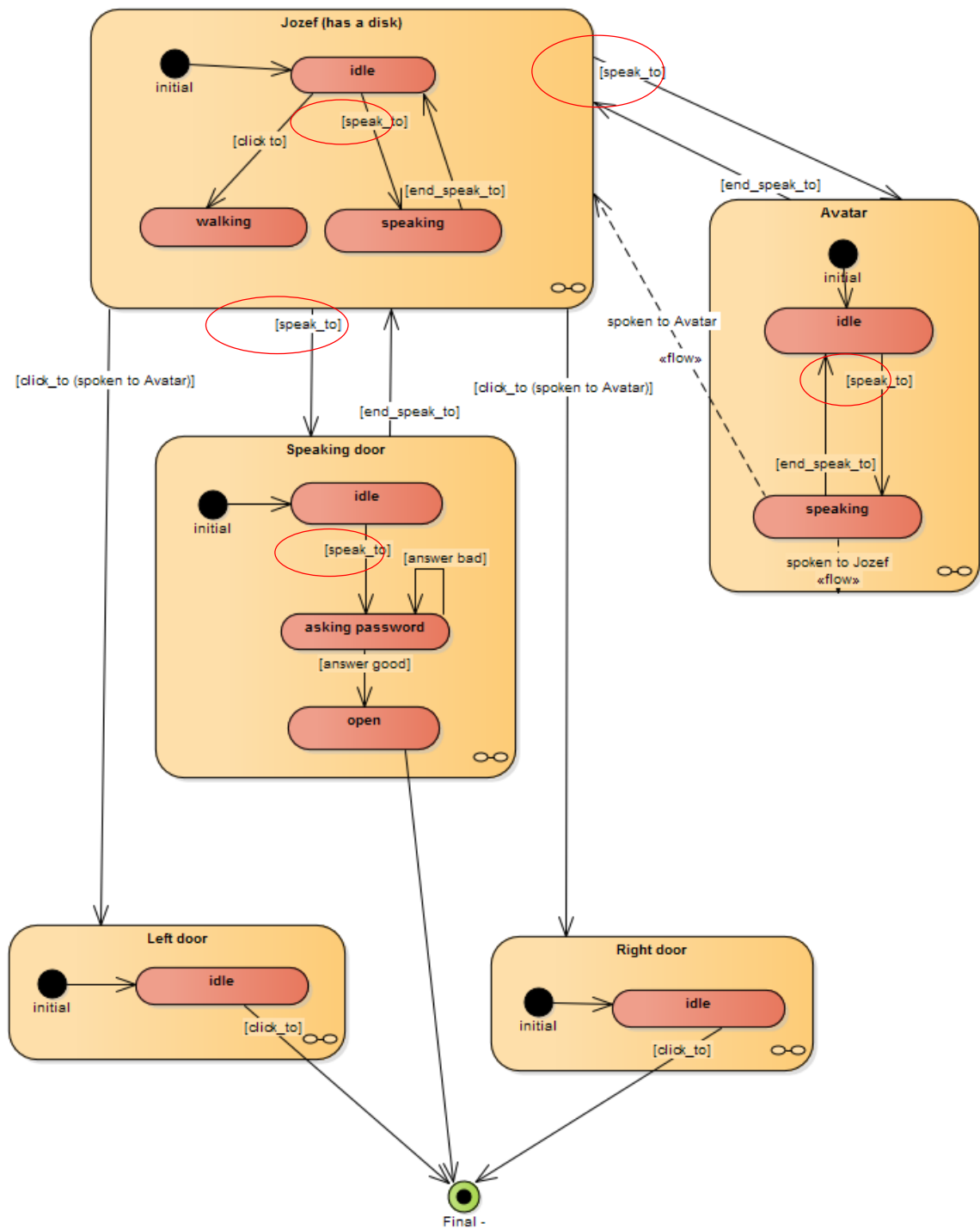
Obrázok 6: Objekt s popísanými stavmi a odovzdávanými znalosťami

- Keď máme popísané všetky objekty, naznačíme všetky interakcie, ktoré je možné medzi objektami vykonávať
  - o ak je interakcia podmienená je dôležité na to nezabudnúť
  - o ak sme použili podmienenú interakciu je dôležité, aby objekt nejakým spôsobom túto znalosť, ktorá vystupuje ako podmienka, mohol získať



Obrázok 7: Časť diagramu, znázorňuje podmienenú interakciu a možnosť získania potrebnej znalosť

- ak objekt vyvoláva interakciu k druhému objektu, musí sa táto interakcia vyskytovať aj v stavových diagramoch oboch objektov, ktoré naznačujú zmenu stavu daných objektov v prípade interakcie

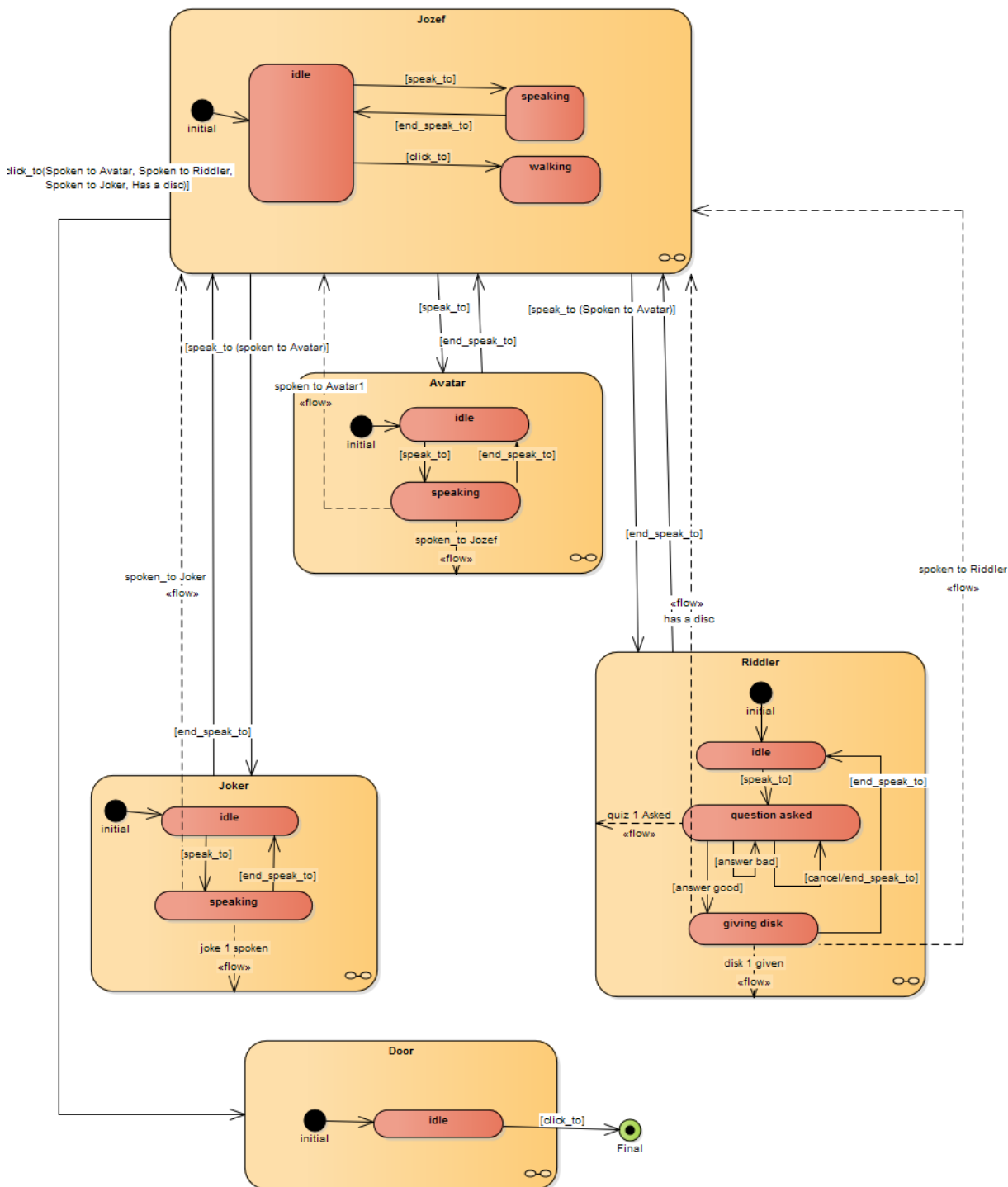


Obrázok 8: Časť diagramu, znázorňuje použitie interakcie `speak_to`, Jozef môže interagovať s objektom `Speaking door` a `Avatar` pomocou interakcie `speak_to` (dve šípky vedú k dvom krabičkám), avšak vnútri sa jeho stav mení stále rovnako



## 2.3.4 Vytvorené diagramy

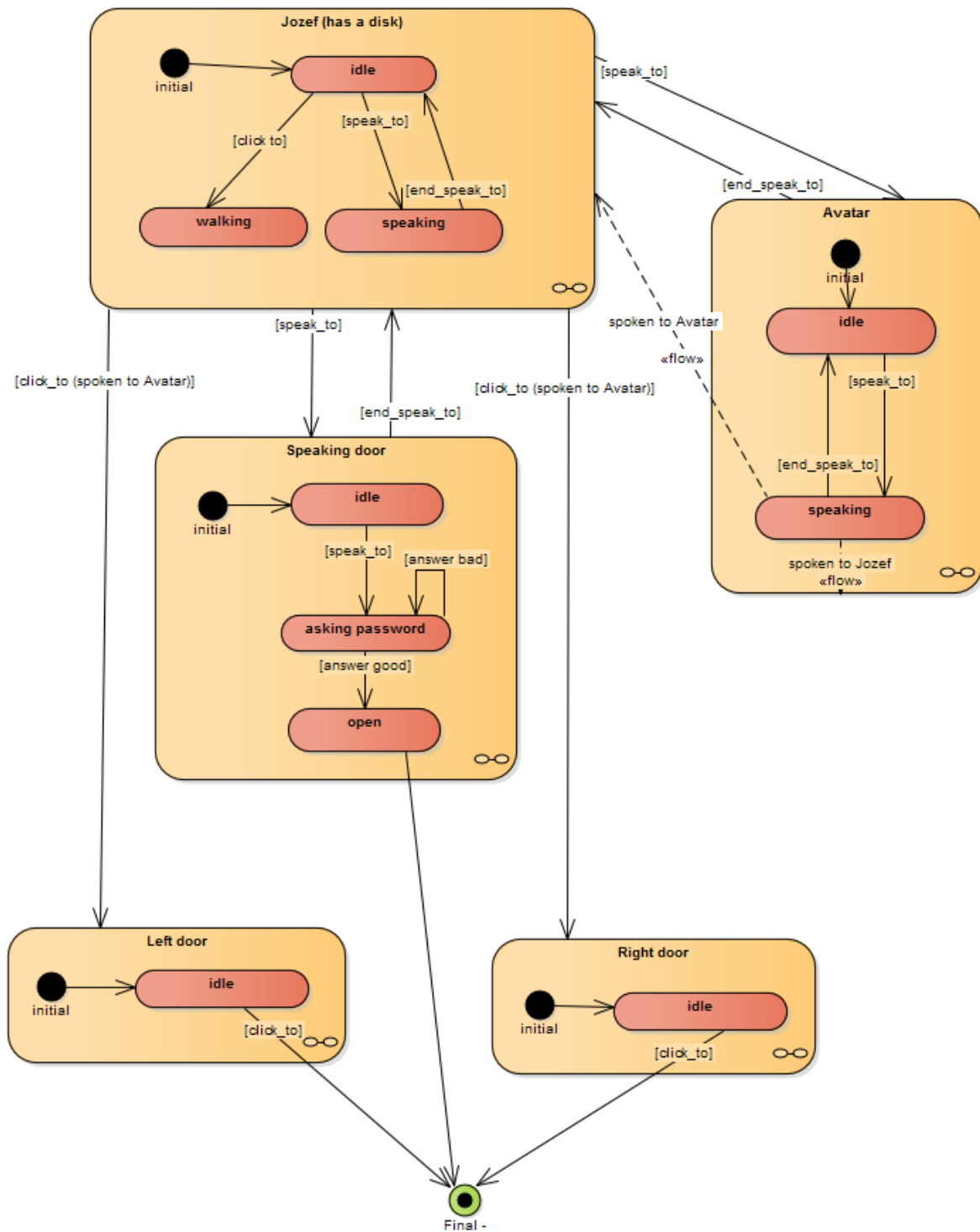
### Scéna 1



Obrázok 9: Stavový priestor scény 1

Tento diagram popisuje všetky stavy pre každý objekt v prvej scény a ich možné vzájomné interakcie. Vidíme teda, že v prvej scény sa nachádza päť objektov, vrátane hlavnej postavy. Každý objekt sa pri načítaní scény ocitne v stave `idle` a pomocou vzájomných interakcií sa môže prepínať do iných stavov. Interakcie medzi dverami, Jokerom a Riddlerom sú podmienené aj znalosťami hlavnej postavy.

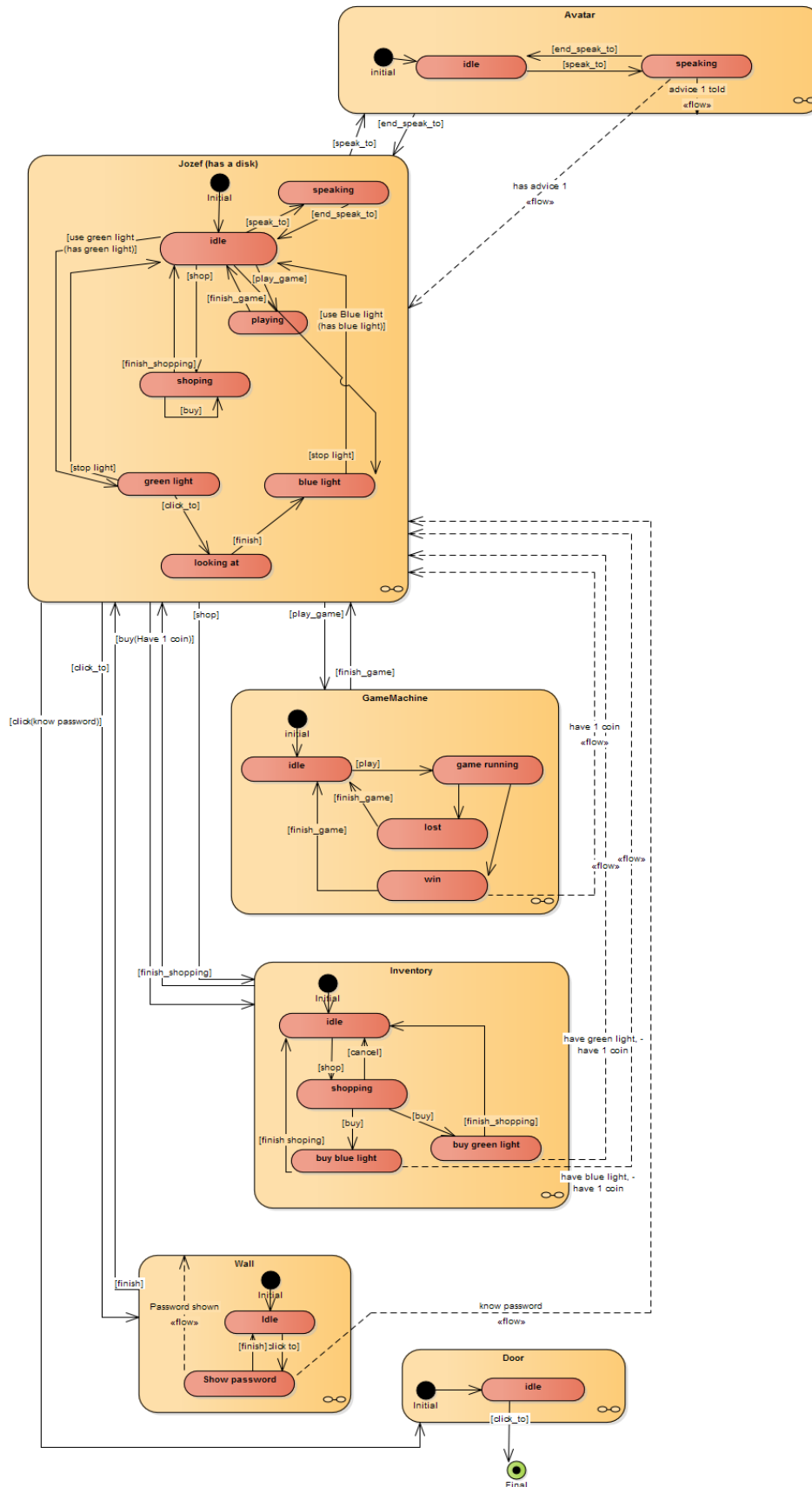
## Scéna 2



Obrázok 10: Stavový priestor scény 2

Obrázok č. 10 opisuje stavový priestor druhej scény. Nachádza sa tu 5 objektov, každý objekt začína v stave *idle*. Podmienené interakcie znalosťami sú pri interakcii medzi hlavnou postavou a dverami. Z tejto scény môžeme odísť pri interakcii s *Left door*, *Right door* alebo *Speaking door*.

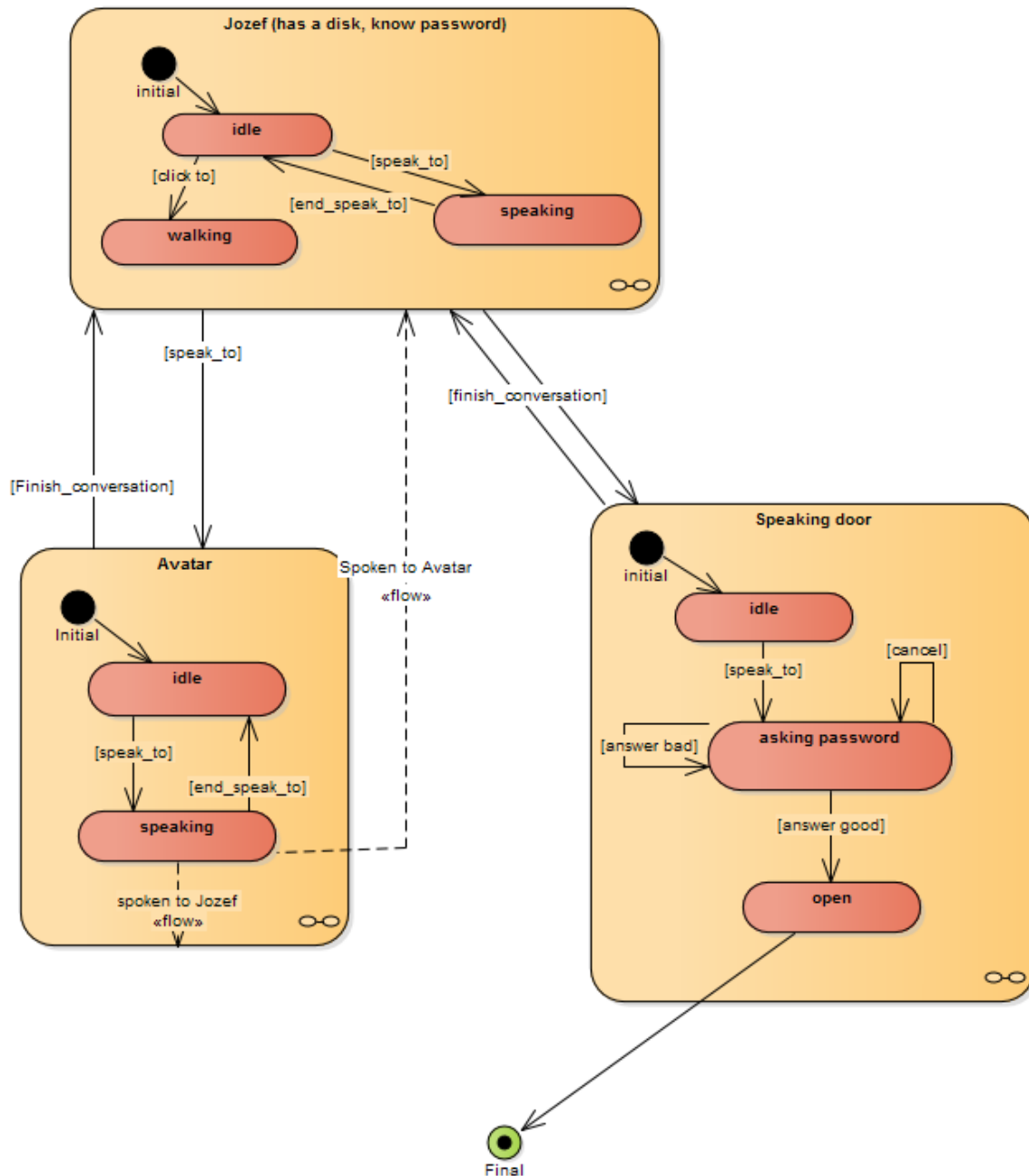
### Scéna 3



Obrázok 11: Stavový priestor scény 3

Tretia scéna je v doposiaľ navrhnutom príbehu hry najzložitejšia. Aby mohol hráč túto hru opustiť musí interagovať s dverami interakciou *click\_to* a so znalosťou *Have password*. Avšak kým získa heslo musí vykonať niekoľko interakcií – zahrať si hru, získať v nej *1 coin*, kúpiť si svetlo v *Inventory* a zasvietiť na stenu, ktoré mu ukáže heslo.

#### Scéna 4



Obrázok 12: Stavový priestor scény 4

Scéna č. 4 je veľmi podobná druhej scéne avšak už zmizli objekty *Right door* a *Left door*, ktoré viedli do scény 3. Táto scéna slúži na aplikáciu hesla z predchádzajúcej scény.

## 2.4 Reprezentácia interakcií

Vychádzajúc z diagramov stavového priestoru hry sme zostrojili reprezentáciu interakcií, z ktorej bude možné počas hry zistiť, ktorá interakcia môže medzi ktorými objektami

prebehnúť a zároveň sú v nej zadané akcie, ktoré sa majú pri tejto interakcii vykonať. Všetky informácie o interakciách sa nachádzajú v xml súbore, ktorý má nasledovnú štruktúru.

#### 2.4.1.1 Základné dátové typy

Koreňový element je **definitions** a obsahuje 5 základných podtypov:

- **gameEntities** – obsahuje všetky herné entity, ktoré sa budú v súbore využívať
- **scenes** – obsahuje všetky scény, ktoré sa budú v súbore využívať
- **interactionTypes** – definuje typy interakcií využité v súbore
- **states** – definuje stavy využité v súbore
- **interactions** – definuje jednotlivé interakcie

Každý z týchto typov môže obsahovať 1 alebo viac elementov príslušného typu (**gameEntities** – **gameEntity**, **scenes** – **scene**, **interactionTypes** – **interactionType**, **states** – **state**, **interactions** - **interaction**).

##### GameEntity

- má atribút **id**, ktorý musí byť zadaný a jedinečný
- ďalej obsahuje element **name** typu string, ktorý obsahuje meno entity

##### Scene

- má atribút **id**, ktorý musí byť zadaný a jedinečný
- **name** typu string – obsahuje meno scény
- môže obsahovať elementy **gameEntity**, ktoré musia mať zadaný atribút **ref**, v ktorom je určená referencia na id existujúceho elementu **gameEntity** (definovaného v **gameEntities**)

##### InteractionType

- atribút **id**, ktorý musí byť zadaný a jedinečný
- **name** typu string – obsahuje názov interakcie

##### State

- atribút **id**, ktorý musí byť zadaný a jedinečný
- **name** typu string – obsahuje názov stavu

##### Interaction

- atribút **id**, ktorý musí byť zadaný a jedinečný
- ďalej obsahuje elementy:
  - **source** – musí mať zadaný atribút **ref**, ktorý odkazuje na **gameEntity id**
  - **destination** – musí mať zadaný atribút **ref**, ktorý odkazuje na **gameEntity id**
  - **scene** – musí mať zadaný atribút **ref**, ktorý odkazuje na **scene id**
  - **interactionType** – musí mať zadaný atribút **ref**, ktorý odkazuje na **interactionType id**
  - **requiredObjects** – 1 alebo viac elementov typu **requiredObject**
  - **possibilities** – 1 alebo viac elementov typu **possibility**
- Interakcia je určená zdrojom (source), cieľom (destination), scénou a typom. Zdroj je herná entita, ktorá interakciu vyvoláva a cieľ je herná entita, s ktorou zdroj chce interagovať.
- Scene určuje, v ktorej scéne sa interakcia nachádza

- Interaction type určuje typ interakcie, ktorú zdroj interakcie využíva (napr. speakTo, clickTo)
- Takto určená interakcia má viac možností, ako môže prebehnúť (napríklad ak chce gameEntity „Jozef“ uskutočniť interackiu „speakTo“ s objektom „Avatar“, tak môže táto interakcia prebehnúť rozdielne predtým ako sa porozprával s hádankárkou a potom)
- Možnosti ako môže interakcia prebehnúť sa rozlišujú podľa stavov a znalostí niektorých objektov v scéne. Objekty, ktoré sú potrebné na rozlíšenie interakcie sú potom zadané v **requiredObjects**.
- Samotné možnosti, ktorými môže interakcia prebehnúť sú určené v elemente **possibilities**

### RequiredObject

- musí mať zadaný atribút **ref**, ktorý odkazuje na **gameEntity id**

### Possibility

- atribút **id**, ktorý musí byť zadaný a jedinečný
- obsahuje 2 základné skupiny elementov:
  - **conditions** – podmienky, ktoré musia byť splnené, aby sa táto možnosť mohla vykonať
  - **actions** – akcie, ktoré sa vykonajú, keď budú splnené podmienky
- Ak budú splnené všetky podmienky určené v conditions môžu sa vykonať akcie z actions

### Condition

- Jedna podmienka, ktorá určuje v akom stave musí byť herná entita alebo aké musí mať vlastnosti
- Musí obsahovať element **gameEntity**, ktorý musí obsahovať atribút **ref**, odkazujúci na **gameEntity id**
- Musí obsahovať jeden z dvoch typov elementov (ale nie obidva) – **state** alebo **knowledge**
- **state** – určuje stav, v ktorom musí byť herná entita; musí obsahovať atribút **ref**, odkazujúci na **state id**
- **knowledge** – elementov tohto typu môže byť 1 a viac a obsahujú požadovanú znalosť hernej entity

## Action

- Jedna akcia, ktorú spracuje herný interpret a vykoná ju v hre
- Sú definované typy akcií:
  - **changeState** – zmení stav hernej entity
  - **addKnowledge** – pridá požadované znalosti hernej entite
  - **executeDialog** – vykoná externe definovaný dialóg
  - **doConversation** – vykoná externe definovanú konverzáciu
  - **sayJoke** – akcia týkajúca sa postavy Vtipkárky; zavolá povedanie jedného vtipu
  - **addItem** – pridá objekt do inventáru hráča
  - **removeItem** – odoberie objekt z inventáru hráča
  - **leaveScene** – opustí aktuálnu a načína novú scénu
  - **unityAction** – vykoná jednu zo zdefinovaných akcií v Unity Engine
  - **startAvatarBlink** – spustí blikanie avatarov, keď hráčovi oznamujú, že majú pre neho radu
  - **stopAvatarBlink** – zastaví blikanie avatarov
- **changeState** musí obsahovať element **gameEntity** s atribútom **ref** odkazujúcim na **gameEntity id** a element **state** s atribútom **ref** odkazujúcim na **state id**
- **addKnowledge** musí obsahovať element **gameEntity** s atribútom **ref** odkazujúcim na **gameEntity id** a jeden alebo viac elementov **knowledge**, ktoré obsahujú reťazec so znalosťou, ktorá sa má pridať
- **executeDialog** – obsahuje id definovaného dialógu
- **doConversation** – obsahuje id definovanej konverzácie
- **addItem** – musí obsahovať vnorený element **item**, ktorý pomocou atribútu **ref** ukazuje na id objektu, ktorý sa má pridať
- **removeItem** – musí obsahovať referenciu na objekt rovnakým spôsobom ako pri **addItem**
- **leaveScene** – musí obsahovať vnorený element **toScene**, ktorý obsahuje atribút **ref** s referenciou na scénu, do ktorej sa má prejsť
- **unityAction** – obsahuje definované typy:
  - **changeSpriteLayer** – zmení vrstvu zobrazovania textúry v unity
  - **changeHotspotState** – zmení stav hotspotu
  - **sendMessage** – pošle ľubovoľnú správu ľubovoľnému Unity objektu
  - **playSound** – prehrá určený zvuk
  - **destroyGameObject** – zmaže Unity objekt
  - **pickGameObject** – spustí animáciu pridania objektu do inventáru a zmaže daný objekt zo scény
  - **openInventory** – otvorí inventár hráča
  - **openShop** – otvorí obchod hráča

2.4.1.2 Príklad reprezentácie:

```
<definitions
  <gameEntities>
    <gameEntity id="jozef">
      <name>Jozef</name>
    </gameEntity>
    <gameEntity id="avatar">
      <name>Avatar</name>
    </gameEntity>
  </gameEntities>
```

```

</gameEntities>
<scenes>
  <scene id="scene_1">
    <name>Scena 1</name>
    <gameEntity ref="jozef"/>
    <gameEntity ref="avatar"/>
  </scene>
</scenes>
<interactionTypes>
  <interactionType id="speakTo">
    <name>speakTo</name>
  </interactionType>
</interactionTypes>
<states>
  <state id="idle">
    <name>Idle</name>
  </state>
  <state id="speaking">
    <name>Speaking</name>
  </state>
</states>
<interactions>
  <interaction id="int1">
    <source ref="jozef"/>
    <destination ref="avatar"/>
    <scene ref="scene_1"/>
    <interactionType ref="speakTo"/>
    <requiredObjects>
      <requiredObject ref="jozef"/>
      <requiredObject ref="avatar"/>
    </requiredObjects>
    <possibilities>
      <possibility id="intlpos1">
        <conditions>
          <condition>
            <gameEntity ref="jozef"/>
            <state ref="idle"/>
          </condition>
          <condition>
            <gameEntity ref="avatar"/>
            <state ref="idle"/>
          </condition>
        </conditions>
        <actions>
          <changeState>
            <gameEntity ref="jozef"/>
            <state ref="speaking"/>
          </changeState>
          <changeState>
            <gameEntity ref="avatar"/>
            <state ref="speaking"/>
          </changeState>
          <executeDialog>1</executeDialog>
        </actions>
      </possibility>
    </possibilities>
  </interaction>
</interactions>

```



```
    </interaction>
  </interactions>
</definitions>
```

## 2.5 Reprezentácia dialógov

Podobným spôsobom ako pri interakciách je navrhnutá aj reprezentácia dialógov v hre. Dialóg predstavuje predpísaný rozhovor medzi postavou hráča a inou postavou v hre. Narozdiel od konverzácie, do dialógu hráč nemôže vstúpiť, môže iba sledovať jeho priebeh.

Pre každú scénu existuje jeden samostatný xml súbor, v ktorom sú zadané všetky texty dialógov, ktoré sa v nej odohrávajú. Každý dialóg má v scéne jedinečný identifikátor, podľa ktorého je možné ho zavolať z akcie **executeDialog**.

### 2.5.1.1 Základné dátové typy

Koreňový element **definitions** obsahuje 3 typy vnorených elementov:

- **gameEntities** – obsahuje všetky herné entity, ktoré sa budú v súbore využívať
- **interactionTypes** – definuje typy interakcií využité v súbore
- **dialogs** – definuje samotné dialógy

Každý z týchto typov môže obsahovať jeden alebo viac elementov príslušného typu (**gameEntities** – **gameEntity**, **interactionTypes** – **interactionType**, **conversations** - **conversation**).

**GameEntity** obsahuje

- atribút **id**, ktorý musí byť zadaný a jedinečný
- element **name** typu string, ktorý obsahuje meno entity

**InteractionType**

- atribút **id**, ktorý musí byť zadaný a jedinečný
- element **name** typu string – obsahuje názov interakcie

**Dialog**

- atribút **id**, ktorý musí byť zadaný a jedinečný
- vnorené elementy:
  - **interactionPossibility** – v atribúte **ref** obsahuje referenciu na **interactionPossibility** z dokumentu s interakciami pre príslušnú scénu
  - **lines** – obsahuje jeden alebo viac vnorených elementov typu **line**, ktoré obsahujú jednotlivé texty dialógov
  - **interaction** – obsahuje interakciu, ktorá sa má vykonať na konci dialógu

## Line

- obsahuje dva vnorené elementy:
  - **gameEntity** – v atribúte **ref** obsahuje referenciu na zadanú **gameEntity**, ktorá hovorí danú repliku
  - **text** – obsahuje text samotnej repliky
- jednotlivé elementy typu **line** nasledujú za sebou v tom poradí, v akom sa majú jednotlivé repliky zobrazit' v hre

## Interaction

- obsahuje vnorené elementy:
  - **source** – v atribúte **ref** obsahuje referenciu na zadanú **gameEntity**, ktorá je zdrojom finálnej interakcie
  - **destination** – v atribúte **ref** obsahuje referenciu na zadanú **gameEntity**, ktorá je cieľom finálnej interakcie
  - **interactionType** – v atribúte **ref** obsahuje referenciu na zadaný **interactionType**, ktorý určuje typ finálnej konverzácie

### 2.5.1.2 Príklad reprezentácie

```
<definitions>
  <gameEntities>
    <gameEntity id="jozef">
      <name>Jozef</name>
    </gameEntity>
    <gameEntity id="joker">
      <name>Joker</name>
    </gameEntity>
  </gameEntities>
  <interactionTypes>
    <interactionType id="finishConversation">
      <name>finishConversation</name>
    </interactionType>
  </interactionTypes>
  <dialogs>
    <dialog id="6">
      <interactionPossibility ref="int5pos1"/>
      <lines>
        <line>
          <gameEntity ref="joker"/>
          <text>Vieš čo dostaneš, keď dáš zmraziť
            jaternicu?</text>
        </line>
        <line>
          <gameEntity ref="jozef"/>
          <text>Nie.</text>
        </line>
        <line>
          <gameEntity ref="joker"/>
          <text>SNEHURKU! Hahahahaha chápeš? Sne .... Hurku.
            Akože sneh a hurka.... No nič, pokračuj v
            ceste.</text>
        </line>
        <line>
          <gameEntity ref="jozef"/>

```

```

        <text>Ach jaj.</text>
    </line>
</lines>
<interaction id="int3">
    <source ref="joker"/>
    <destination ref="jozef"/>
    <interactionType ref="finishConversation"/>
</interaction>
</dialog>
</dialogs>
</definitions>

```

## 2.6 Reprezentácia konverzácií

V prípade, že Avatar alebo iná postava v hre potrebuje komunikovať s hráčom formou dialógu je to umožnené pomocou akcie doConversation. Táto akcia vyhladá v príslušnom Xml súbore podľa id konverzácie jej koreň. Konverzácia očakáva od hráča hlasový vstup a chod dialógu sa upravuje podľa zachytených kľúčových slov.

Základné dátové typy

Existuje jeden samostatný xml súbor pre každú scénu v hre, ktorý obsahuje texty prislúchajúcich konverzácií identifikovaných jedinečným id číslom

### 2.6.1.1 Základné dátové typy

Koreňový element je opäť **definitions**, ktorý obsahuje 3 podtypy:

- **gameEntities** – obsahuje všetky herné entity, ktoré sa budú v súbore využívať
- **interactionTypes** – definuje typy interakcií využité v súbore
- **conversations** – definuje samotné konverzácie

Každý z týchto typov môže obsahovať niekoľko alebo viac elementov príslušného typu (**gameEntities** – **gameEntity**, **interactionTypes** – **interactionType**, **conversations** – **conversation**).

**GameEntity** obsahuje

- atribút **id**, ktorý musí byť zadaný a jedinečný
- element **name** typu string, ktorý obsahuje meno entity

**InteractionType**

- atribút **id**, ktorý musí byť zadaný a jedinečný
- element **name** typu string – obsahuje názov interakcie

**Conversation**

- atribút **id**, ktorý musí byť zadaný a jedinečný
- elementy:
  - **avatarSentences** – obsahuje všetky vety, ktoré môže/ vie avatar alebo iná postava, ktorá využíva konverzáciu v danom dialógu povedať
  - **avatarAdvices** – v prípade, že avatari dávajú radu v rámci konverzácie, nachádzajú sa v tomto elemente – kvôli zabezpečeniu náhodnému poradiu, ak sa v konverzácií nevyskytuje rada, musí existovať jeden pod element **advice**, ktorý je prázdny
  - **defaultResponse** – text postavy, ktorý je povedaný ak sa nenájde žiadna vyhovujúca veta **sentence** z **avatarSentences**

- **playerResponses** – zoznam možných / predpokladaných hráčových odpovedí
- **interaction** – interakcia vykonaná po skončení rozhovoru
- Všetky predpísané repliky (**avatarSentences**) postáv sú definované jedinečným indexovým číslom, sú číslované postupne od 0, 1, 2 .. n
  - atribút **id** elementu **sentence**, ktorý je dieťaťom **avatarSentences**.
- Na každú repliku môže byť definovaných niekoľko možných odpovedí (**playerResponse**), ktoré sa rozlišujú na základe kľúčových slov
  - element **playerResponse** obsahuje dva elementy:
    - **avatarSentence** - má atribút **ref**, ktorý obsahuje id **sentence**, na ktorú uvádzame možné odpovede
    - **possibleResponses** – zoznam kľúčových slov a možných odpovedí
- Po skončení dialógu je zavolaná interakcia
- Interakcia je určená zdrojom (**source**), cieľom (**destination**), a typom (**interactionType**). Zdroj je herná entita, ktorá interakciu vyvoláva a cieľ je herná entita, s ktorou zdroj chce interagovať.

**avatarSentence** obsahuje minimálne jeden a viac elementov **sentenceEntity**

**sentenceEntity** obsahuje:

- **sentence** – povinný element, práve jeden výskyt v **sentenceEntity**
  - obsahuje text vety a atribút **ref** je jej jedinečný index
- **waitForPlayerAnswer** – voliteľný element – 0 alebo 1 výskyt
  - v prípade, že má postava povedať za sebou viac viet, do tohto elementu sa zadá index nasledujúcej vety (nečaká sa na hráčovú odpoveď)
- **gameEntity** – voliteľný element – 0 alebo 1 výskyt
  - veta sa povie hlasom postavy, ktorej meno je zadané v atribúte **ref**
- **interaction** – voliteľný element – 0 alebo 1 výskyt
  - Po povedaní vety sa hneď vykoná akcia, ktorá sa vyhladá pomocou interakčného manažéra
  - Interakcia je určená zdrojom (**source**), cieľom (**destination**), a typom (**interactionType**). Zdroj je herná entita, ktorá interakciu vyvoláva a cieľ je herná entita, s ktorou zdroj chce interagovať
- **useTextInput** – voliteľný element – 0 alebo 1 výskyt
  - vždy sa od hráča očakáva hovorená odpoveď, v prípade, že element **useTextInput** obsahuje text **true**, namiesto hovorenej očakávame textový vstup – otvorí sa textové okno v hre
- **isFinal** – voliteľný element – 0 alebo 1 výskyt
  - v prípade, že táto veta ukončuje dialóg, obsahuje text **true**
- **advice** – voliteľný element – 0 alebo 1 výskyt
  - po povedaní tejto vety dá automaticky radu s daným indexom

**possibleResponses** obsahuje minimálne jeden a viac elementov **possibleResponse**

**possibleResponse** obsahuje:

- **keyWords** – povinný element, práve jeden výskyt v **possibleResponse**
  - zoznam kľúčových slov, ktoré očakávame v hráčovej odpovedi, musí byť v tvare zoznamu slov oddelených čiarkou a v základnom tvare slova

- v prípade, že je zoznam slov prázdny túto odpoveď berieme ako základnú (v prípade, že nenájde žiadnu zhodu), môže byť len jeden **possibleResponse** s prázdny zoznamom kľúčových slov
- **responseTarget** – povinný element, práve jeden výskyt v **possibleResponse**
  - atribút **ref** obsahuje id vety (elementu **sentence**), ktorú ma postava odpovedať
- **Advice** – voliteľný element – 0 alebo 1 výskyt
  - ak má na túto hráčovu vetu dať avatar odpoveď tak použijeme tento element s číslom rady **advice** z **avatarAdvices**
- **isFinal** – voliteľný element – 0 alebo 1 výskyt
  - v prípade, že táto odpoveď hráča ukončuje dialóg, obsahuje text **true**

### 2.6.1.2 Príklad reprezentácie

```

<definitions
  <gameEntities>
    <gameEntity id="jozef">
      <name>Jozef</name>
    </gameEntity>
    <gameEntity id="avatar">
      <name>Avatar</name>
    </gameEntity>
  </gameEntities>
  <interactionTypes>
    <interactionType id="finishConversation">
      <name>finishConversation</name>
    </interactionType>
    <interactionType id="openInventory">
      <name>Give Disk</name>
    </interactionType>
  </interactionTypes>
  <conversations>
    <conversation id="1">
      <avatarSentences>
        <sentenceEntity>
          <sentence id="0">Vidíš v miestnosti ešte niekoho?</sentence>
        </sentenceEntity>
        <sentenceEntity>
          <sentence id="1">Opíš nám ako vyzerajú.</sentence>
        </sentenceEntity>
        <sentenceEntity>
          <sentence id="2">Ako vyzerajú?</sentence>
        </sentenceEntity>
        <sentenceEntity>
          <sentence id="3">Správne</sentence>
          <waitForPlayerAnswer>4</waitForPlayerAnswer>
          <gameEntity ref="riddler"/>
          <interaction id="con2int1">
            <source ref="jozef"/>
            <destination ref="inventory"/>
            <interactionType ref="openInventory"/>
          </interaction>
        </sentenceEntity>
        <sentenceEntity>
          <sentence id="4"></sentence>
        </sentenceEntity>
      </avatarSentences>
    </conversation>
  </conversations>
</definitions>

```

```

        <isFinal>true</isFinal>
        <advice>1</advice>
    </sentenceEntity>
</avatarSentences>
<avatarAdvices>
    <advice id="avatar1-pos1">Podľa mňa by si sa mal ísť
        porozprávať s tou nízkou.</advice>
    <advice id="avatar2-pos1">Podľa mňa by si sa mal ísť
        porozprávať s tou vysokou.</advice>
</avatarAdvices>
<defaultResponse>
    <responseTarget ref="5"/>
    <isFinal>>false</isFinal>
</defaultResponse>
<playerResponses>
    <playerResponse>
        <avatarSentence ref="0"/>
        <possibleResponses>
            <possibleResponse>
                <keyWords></keyWords>
                <responseTarget ref="1"/>
            </possibleResponse>
            <possibleResponse>
                <keyWords>áno, vidieť, jasné, samozrejmosť
            </keyWords>
                <responseTarget ref="2"/>
            </possibleResponse>
            <possibleResponse>
                <keyWords>áno, vidieť, jasné, samozrejmosť, žena, ba
                    ba, dievča, slečna, osoba
            </keyWords>
                <responseTarget ref="1"/>
            </possibleResponse>
            <possibleResponse>
                <keyWords>nie, nevidieť, nikto, nezdať</keyWords>
                <responseTarget ref="3"/>
            </possibleResponse>
        </possibleResponses>
    </playerResponse>
</playerResponses>
<interaction id="int1">
    <source ref="avatar"/>
    <destination ref="jozef"/>
    <interactionType ref="finishConversation"/>
</interaction>
</conversation>
</conversations>
</definitions>

```

## 2.7 Implementácia

Hru sme sa rozhodli implementovať s použitím herného enginu Unity, ktorý zjednodušuje proces grafického vytvorenia scény a jej integrácie do celkového systému. Unity umožňuje vytvárať herné objekty, ktoré môžu byť graficky reprezentované a pridelovať im jednotlivé komponenty.

Jedným z týchto komponentov je Script, ktorý umožňuje pomocou programátorského kódu riadiť priebeh a správanie aplikácie. Scripty je možné v Unity písať v jazykoch JavaScript alebo C#, pričom my sme sa rozhodli implementovať aplikáciu v jazyku C#.

Unity zároveň poskytuje možnosť používania rozšírení v podobe knižníc, ktoré je možné importovať do projektu. V našej hre využívame 3 rozšírenia a to **Adventure Creator**, ktoré pomáha pracovať s pohybom po 2D scéne a **Puppet2D**, kde je možné vytvoriť animáciu 2D postavy pomocou vytvorenia kostry. Integrovali sme takisto plugin **UnityTestTools**, ktorý poskytuje framework na zjednodušenie procesu testovania.

### 2.7.1 Pohyb po 2D scéne

Ako prvý používateľský príbeh sme sa rozhodli implementovať pohyb hráča po scéne.

Hlavný scenár pohybu hráča po scéne sa skladá z nasledujúcich krokov:

1. Zobrazí sa scéna, ktorá je reprezentovaná dvojrozmerným obrázkom
2. Zobrazí sa obrázok hlavnej postavy
3. Hráč pomocou kurzora myši kliknutím zvolí miesto, kam sa má postava na scéne presunúť
4. Postava sa presunie na zadané miesto v priestore, pričom sa zobrazuje animácia postavy v danom smere a simuluje sa pohyb v priestore zmenšovaním postavy s ohľadom na perspektívu

Ďalej sme identifikovali nasledujúce zvláštne prípady:

- Ak hráč zvolí možnosť presunu na plochu, na ktorú nie je možné v hre pristúpiť (napríklad stena, obloha alebo voda), postava sa musí pohybovať vo zvolenom smere a musí zastaviť v poslednom bode, ktorý sa nachádza na ploche, na ktorú je možné pristúpiť.
- Ak sa uprostred scény nachádza objekt reprezentovaný ako 2D obrázok, ktorý je možné obísť spredu aj zozadu, postava musí mať túto možnosť a musí byť možné ju vykresliť buď pred objektom alebo za objektom v závislosti od jej polohy.

Tieto požiadavky sa podarilo naplniť vďaka použitiu rozšírenia **Adventure Creator**. Toto rozšírenie umožňuje zdefinovať 2D scénu skladajúcu sa z viacerých dvojrozmerných obrázkov (sprite-ov), pričom každému je možné číselne zdefinovať poradie v akom je zobrazený. Na tejto scéne je možné zdefinovať komponent 2D polygónu, ktorý určuje plochu, po ktorej sa môže postava v scéne pohybovať. Táto plocha sa nazýva NavMesh. Do tejto plochy je v prípade potreby ďalej možné pridávať ďalšie polygóny, ktoré určujú plochy, na ktoré hráč nemôže vstúpiť (tzv diery). Ak má obrázok postavy zdefinovaný pivot na spodku obrázku (tam, kde má nohy), je mu umožnené vstúpiť nohami presne iba po okraje zdefinovanej zóny.

Ďalším dôležitým prvkom pri pohybe postavy po mape je script Sorting map, ktorý umožňuje scénu rozdeliť na oblasti, v ktorých bude zobrazenie postavy nejakým spôsobom zmenené od štandardného zobrazenia. Ak sa teda v scéne nachádza objekt, ktorý musí postava byť schopná obísť, je teda možné scénu rozdeliť na dve časti a to za objektom a pred objektom. Potom je možné zdefinovať aké bude v danej časti poradie vykresľovania vrstiev a vďaka tomu je možné dosiahnuť efekt obchádzania daného predmetu (simulácia priestoru v 2D scéne).

Ďalej je možné pomocou komponentu *Sorting map* zadefinovať veľkosť hráča v koncových bodoch zadefinovaných oblastí. To je možné využiť napríklad vtedy, ak je hráčovi umožnený pohyb po miestnosti, avšak je potrebné simulovať perspektívu kamery tým spôsobom, že sa postava v „dialke“ zmenšuje. V tomto prípade je možné na konci miestnosti zadefinovať veľkosť postavy iba na zlomok pôvodnej veľkosti a na začiatku miestnosti (najbližšie ku kamere) zadefinovať pôvodnú veľkosť. *Sorting map* potom pomocou lineárnej interpolácie prepočíta veľkosť postavy v jednotlivých bodoch „priestoru“, čo simuluje prechádzanie po miestnosti a vzdďľovanie od kamery.

Hlavná postava je do miestnosti pridaná ako herný objekt s komponentom *Player*, čo je tiež *Script* poskytnutý z rozšírenia *Adventure Creator*. Tento komponent umožňuje zadefinovať základné parametre hlavnej postavy ako napríklad rýchlosť pohybu, zastavenia, otáčania, behu a ďalej príslušné animácie, ktoré sa majú pri zobrazovaní pohybu používať. V základnom režime je možné poskytnúť 4 typy animácie a to

- Idle – používa sa, keď postava stojí na mieste,
- Walk – používa sa, keď sa postava prechádza,
- Run – používa sa, keď postava beží
- Talk – používa sa, keď postava rozpráva.

### 2.7.2 Animácia postavy

V rámci pohybu po scéne, je dôležitý aspekt aj animácia postavy, ktorá musí pôsobiť autenticky. Na implementáciu sme si po prvotných pokusoch vybrali doplnok do Unity nazvaný *Puppet2D*. Tento nástroj umožňuje definovať kostru vytvorenému obrázku a po spojení obrázka s touto kostrou je možné vytvárať pózy, z ktorých sa skladá animácia.

Proces animácie pozostáva z nasledujúcich krokov:

1. Importovanie kresieb do Unity a ich vhodné umiestnenie na vytvorenie východiskovej pózy postavy
2. Vytvorenie kostí reprezentujúcich hlavné časti tela a ich spojenie pomocou kľbov. Hierarchia kostí (rodičia a deti) je dôležitá pri neskoršom animovaní
3. Určenie kontrolných bodov využívajúcich princípy IK (inverse kinematics), bodov na otáčanie a bodov na posun celej kostry
4. Prepojenie kresieb a kostry aby sa mohla grafika postavy pohybovať podľa polohy jednotlivých kostí. Podľa situácie je vhodné použiť jednoduché napojenie na rodiča (kde však nenastáva deformácia a teda zmena obrazovej predlohy) alebo konverziu na sieť bodov, ktorá túto možnosť poskytuje (po upravení váh jednotlivých bodov aby deformácia pôsobila realisticky)
5. Nastavenie vhodných pozícií kostry pre jednotlivé kľúčové polohy animácie
6. Vo finálnom kroku sa vyladia možné nedostatky a interpoláciou sa vytvoria medzi-pozície tvoriace kompletnú animáciu

### 2.7.3 Interakcia s objektami

Objekty, ktoré sa vyskytujú v scéne a sú schopné vykonávať interakciu definovanú v stavových diagramoch majú na sebe triedu *GameEntity*. Táto trieda nám drží informácie o možných interakciách, znalostiach a stavoch konkrétneho objektu. Po kliknutí na takýto objekt v hre sa automaticky zavolá metóda *onClick*. V tejto metóde sa pomocou manažéra akcií zistia akcie, ktoré sa majú vykonať pre daný objekt, v danom stave, s danými znalosťami a *ActionInterpreter* ich následne preloží do reálnej akcie v hre.



#### 2.7.3.1 Manažér interakcií

Manažér interakcií je modul, ktorý zabezpečuje prácu s XML súborom obsahujúcim informácie o interakciách. Jeho základná funkcionalita zahŕňa 2 základné metódy a to vyhľadanie interakcie a vrátenie akcií pre interakciu.

Vyhľadanie interakcie má na vstupe 2 objekty, ktoré chcú interagovať, typ ich interakcie a scénu, v ktorej sa nachádzajú. Úlohou tejto metódy je vyhľadať v dokumente interakcií vyhovujúcu interakciu a vrátiť ju naspäť spolu so zoznamom objektov, ktoré budú potrebné na rozlíšenie akcií, ktoré sa majú vykonať.

Metóda na vrátenie akcií má na vstupe interakciu a zoznam objektov, ktoré sú potrebné k rozlíšeniu akcií pri danej interakcii vrátane ich aktuálnych stavov a znalostí. Na základe týchto stavov a znalostí následne vyhľadá takú možnosť interakcie, ktorá vyhovuje aktuálnym podmienkam hry.

Na prehľadávanie XML dokumentu je použitý jazyk XPath pomocou ktorého je možné získavať konkrétne elementy alebo množiny elementov.

#### 2.7.4 Tvorba grafiky

Postup tvorby grafických prvkov hry sa začína nakreslením podkladových skíc. Zo začiatku boli skice kreslené na papier a následne vo forme fotografií prenesené do digitálnej podoby na ďalšie spracovanie. Tento postup bol však zdĺhavý a z viacerých dôvodov nepraktický. Po novom sú skice vytvárané za použitia grafického tabletu a Adobe Photoshop, čo výrazne zjednodušuje prípadnú úpravu skíc.

Podkladové skice sa následne prekreslia pomocou programu RealDraw. Tento program nám umožňuje rýchlo a jednoducho prerobiť skice do ich finálnej podoby. RealDraw, na rozdiel od ostatných podobných editorov, poskytuje možnosť pracovať s vektorovou aj rastrovou grafikou zároveň. Keďže v rámci vytvárania grafických prvkov hry je potrebná práca s textúrami, táto vlastnosť pre nás predstavuje veľkú výhodu. Po dokončení finálnej podoby grafického prvku umožňuje tento nástroj exportovať obrázky pomocou funkcie „megarender“ vo všetkých štandardných formátoch, a tiež rôznych rozlíšeniach. Pre naše účely využívame export do .png formátu.

##### Grafika scén

Pri kreslení scény sa najskôr nakreslí podlaha s múrmi a následne sa pridávajú všetky ostatné prvky tak aby boli generické a mohli sa znova použiť v iných scénach (napr. dvere, svietniky, okná, stĺpy, stoly..). Na nakreslené objekty sa pridávajú efekty a zaoblenia tak aby vyzerali krajšie, no zároveň treba dodržať jednotný štýl v celej hre.

Ako prvá bola nakreslená scéna 3 a prešla viacerými iteráciami vylepšovania.



*Obrázok 13: scéna 3 00*

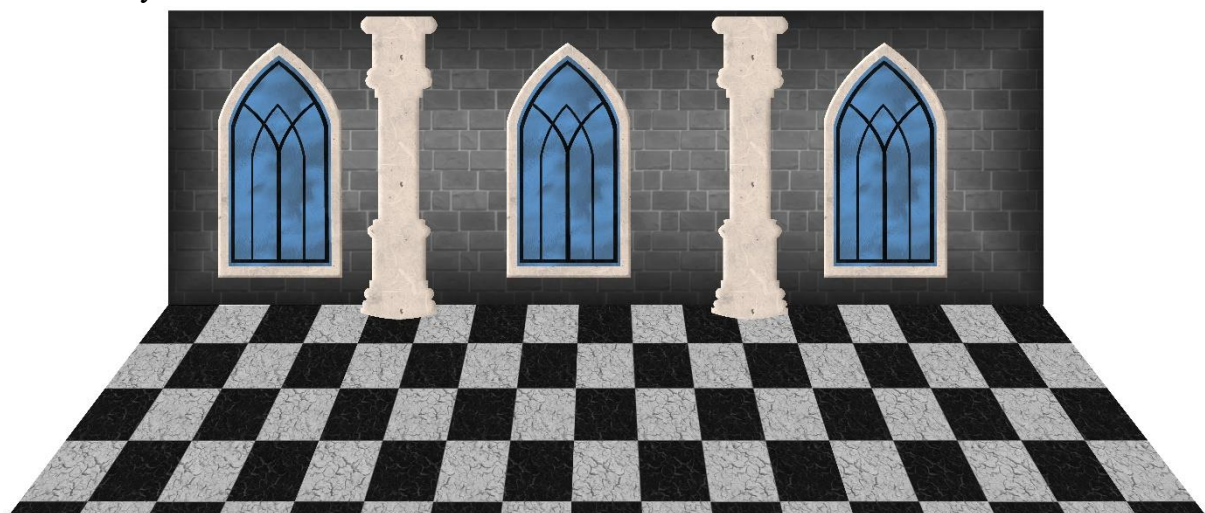


*Obrázok 14: scéna 3 01*



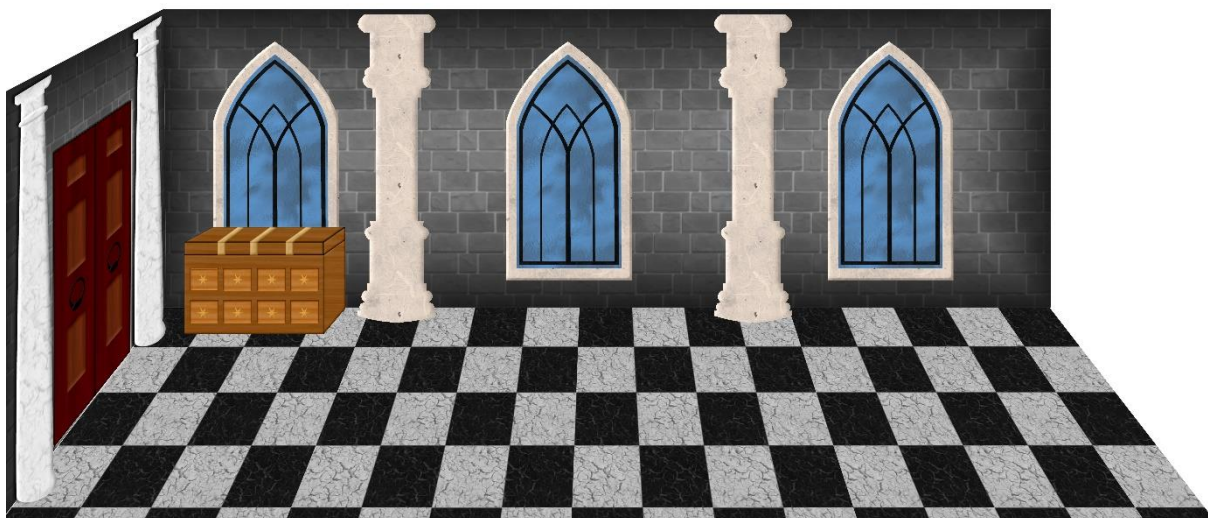
*Obrázok 15: scéna 3 02*

Ako druhá bola nakreslená scéna 2 ktorá je len prechodná chodba preto nemusí mať bočné steny.



*Obrázok 16: scéna 2 00*

Scéna 1 kde hráč začne prvý krát ovládať hru.



Obrázok 17: scéna 1 00



Obrázok 18: scéna 1 01



Obrázok 19: scéna 1 02

V rámci implementovanej minihry bol použitý osobitný, špecifický typ grafiky výzorne odlišný od zvyšku hry. Tvorba tejto grafiky je podrobne opísaná v kapitole, ktorá sa minihrou zaoberá, no na jej spustenie bolo potrebné navrhnuť nejaký grafický prvok, ktorý upúta hráčovú pozornosť. Ten je zobrazený na obrázku číslo X.



Obrázok 20: prístroj na spúšťanie minihry

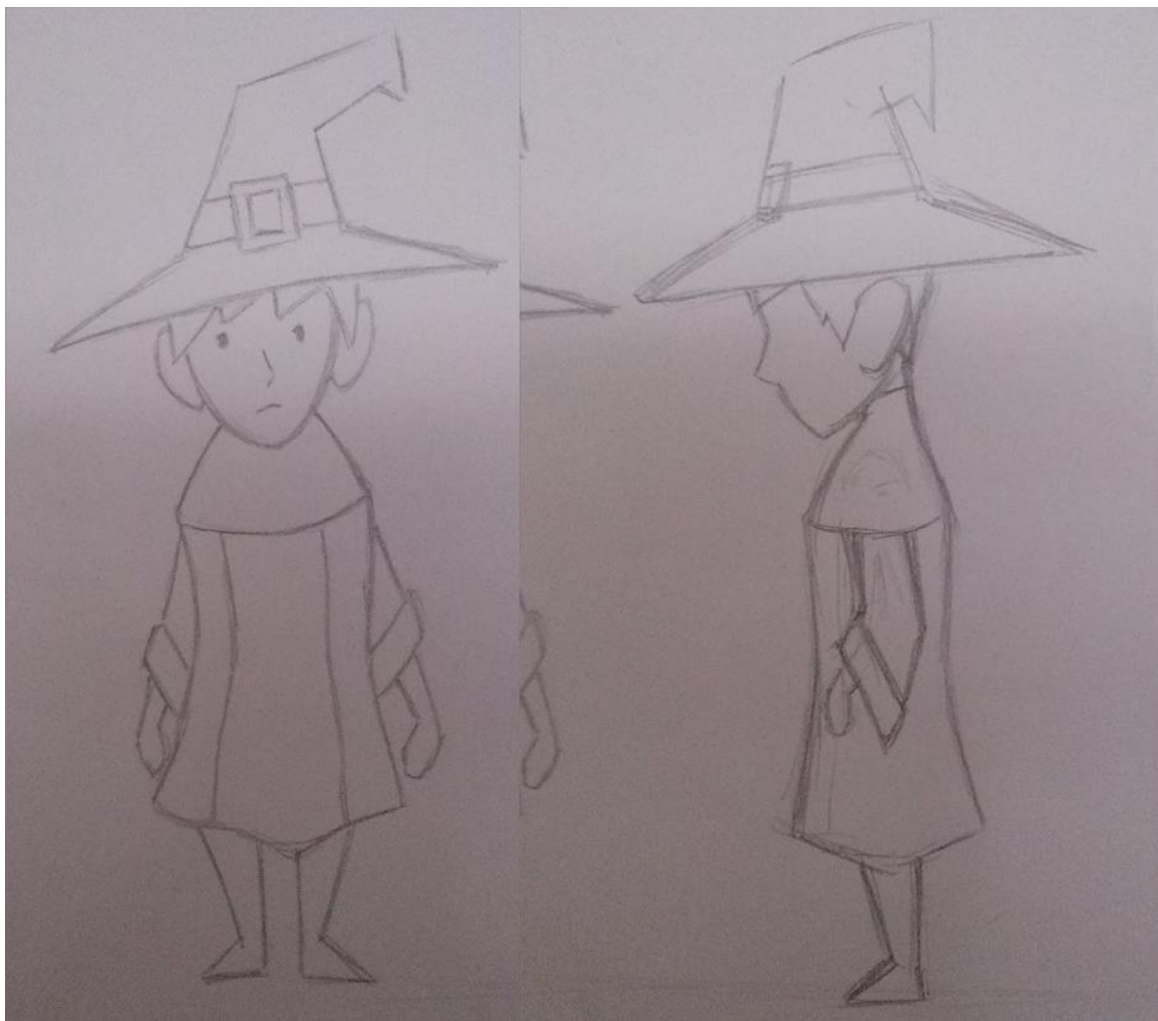
#### 2.7.4.1 Grafika postáv

Dizajn grafickej reprezentácie postáv sa neustále vyvíja. Hlavná postava hry už prešla mnohými iteráciami a aj napriek tomu, že aktuálna verzia je vyhlásená za finálnu, nie je možné vylúčiť, že v budúcnosti budú nasledovať ďalšie zmeny. Na nasledujúcich obrázkoch, ktoré predstavujú mnohé iterácie hlavnej postavy je možné vidieť ako úroveň detailov postupne rástla. Taktiež je možné pozorovať zmenu farebnej schémy, odstránenie výrazných obrysových čiar a prechod od používania čistej farby na jej kombinovanie s textúrami, rovnako ako pri vytváraní herných scén. Dôvod pre využitie textúr je vizuálna konzistentnosť hry aby postava vizuálne ladila so svojím okolím. Výrazné rozdiely v jednotlivých iteráciách sú spôsobené aj tým, že na dizajne hlavnej postavy sa spoiatku podieľali viacerí členovia tímu.

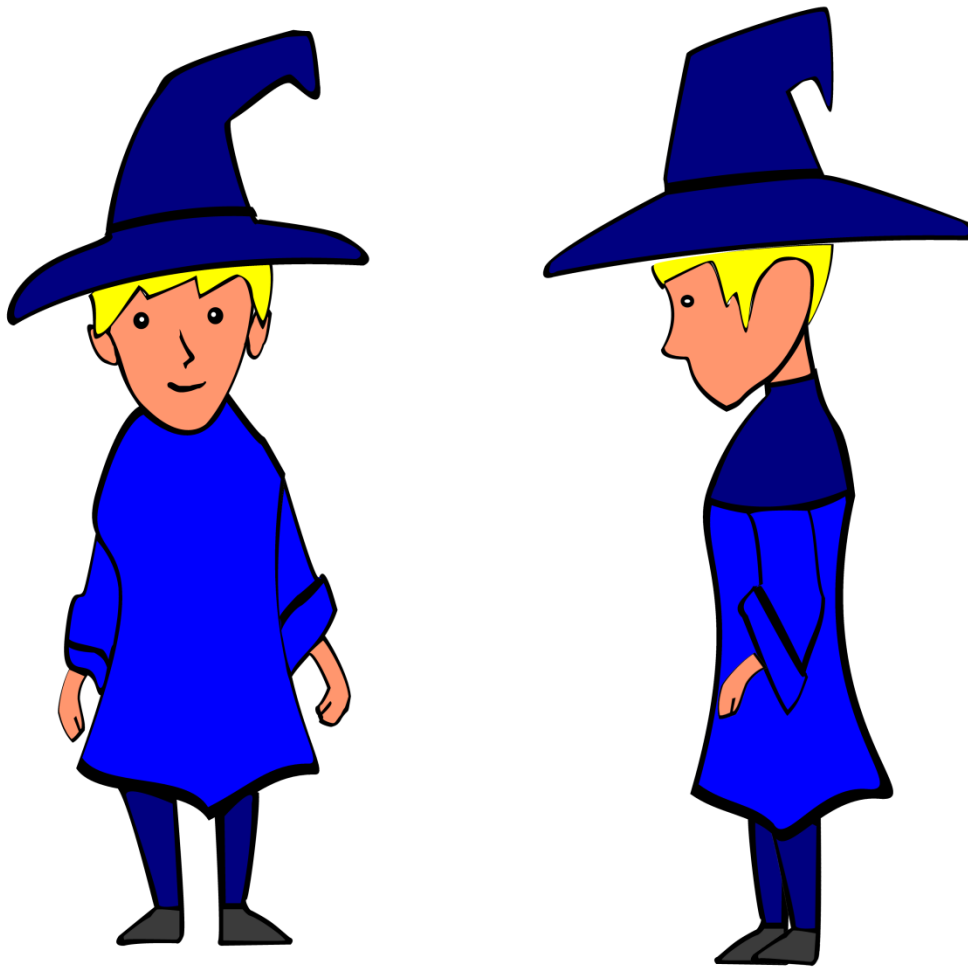
Jednoduchosť pôvodnej podkladovej skice, ktorú je možné vidieť na obr. č. 20, je spôsobená aj tým, že pôvodne sme hru plánovali robiť za použitia 3D scény a modelov. Prvotná skica teda nemala slúžiť ako podklad pre následnú kresbu realizovanú v RealDraw ale pre jej 3D model. Od tejto možnosti sme sa však po čase odklonili a bolo nutné prepracovať návrh aby spĺňal aby bol vhodný aj na implementáciu v 2D priestore.

Vzhľadom na to, že postavu je nutné animovať, nie je žiadúce ju exportovať z RealDraw ako celok vo forme jedného .png obrázku. Namiesto toho je potrebné vytvoriť osobitné súbory pre všetky časti tela či odevu, ktoré sa budú po animáciách hýbať nezávisle od ostatných, respektíve inak ako ostatné. Časti hlavnej postavy pripravené na animáciu sú zobrazené na obr. č. 24. Kvôli animáciám tiež treba nakresliť postavu zo všetkých strán (presne z troch keďže z boku je postava rovnaká sprava aj zľava). Rôzne iterácie grafickej

reprezentácie hlavnej postavy sú zobrazené na obrázkoch 21, 22 a 23. Na obrázku 25 je zobrazená záporná postava hry, Vlado, a na obrázku 26 zase avatar, hráčov pomocník. Grafika oboch týchto postáv je pripravená na animáciu.



*Obrázok 21: Prvá skica hlavnej postavy*



*Obrázok 22: Grafická reprezentácia hlavnej postavy 1.0*



*Obrázok 23: Grafická reprezentácia hlavnej postavy, iterácie 3.0 spredu, 5.0 z boku a 1.0 zozadu*

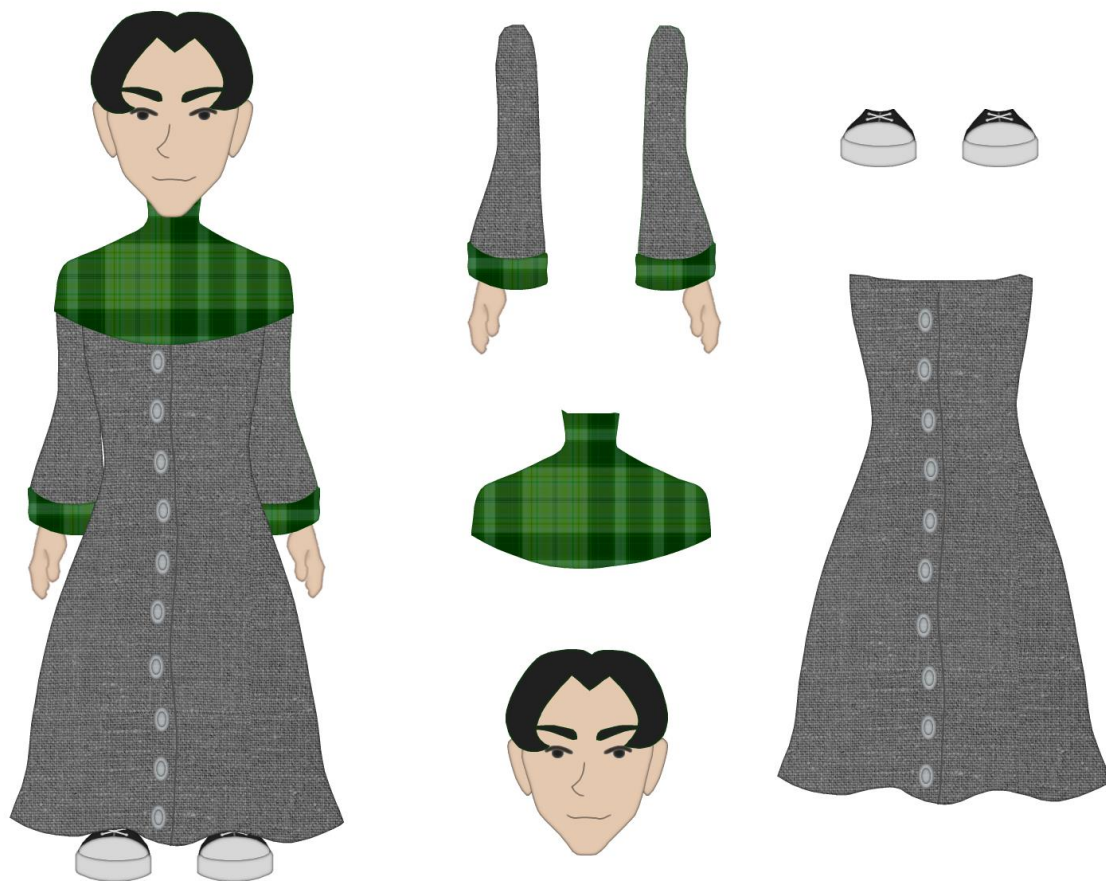




*Obrázok 24: Grafická reprezentácia hlavnej postavy, iterácie 4.0 spredu, 6.0 z boku a 2.0 zozadu*



*Obrázok 25: Grafická reprezentácia hlavnej postavy, iterácia 3.0 spredu, pripravená na animáciu*



*Obrázok 26: Grafická reprezentácia zápornej postavy, iterácia 2.0 spredu, pripravená na animáciu*



Obrázok 27: Grafická reprezentácia avatara, iterácia 1.0 spredu, pripravená na animáciu

### 2.7.5 Výpis na obrazovku

Výpis na obrazovku zabezpečuje trieda „WriteToScreen“. Funkcia Start() vyhledá a vytvorí referenciu na GameObject dialog ktorý je typu UI Text a zabezpečuje výpis textu na obrazovku. V objekte dialog sa môžu nastaviť parametre ako font, veľkosť, farba, umiestnenie a iné. Funkcia update sleduje či užívateľ stlačil stredné tlačidlo myši, ak áno schová text vypísaný na obrazovke. Funkcia setText(string text) zobrazí text ktorý jej prišiel ako parameter na obrazovku.

```
using UnityEngine;
```

```
using UnityEngine.UI;
```

```
using System.Collections;
```

```
public class WriteToScreen : MonoBehaviour {
```

```
    //textField for displayng text
```

```
    public Text objectText;
```

```
    public GameObject dialog;
```

```

// Use this for initialization
void Start () {
    dialog = GameObject.Find("dialog");
    objectText = dialog.GetComponent<Text>();
}

// Update is called once per frame
void Update () {
    if (Input.GetMouseButton(2)) {
        objectText.enabled = false;
    }
}

public void setText(string text) {
    objectText.enabled = true;
    objectText.text = text;
}
}

```

V každom skripte kde je potrebný výpis textu na obrazovku sa len vytvorí referencia na túto triedu.

```
public WriteToScreen write;
```

Vo funkcií Start() daného skriptu sa inicializuje

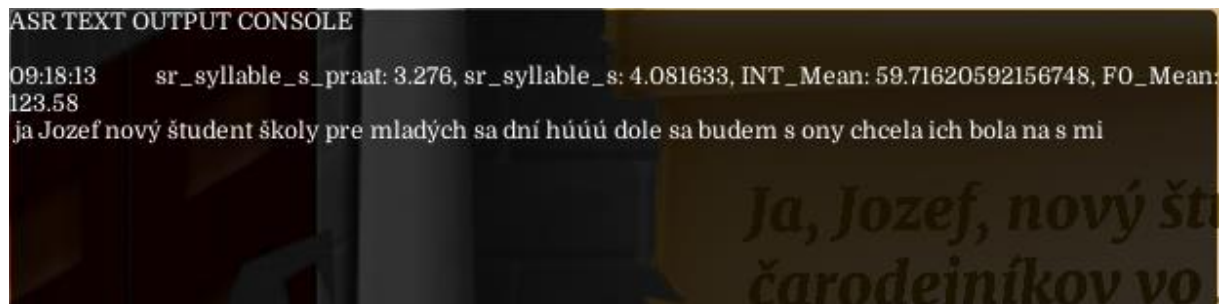
```
write = GetComponent<WriteToScreen>();
```

a potom sa už len jednoducho prístupuje k funkcii setText(text)

```
write.setText(„Text na vypísanie“);
```

## 2.7.1 Konzola pre ladenie ASR modulu

Na objednávku zákazníka bol výpis na obrazovku znovupoužitý na vytvorenie funkcionality vypisovania priebežných výstupov z ASR počas hry do konzoly vrámci rozhrania hry, za účelom možnosti ladenia ASR modulu za behu.



Obrázok 28 Ukážka konzoly pre ladenie ASR

V konzole je zobrazený čas návratu hodnôt z ASR modulu, namerané parametre reči a rozpoznávaný text z hlasu.

Na základe parametra nastaveného switchom v nastaveniach sa pri každom volaní funkcie WaitForResponseGetText vypíšu, alebo nevypíšu parametre a text do ASR konzoly pomocou funkcie ToConsole.

```

if (SettingsManager.DebugConsole())
    write.ToConsole(DateTime.Now.ToString("HH:mm:ss \t\t ")
        + GameController.PlayerLastReaction + " " + textOfSpeech + "\n");

```

Funkcia ToConsole udržiava pole riadkov v zobraziteľných v konzole a keď dosiahne limit (10 riadkov) nahrádza najstaršie riadky najnovšími.

```

public void ToConsole(string newLine)
{

```

```

panelConsole.SetActive(true);
lines.Dequeue();
lines.Enqueue(newLine);
consoleText.text = "";
foreach (var line in lines.ToList())
{
    consoleText.text = line + consoleText.text;
}
consoleText.text = "ASR TEXT OUTPUT CONSOLE\n\n" + consoleText.text;
}

```

Pred výpisom sa konzola aktivuje aby v prípade že sa jedná o prvý výpis v rámci dialógu bola zobrazená.

## 2.7.2 Inventár



Obrázok 29 Ukážka inventáru a nápovedy

Inventár v hre pozostáva z 4 kľúčových skriptov.

Inventor, Inventory, ItemDatabase a InventoryItem.

InventoryItem je trieda popisujúca objekty z hry. Každý objekt obsahuje parametre:

```

public string itemName;
public int itemID;
public string itemDesc;
public Texture2D itemIcon;
public ItemType itemType;

public bool sellable;
public int price;

```

ItemDatabase obsahuje List InventoryItemov.

Inventor je trieda obsahujúca frontend inventára. Obsahuje metódy:

`public void OnClickInventory()` - zobrazí / skryje inventár.

`void Update()` - sleduje či bolo stlačené tlačidlo na zobrazenie / skrytie inventára. Taktiež sleduje či bolo stlačené ľavé tlačidlo myši.

Vykresľovanie inventára pozostáva zo zobrazenia pozadia, vykreslenia jednotlivých slotov a vykreslenia objektov. Na vykresľovanie sa používa metóda `GUI.Box()` a `GUI.DrawTexture`. Obe metódy používajú statické súradnice `Rect`.

Inventár umožňuje presúvanie objektov, ťahanie objektov, prehadzovanie objektov. Po nesprávnom použití objektu sa objekt vráti do inventára.

Trieda `Inventory` obsahuje funkcie na pridávanie, odstraňovanie, zisťovanie či inventár obsahuje daný objekt.

`public void AddItem(int id)` - funguje tak, že sa napíše id objektu, ktorý chceme pridať a z `ItemDatabase` sa vyberie objekt s daným id a pridá sa do inventára.

### 2.7.3 Obchod

Obchod je v hre implementovaný ako nadstavba nad herným inventárom a umožňuje nákup dostupných vecí a zobrazenie hernej meny.

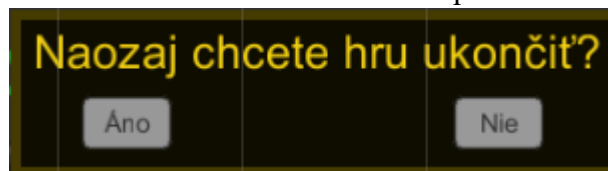
V obchode sa vykreslia všetky predmety z databázy, ktoré majú nastavený atribút `sellable` na `true` a v ich popise sa zobrazí nákupná cena. Ak hráč potrebuje niektorý z predmetov, tak si ho môže kliknutím naň kúpiť. Následne sa skontroluje či má hráč dostatok peňazí a výsledok je zvukovo indikovaný, v prípade úspechu sa konkrétny predmet presunie hráčovi do inventára.



Obrázok 30 Ukážka obchodu a nápovedy

### 2.7.4 Ukončenie hry

Po kliknutí na tlačidlo krížika v pravej hornej časti obrazovky vyskočí okno kde môžeme potvrdiť ukončenie hry a zavolať sa funkcia `Application.Quit()`; alebo môžeme okno vypnúť a pokračovať ďalej v hre. Tlačidlo krížika na klik zobrazí potvrdzovacie okno o vypnutí hry.



Obrázok 31 Ukončovacie okno

Kliknutím na `Nie` sa okno znova schová a hra pokračuje. Kliknutím na `Áno` sa zavolať ukončenie hry. Ukončenie funguje len v exportovanej verzii, nie v editore.

### 2.7.5 Animácia dverí

Animácia dverí je ovládaná skriptom „`DvereAnimacia.cs`“. Tu sa overujú 2 podmienky, či používateľ klikol na dvere, čiže sa od nich očakáva interakcia. Ďalej či prišiel zvukový záznam zo SAV s rečou. Ak sú obe dve podmienky splnené zapne sa animácia reči. Po dohratí nahrávky sa animácia vypne a stavy sa vrátia do stavu idle.

## 2.7.6 Hlasové moduly SAV

Dôležitou časťou nášho projektu je využívanie hlasového modulu. SAV nám poskytuje rozhranie s dvoma webovými službami na TTS (text to speech) a ASR (automatic speech recognition).

### 2.7.6.1 TTS

Táto služba slúži na preklad textu do hovoreného slova. Momentálne používame dve verzie tejto služby. Jedna je zameraná len na preklad do hovorenej reči podľa konkrétnych parametrov. Druhá poskytuje aj automatické prispôbovanie hráčovi.

Posielajú sa niekoľké parametre:

#### Prvá verzia

Na dohodnutej url adrese komunikujeme so SAV pomocou formulára, ktorý obsahuje:

- voice: výber z ponuky hlasov
- text: text na preklad
- link: logická hodnota, či sa má odpoveď presmerovať alebo má vrátiť link na túto adresu
- rate: percentuálna hodnota rýchlosti reči
- pitch: percentuálna hodnota výšky hlasu
- volume: percentuálna hodnota hlasitosti

Odpoveďou je zvuk so vzorkovacou frekvenciou 16000Hz alebo 48000Hz, ktorý sa následne prehrá. Funkcia *BuildWWWRequestSpeak* slúži na vytvorenie webového formulára, ktorý sa posielajú na SAV server na spracovanie.

```
private WWW BuildWWWRequestSpeak(string voice, string text, int rate, int pitch, int
volume)
{
    var form = new WWWForm();
    form.AddField("voice", voice);
    form.AddField("text", text);
    form.AddField("Link", "false");

    var speed = rate.ToString(CultureInfo.CurrentCulture) + „%“;
    var intensity = volume.intensity.ToString(CultureInfo.CurrentCulture) + „%“;
    var pitch = pitch.pitch.ToString(CultureInfo.CurrentCulture) + „%“;
    form.AddField("rate", speed);
    form.AddField("pitch", pitch);
    form.AddField("volume", intensity);
    return new WWW(UrlBase + UrlSpeak, form);
}
```

#### Druhá verzia

Poskytuje možnosť prispôbenia hlasu na základe údajov o základných parametroch reči – nameranej na začiatku hry a posledných nameraných hodnotách.

Na dohodnutej url adrese komunikujeme so SAV pomocou formulára, ktorý obsahuje:

- voice: výber z ponuky hlasov
- text: text na preklad
- link: logická hodnota, či sa má odpoveď presmerovať alebo má vrátiť link na túto adresu



- rate\_n: presná hodnota v slabikách za sekundu syl/s základnej rýchlosti reči hráča
- pitch\_n: presná hodnota v Hz výšky hlasu hráča
- volume\_n: presná hodnota v dB hlasitosti reči hráča
- rate: presná hodnota v syl/s poslednej nameranej rýchlosti reči hráča
- pitch: presná hodnota v Hz poslednej nameranej rýchlosti reči hráča
- volume: presná hodnota v dB poslednej nameranej rýchlosti reči hráča
- mode: mód prispôsobovania – priame prispôsobovanie, neprispôsobovanie a základ

Máme funkciu, ktorá sa volá rovnako ako pre prvú verziu TTS, ale má iné parametre a slúži na vytvorenie formulára pre prispôsobovaný hlas.

```
private WWW BuildWWWRequestSpeak(string voice, string text, bool adapt)
{
    var form = new WWWForm();
    form.AddField("voice", voice);
    form.AddField("text", text);
    form.AddField("Link", "false");

    var speed = Last.speed.ToString();
    var intensity = Last.intensity.ToString();
    var pitch = Last.pitch.ToString();
    form.AddField("rate", speed);
    form.AddField("pitch", pitch);
    form.AddField("volume", intensity);
    var rn = Oath.speed.ToString();
    var pn = Oath.pitch.ToString();
    var vn = Oath.intensity.ToString();
    form.AddField("rate_n", rn);
    form.AddField("pitch_n", pn);
    form.AddField("volume_n", vn);
    var mode = adapt
        ? StringEnum.GetStringValue(Strings.Strings.SavModes.adapt)
        : StringEnum.GetStringValue(Strings.Strings.SavModes.none);
    form.AddField("mode", mode);
    return new WWW(UrlBase + UrlSpeak, form);
}
```

#### 2.7.6.2 ASR

Opačný hlasový modul ako TTS je ASR a jeho úlohou je hovorený text – nahraný zvuk preložiť čo najlepšie do textu. Táto služba prijíma zvukový súbor s hovoreným slovom a parameter pohlavia. V prvých iteráciách tento parameter chýbal, avšak postupným testovaním sa overilo, že keď SAV poskytneme pohlavie hráča je rozpoznanie textu presnejšie.

Opäť máme pripravenú funkciu, ktorá pre danú službu poskladá formulár z bytového poľa so zvukom.

```
public void TranslateSpeechToText(byte[] speech)
{
    var form = new WWWForm();
    form.AddBinaryData("audio", speech, "speech.wav", "audio/wav");
    form.AddField("gender", Gender);

    var request = new WWW(UrlBase + UrlWrite, form);
    StartCoroutine(WaitForResponseGetText(request));
}
```

}

Odpoveďou tejto služby je Xml súbor s nameranými parametrami reči a preloženým textom. Obsahuje parametre ako:

- Výška hlasu v Hz
- Hlasitosť / intenzita v dB
- Počet hlások za sekundu
- Počet slabík za sekundu
- Počet slov za sekundu
- Počet hlások celkovo
- Počet slabík celkovo
- Počet slov celkovo
- Text

Z týchto hodnôt je pre nás zaujímavý samotný text, výška, intenzita hlasu a počet slabík za sekundu čo nám udáva rýchlosť reči pri ďalej práci s prispôbovaním.

### 2.7.7 Čítanie prísahy pre zistenie základných parametrov reči

V našom projekte je dôležité prispôbovanie reči avatarov, ktorí dávajú rady hráčovi. Keďže každý hráč rozpráva iným spôsobom je potrebné na začiatku nakalibrovať základné parametre reči. Po niekoľkých konzultáciách s externým zadávateľom a pracovníkom SAV sme sa dohodli, že na začiatku každého spustenia hry je hráč povinný prečítať text. Aby nám to sedelo do príbehu, text je prísaha študenta, ktorý nastupuje do školy.

Hráč je vyzvaný aby text čítal po častiach, po odsekom alebo vetách, aby sme mali niekoľko meraní. Po každom prečítaní úseku textu pošleme tento text to ASR, ktorý nám vráti parametre reči. Postupne si ukladáme všetky namerané hodnoty rýchlosti, výšky a hlasitosti.

Keď hráč dokončí čítanie z nameraných hodnôt odstránime minimálnu a maximálnu hodnotu a zo zvyšných vypočítame priemer. Takto dostaneme tri parametre reči, ktoré vyjadrujú základné rozprávanie hráča.

Následne počas hry porovnávame tieto prvé namerané hodnoty s aktuálnymi hodnotami. Ak hráč počas hry začne rozprávať rýchlejšie, prispôbený avatar tiež hovorí jeho repliky rýchlejšie. To platí pre všetky ostatné parametre.

### 2.7.8 Správanie vtipkárky

V hre sa vyskytujú postavy so špeciálnym správaním. Jednou z nich je aj vtipkárka. Úlohou tejto postavy je pobaviť hráča rozprávaním vtipov.

Keď hráč v hre narazí na vtipkárku, môže sa jej prihovoriť a ona mu povie vtip. Prebieha to nasledovne:

- Hráč klikne na postavu, hra deteguje akciu a manažér interakcií identifikuje akciu sayJoke
- Xml parser následne vyhladá všetky vtipy dostupné pre danú scénu uložené v xml súbore spolu s ostatnými dialógmi (obsahuje atribút type="joke")
- Skontrolujeme stav hádankárky
  - Ak ešte beží časomiera 30 sekúnd: pozri sa na počet doteraz povedaných vtipov
  - Ak čas už vypršal: nastav počítanie času a počtu povedaných vtipov na 0

- Ak hádankárka nepovedala viac ako 3 vtipy, je ochotná povedať vtip, ktorý doteraz nepovedala z jej zoznamu – postupuje postupne v poradí v akom boli v xml súbore

Obmedzujeme počet povedaných vtipov za sebou na 3 z dôvodu, aby sa nám hráč nezasekol len pri hádankárke a počúval vtipy. Avšak jej počítadlo vynulujeme po čase 30 sekúnd, aby mal hráč opäť možnosť vypočítať si ďalšie vtipy a nenadobudol pocit, že hra pozná len tie prvé tri vtipy.

Nižšie vidíme ukážku xml, ktorá obsahuje vtip.

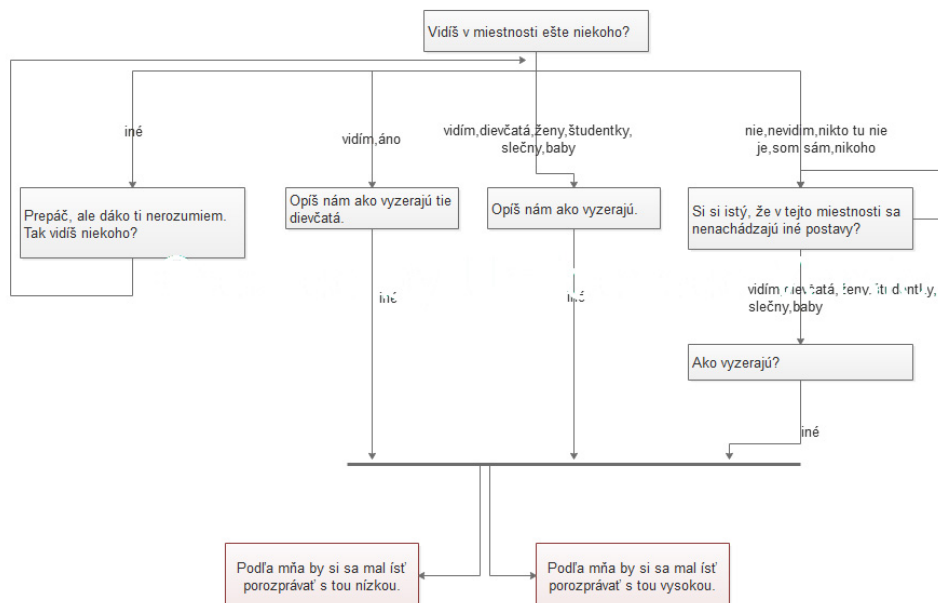
```
<dialog id="10" type="joke">
  <interactionPossibility ref="int5pos1"/>
  <lines>
    <line>
      <gameEntity ref="joker"/>
      <text>Prečo Chuck Norris nepije s východniarmi?</text>
    </line>
    <line>
      <gameEntity ref="joker"/>
      <text>Nebude si predsa kažiť imidž neporaziteľného
chlapíka.</text>
    </line>
  </lines>
  <interaction id="int7">
    <source ref="joker"/>
    <destination ref="jozef"/>
    <interactionType ref="finishConversation"/>
  </interaction>
</dialog>
```

## 2.7.9 ChatBot – Dialógový manažér

V našom návrhu uvádzame popis modelu konverzácií uložených v samostatných Xml súboroch. V tejto štruktúre je uložené správanie konverzácie medzi postavou v hre a hráčom. Môžeme si to predstaviť ako nejaký rozhodovací strom konverzácie kde vrcholy sú texty, ktoré hovorí postava a prechody medzi nimi sú podmienené zoznamom kľúčových slov. Na obrázku uvádzame grafický príklad konverzácie v druhej scéne medzi avatarmi a hráčom. Môžeme vidieť, že avatar položí otázku a na základe rozpoznávaného textu vyberieme najvhodnejšiu cestu. V prípade, že nesedí ani jedna možná cesta, ideme po vetve „iné“, ktorá položí opäť tú istú otázku.

Na konci sa cesty zbiehajú a je viac menej jedno čo hráč odpovie, avatari mu na konci dajú radu. Rada je vyznačená červeným a nie je v poradí, pretože tieto dve vety sa vykonávajú v náhodnom poradí, preto sú v strome znázornené paralelne.

## Scéna 2



Obrázok 32: Rozhodovací strom konverzácie, vrcholy - texty postavy, prechody - zoznam kľúčových slov

V aplikácií pre celé fungovanie konverzácií je vytvorený malý komponent s názvom ChatBot, ktorý poskytuje preklad Xml súboru do potrebných dátových štruktúr a následnú reprezentáciu konverzácie v hre.

Bežný postup je nasledovný:

- V interakčnom manažeri sa po hráčovom kliknutí na postavu v hre zavolá akcia doConversation - zavolaním funkcie DoConversation(id) v ChatBot.cs s daným id
- Následne sa v príslušnom Xml súbore vyhledá konverzácia s príslušným id a načíta do dátových štruktúr vhodných na použitie pomocou funkcie InitConversation(id)
- Na načítanie z Xml súboru nám slúži trieda ConversationXmlParser, ktorá sa stará o správne prečítanie a „pochopenie“ údajov uložených v externom súbore
- Po úspešnej inicializácii sa nastaví stav chatBota na aktívny a v tomto stave stále sleduje priebeh dialógu
- V každej novej snímke (frame) chatBot zisťuje či nie je koniec, či už neprebíha nejaká akcia a ak nie či má hovoriť postava (zavolá funkciu TellSentence(), ktorá povie príslušnú vetu) alebo hráč (zavolá funkciu ListenResponse(), ktorá čaká na odpoveď)
- Ak hovorí postava (TellSentence) z inštancie konverzácie získame aktuálnu vetu (sentenceEntity), ktorá obsahuje
  - text, ktorý má povedať postava
  - Prípadne informáciu že je posledná
  - Či má dať následne radu
  - Či sa čaká na odpoveď hráča, alebo sa má povedať ďalšia veta opäť postavou
  - Hlas postavy (ak nie je uvedené – striedajú sa hlasy avatarov)
- Povie sa text a na základe nepovinných údajov sa vykonajú príslušné akcie
- Ak hovorí hráč (ListenResponse) z aktuálnej sentenceEntity zistíme, či netreba použiť textový vstup a podľa toho prispôbime spôsob zobrazenia výzvy na vstup (Obrázok 29)



Obrázok 33: Rôzne zobrazenia výzvy na vstup od hráča

- Následne hra čaká na vstup, či už napísanie odpovede a potvrdenie enterom, alebo stlačenie R a hovorenie textu až kým sa klávesa R nepustí
- Hneď po skončení vstupu sa spracuje
- Spracovanie hovoreného slova prebieha v dvoch krokoch
  - Najprv sa zvukový formát pošle do ASR poskytovaného SAV
  - Následne sa rozpoznaný text pošle do lematizéra<sup>1</sup>
- Teraz máme k dispozícii zoznam slov, ktoré hráč povedal v základom tvare
- Pre danú sentenceEntity prechádzame možné odpovede – zoznamy kľúčových slov a hľadáme taký zoznam, ktorý sa počtom najviac zhoduje (ak nenájdeme použijeme prázdny)
- K nájdenému zoznamu slov vieme určiť akou vetou má postava v hre odpovedať
- A takto sa pokračuje až kým sa nenájde veta postavy alebo zoznam kľúčových slov hráča, ktorý ukončuje dialóg.

Najdôležitejšou časťou všetkých rozhovorov je správna príprava ASR z našej strany. Pre správne fungovanie potrebuje SAV čo najlepší zoznam viet a slov, ktoré očakávame od hráča. Preto je potrebné pre každú konverzáciu čo najlepšie premyslieť možné odpovede a vety postáv formulovať tak, aby sa čo najviac zúžila množina odpovedí.

Následne potrebujeme komunikovať tieto texty do SAV aj s našim externým zadávateľom. Často sa snažíme využiť aj ich pomoc pre vhodné formulovanie a skladanie všetkých stromov konverzácií ale hlavne možných kľúčových slov.

### 2.7.10 MiniHra – Flappy Harry

Táto funkcionality predstavuje pre hráča spôsob akým môže získavať v rámci hry mince, fiktívnu menu, za ktorú sa nakupuje v obchode. Nám umožní viesť do hry ďalšie zo zábavných prvkov slúžiacich na vtiahnutie hráča do hry. Hlavnou úlohou hráča je vyhýbať sa pri lete prekážkam a nepriateľom, ktorý ho naháňajú a zároveň zbierať mince i životy.

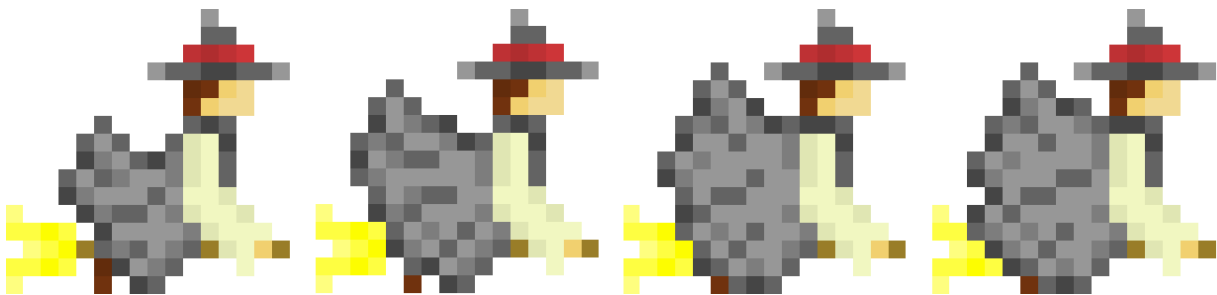
Keďže minihra sa spúšťa cez prístroj pripomínajúci staré arkádové hry, rozhodli sme sa, že jej grafika bude tieto staršie hry pripomínať tiež. Bolo potrebné vytvoriť grafické reprezentácie letiacej postavy, ktorú ovláda hráč, prekážok, nepriateľov, mincí a životov, ktoré hráč zbiera a tiež prostredie samotné. Všetky pohybujúce sa prvky bolo tiež nutné vytvoriť tak aby bolo následne možné ich animovať, čiže ako sled troch a viacerých obrázkov

<sup>1</sup> Lematizér: Lematizácia je určenie základného (slovníkového) tvaru slova, vyvinutú na FIIT STU <http://text.fiit.stuba.sk/#sekcia-lematizacia>

v rôznych štádiách pohybu. Hráčom ovládaná postavička prešla viacerými iteráciami, ktoré môžeme vidieť na obrázkoch číslo 33 a 34.



Obrázok 34: Grafická reprezentácia hráčom ovládanej postavy v minihre, iterácia 1.0, pripravená na animáciu



Obrázok 35: Grafická reprezentácia hráčom ovládanej postavy v minihre, iterácia 2.0, pripravená na animáciu

Model minihier spúšťaných cez automat je rozšíriteľný o ďalšie minihry a nové diskety v hlavnej hre. Implementácia minihry FlappyHarry je rozdelená na skripty ovládajúce herné entity a potom na skripty reagujúce na podnety z herných entít.

Jadrom je trieda `Flappy` ktorá implementuje skript pre správanie objektu hráča v prostredí. Tu je implementované ovládanie hráčovej postavy klávesami aj reakcie postavy na interakcie s okolitými objektami. Taktiež je tu vyhodnocované, či je hra zapauzovaná alebo ukončená.

Po kolízií s príšerkou alebo prekážkou hráč stratí život, je „nehmotný“ a obrázok postavy zbledne, zároveň sa hra spomalí a toto celé trvá niekoľko sekúnd, kým neskončí asynchronný task `AfterHurtProtection` ktorý čaká počet sekúnd nastavený parametrom `ProtectionTime` priamo v Unity.

```
IEnumerator AfterHurtProtection()
{
    // Mutex lock
    _protectionActive = true;
    // Animation change
    GetComponent<Animator>().Play("jozef-ghost_");

    // Slow down
    Time.timeScale = SlowDown;
    // Wait
    yield return new WaitForSeconds(ProtectionTime);
    // Speed up to normal
    Time.timeScale = 1.0f;

    // Animation change revert
    GetComponent<Animator>().Play("jozef_");
    // Mutex unlock
    _protectionActive = false;
}
```

Pomocný skript s triedou `CameraFollow` zabezpečuje držanie hráčovej postavy v centre obrazovky počas celej hry. Skript s triedou `Obstacle` je nasadený na každom objekte pohyblivej prekážky (kruhu) a na základe parametrov implementuje periodický pohyb prekážky po vertikálnej osi v 2D priestore.

Pohyb a správanie príšeriek je implementované v triede `Attack` funkciou `LateUpdate` kde sa pri každom novom vykreslení príšerka pootočí smerom k novej pozícii hráča a posunie sa k nemu o vzdialenosť danú rýchlosťou akú má nastavenú v parametroch skriptu v Unity.

```
void LateUpdate()
{
    float    actualDist    =    Vector2.Distance(Ball.transform.position,
transform.position);
    if (actualDist <= Dist)
    {
        Vector2 Heading = (Ball.transform.position - transform.position);
        Vector2 Direction = Heading / Heading.magnitude;

        // Move in the target's direction
        GetComponent<Rigidbody2D>().velocity = Direction*Speed;
        // Turn to the target's direction
        transform.up = GetComponent<Rigidbody2D>().velocity;
    }
}
```

Ďalej triedy `ScoreManager` a `LifeManager` udržiavajú pomocou funkcie `Update()` aktuálny stav skóre hráča na obrazovke v zmysle počtu nazbieraných mincí a zostávajúcich životov (srdiečok). Tieto objekty udržiavajú atribúty so súčasnými hodnotami, no sú upravované externe zo skriptu objektu hráča, čiže `Flappy`.

Napokon v objekte `Pause` je implementovaná funkcionálna pauzovania a odpauzovania hry spolu so zobrazením úvodných inštrukcií k hre a záverečného výsledku hry po skončení hry (buď víťazstvo alebo prehra). Zapauzovanie a odpauzovanie hry je riadené klávesom P, prípadne odpauzovať sa dá aj medzerníkom a rovno tak spôsobiť odrazenie sa. Pauza hry je implementovaná funkciou `private void PauseGame()` znížením parametra `Time.timeScale` na nulu, čo spôsobí zastavenie plynutia času hry, ďalej je zobrazený UI panel s inštrukciami pomocou príkazu `PauseBackground.enabled = true` a hudba hry je taktiež zapauzovaná priamo Unity funkciou nad objektom typu `AudioSource` príkazom `BackgroundMusic.Pause()`.

### 2.7.11 Logovanie

Hra obsahuje viaceré rozhodnutia. Výber týchto rozhodnutí tvorí výstupné dáta na nadväzujúci experiment. Preto je potrebné tieto rozhodnutia ukladať.

Ide o statickú triedu ktorá má tieto hlavné public metódy.

```
public static void Log(int decisionId, string info, int avatar, bool modified)
```

Táto metóda pridáva do výsledného súboru riadok textu s konkrétnym rozhodnutím. Na to je použité číselné identifikačné číslo a popisujúci reťazec. Výsledok rozhodnutia je reprezentovaný poradovým číslom avatara a pre účely výskumu slúži atribút `modified`, ktorý zachycuje či bola rada od avatara rečovo modifikovaná alebo nie.

Logovací skript obsahuje ešte ďalšie, menej podstatné, preťaženia tejto metódy na logovanie iných akcií a udalostí v hre. Na rozoznanie o aký typ logu ide slúžia ešte reťazcové identifikátory `TYPE_ACTION`, `TYPE_DECISION` a `TYPE_OTHER`.

```
public static void Log(string text)
```

Predchádzajúca metóda sa ukázala byť príliš špecifická na vyhovenie ďalším požiadavkám. Preto sme pridali metódu s jediným reťazcovým argumentom a výsledné logy je následne možné parsovať na získanie potrebných informácií.

```
public static void SaveLoggedData()
```

Táto metóda sa pôvodne volala vždy po opustení scény a uložila všetky novo vytvorené rozhodnutia a iné zaznamenané akcie do súboru. Po konzultácii so zákazníkom sa táto metóda volá priebežne po každej vykonanej akcii.

Okrem logovania textu a teda všetkých vykonaných akcií v hre sa logujú aj všetky zvukové záznamy. Na odlišenie jednotlivých herných posedení funguje tento mechanizmus v release verzii tak, že pri spustení hry sa vytvorí unikátny timestamp\_session priečinok, kde sa následne ukladajú tieto záznamy spolu s textovými logmi.

Záznamy majú v názve TTS alebo ASR na rozlíšenie medzi týmito dvoma typmi.

## 2.7.12 Nastavenia

Pred začatím hry sa zobrazí scéna s nastaveniami, v ktorej sa definujú globálne parametre hry. Hlasitosť umožňuje vyvážiť hlasitosti hudby v pozadí, zvukov priestoru v hre a samotných hlasov Avatarov.

Pohlavie hráča je potrebné správne nastaviť pred začiatkom experimentu, aby bol ASR modul schopný rozpoznávať text z reči s čo najvyššou presnosťou, keďže ASR modely boli tréňované separátne pre rôzne pohlavia.

Zobrazenie ASR konzoly je taktiež parametrom hry, pričom by však tento parameter mal byť zapnutý iba pri ladení vývojármi produktu. Zobrazenie TTS textov je parameter, ktorý definuje, či syntetizovaná reč avatarov a hovoriacich postáv v hre bude zároveň aj vypisovaná na obrazovku, alebo bude iba prehraný syntetizovaný zvuk reči daného textu, pričom miesto textu samotného bude zobrazený neutrálny text hry, informujúci hráča, že daná entita práve hovorí, formou *\*[Meno entity] hovorí\**.

Taktiež sa v nastaveniach dá vybrať, ktoré metriky budú použité na meranie rýchlosti reči.

Okrem týchto nastavení priamo ovplyvniteľných hráčom máme pre potreby testovania ešte hru rozlíšenú na dve verzie – debug a release. Podľa toho v akom režime je hra nastavená sú prístupné funkcie na preskočenie dialógov či zmenu počtu peňazí. Logovanie buď vytvára nové priečinky alebo prepisuje jeden súbor a takisto obsah inventára je mierne odlišný.

Počas vývoja sme si vytvorili dva módy hry a to debug a release. V debug móde sú dostupné niektoré funkcie, ktoré sa v release verzii nemôžu nachádzať:

- preskočenie dialógov
- zmenu počtu peňazí
- vypnutie logovania
- obsah inventára je mierne odlišný

Prepínať sa medzi týmito módmi je možné len priamo v kóde, aby tieto funkcie neboli dostupné bežným používateľom.

Klasické nastavenia, ktoré používa hráč sa zobrazia na začiatku pred začatím hry. Bližší popis nastavení je v kapitole 3.2. Obrazovka s nastaveniami je na obrázku 35.

Všetky nastavenia z tejto obrazovky sa ukladajú do statickej public triedy a sú následne dostupné počas celej hry, aby sa hra podľa nich mohla riadiť.



## 2.8 Testovanie

Testovanie v nástroji Unity je realizované pomocou rozšírenia **Unity Test Tools**. Toto rozšírenie umožňuje na samostatnej scéne zadefinovať integračné testy alebo využívať komponenty typu Assert, ktoré je možné priradiť akémukoľvek objektu v scéne. Tieto komponenty potom môžu porovnávať ľubovoľné dátové typy a vyhodnotiť úspešnosť testu na základe splnenia alebo nesplnenia predpokladu.

Toto rozšírenie umožňuje zároveň vytvárať Unit testy pre testovanie vlastných vytvorených tried. Princíp unit testov, čo predstavuje izolovanosť testovaného celku od zvyšku hernej logiky je tu implementovaný pomocou takzvaných Dummy, Moq a Spy objektov, ktoré sa používajú namiesto reálnych objektov. Všetky testy je možné spúšťať z grafického rozhrania, ktoré je integrované priamo v Unity.

Ďalej sme robili manuálne testy prechádzaním hry a overovali sme či funkcionality ostala rovnaká a zároveň sme preverili nové / pridané prvky. Taktiež sme pravidelne posielali export hry zadávateľovi, ktorí nám hlásil prípadné chyby alebo nedostatky.

## 3 Príručky

---

### 3.1 Inštalačná príručka

Exportovanú hru stačí nakopírovať do počítača a spustiť. Hra obsahuje zložku Assets, zložku Data a súbor traspi.exe, ktorý zapne hru. Pred spustením je potrebné stiahnuť a nainštalovať aj DirectX verziu 10 a novšiu.

### 3.2 Používateľská príručka

Pri hraní hry je dôležité aby dané zariadenie bolo pripojené na internet. Pre lepšiu plynulosť hry by rýchlosť internetu mala byť čo najvyššia a traffic čo najnižší.

### 3.2.1 Nastavenia



Obrázok 36 Nastavenia hry v úvode

Pri zapnutí hry sa ukáže okno s nastaveniami. Vieme si zakliknúť či chceme hudbu alebo nie a aká hlasná má hudba byť. Ďalej či chceme zvuky (sem patria špeciálne efekty, zvuky chôdze...) a ich hlasitosť. Posledné políčko v kolónke hlasitosť je hlas reči avatarov.

Pre potreby rozpoznávača nastavíme či je hráč muž alebo žena.

Ďalej si môžeme zvoliť či chceme zobrazovať Automatic Speech Recognition do konzoly alebo nie a tiež či chceme zobrazovať Text To Speech, čiže to čo postavy hovoria aj textovo na obrazovke. Takisto je tu možnosť výberu z viacerých metrík na meranie rýchlosti reči, najmä na testovacie účely.

### 3.2.2 Ovládanie hry

Pohyb po obrazovke je riadený pomocou myšky. Hráč ukáže na miesto kde chce ísť alebo na objekt ktorý chce použiť a ľavým klikom to vykoná. Vyvolanie inventára je pomocou klávesy „I“. Otvorenie obchodu sa vykoná stlačením „O“.

## 3.3 Príručka k súboru s logmi

Vzhľadom na rozsiahlosť logovania ja aj súbor s týmito logmi dosť komplikovaný. Pre tento účel sme spísali túto príručku. Pri logovaní sú podstatné tri hlavné oblasti logovania.

Prvá je zvuková časť, sem spadajú všetky nahrávky a požiadavky na TTS či ASR. Toto logovanie vyzerá nasledovne:

5/19/2016 9:21:25 AM; Output: GameEntity="Avatar" Text="Aha Jozef! Tie klúče budú celkom určite od tejto starobylej truhlice! Zviditeľnili sa určite preto, lebo si sa jej dotkol. Dobre sa rozhodni, ktorý kľúč si vezmeš, lebo použiť môžeš iba jeden z nich."

5/19/2016 9:21:25 AM; TTS request(voice: hmm\_unisyn\_bajnokova, rate: 5.2625535, pitch: 134.5, volume: 61.06903197104398, rate\_n: 3.6788165, pitch\_n: 123.58, volume\_n: 59.71620592156748, mode: adapt)

5/19/2016 9:21:26 AM; Saved file

C:\Users\Lukas\SkyDrive\Patronus\builds\prototyp4TTS2\Assets\Logs\GameSession\_20160519091750361\20160519092126419TTSrecording.wav

V prvom riadku je napísaný text, ktorý chceme pretransformovať na reč. V druhom riadku sú parametre poslanej požiadavky a v treťom riadku je odkaz na tento súbor uložený na disku.

5/19/2016 9:21:53 AM; Saved file

C:\Users\Lukas\SkyDrive\Patronus\builds\prototyp4TTS2\Assets\Logs\GameSession\_20160519091750361\20160519092153283ASRrecording.wav

5/19/2016 9:21:56 AM; ASR Translation: ublíži Ježiš na s don dal je mohla aj s konci koľko jaj takého slova

5/19/2016 9:21:56 AM; ASR Response Data(speed: 4.925203, pitch: 133.41, averageSyllPerS: 5.372406, averagePhonesPerS: 14.163615, averageWordsPerS: 3.663004, speechDuration: 4.095000, syllCount: 22, wordCount: 15, phoneCount: 58, mean: 133.41, sigma: 0)

ASR logovanie funguje obdobne, len poradie týchto riadkov je odlišné. Najskôr tu máme odkaz na disk kde sa nahrávka uložila, následne text, ktorý modul SAV rozpoznal a v poslednom riadku sú parametre, ktoré sa pri tomto rozprávaní odmerali.

Druhá časť je logovanie rozhodnutí, v súbore má takúto štruktúru:

5/19/2016 9:22:31 AM; Advice: id=2, avatar=1, text="Ten oranžový kľúč má zreteľne menej zubov. Radím ti, neriskuj že otvoriš pandorinu skrinku a vezmi ten."

5/19/2016 9:22:41 AM; Advice: id=2, avatar=2, text="Ja som zvedavý na to, čo by si našiel vo vnútri s použitím žltého kľúča. Hovorí sa preda, že risk je zisk."

5/19/2016 9:22:51 AM; Interaction="clickTo": Jozef (idle) KeyRight (idle)

5/19/2016 9:22:57 AM; Interaction advice\_ref=2

V tomto prípade sa v súbore nachádzajú vždy dve rady pričom obidve majú určený identifikátor. Vieme vyčítať ku ktorému avatarovi bol priradený aký text rady. Následne sú všetky akcie asociované s týmto rozhodnutím označené riadkom obsahujúcim atribút advice\_ref z čoho vieme späť priradiť vykonané akcie k radám.

Tretia časť je doplňujúca, predstavuje všetky ostatné vykonané akcie ako napríklad kúpu predmetu, pustenie zvuku, dokončenie konverzácie a podobne. Jej štruktúra je samovysvetľujúca

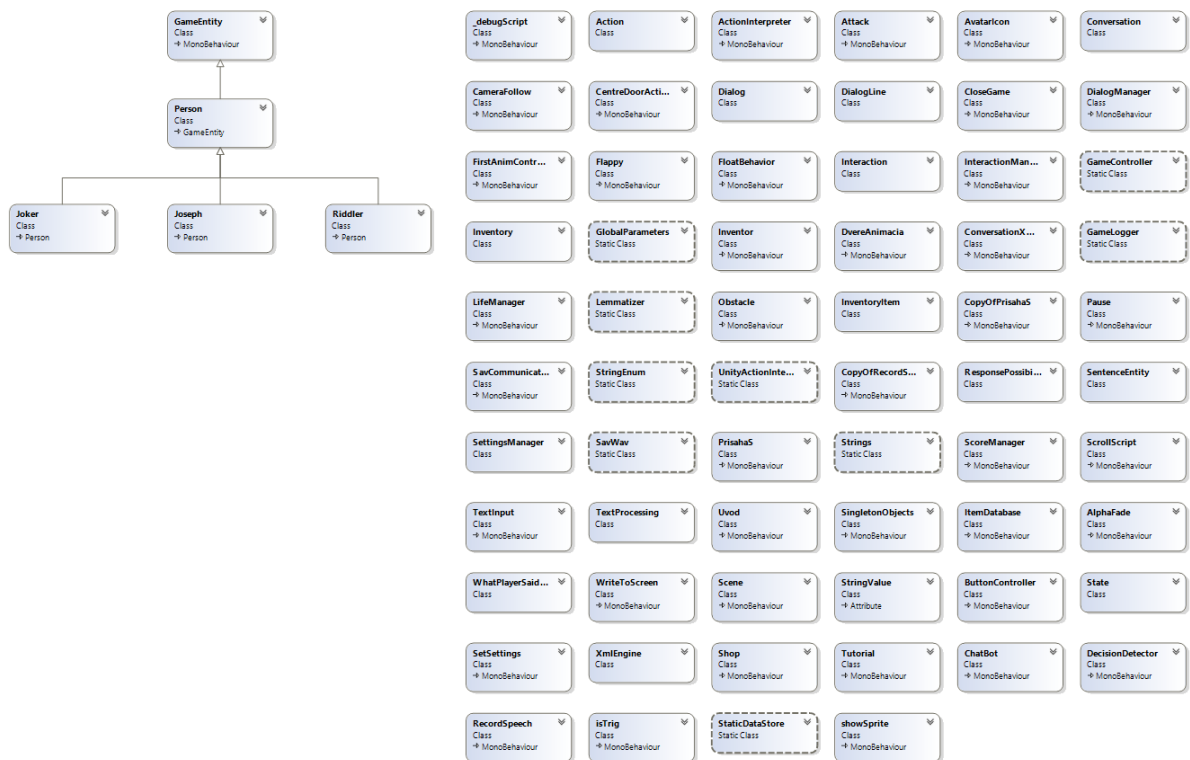
5/19/2016 9:22:57 AM; Action: playSound(chimpanzeeTaken) performed.

5/19/2016 9:22:57 AM; Action: removeItem(13) performed.

5/19/2016 9:22:57 AM; Action: addItem(15) performed.

5/19/2016 9:22:59 AM; Grabbed item: Chimpanzee(15)

## 4 Technická dokumentácia



Obrázok 37: Zoznam tried projektu

V prílohe uvádzame aj konkrétne schémy k Xml súborom použitých v našom projekte

# TÍMOVÝ PROJEKT

## TraSpi

Dokumentácia k riadeniu

Bc. Hajdu Daniela  
Bc. Marták Lukáš  
Bc. Mäsiar Aleš  
Bc. Miškovský Lukáš  
Bc. Moravčíková Zora  
Bc. Šandor Filip

**Tím č. 20:** TraSpi  
**Vedúci projektu:** Ing. Eduard Kuric  
**Predmet:** Tímový projekt 1  
**Ročník:** 2015/2016  
**Mailový kontakt:** [traspi-team@googlegroups.com](mailto:traspi-team@googlegroups.com)

# Obsah

---

|        |  |    |
|--------|--|----|
| 1      | Riadenie projektu .....                        | 1  |
| 1.1    | Role členov tímu a podiel práce .....          | 1  |
| 1.2    | Aplikácie podporujúce manažment projektu ..... | 2  |
| 1.2.1  | Zdieľanie dokumentov .....                     | 2  |
| 1.2.2  | Komunikácia .....                              | 3  |
| 1.2.3  | Evidencia úloh .....                           | 3  |
| 1.2.4  | Verziovanie kódu a projektu .....              | 3  |
| 1.2.5  | Nastavenie projektu .....                      | 3  |
| 1.3    | Sumarizácia šprintov .....                     | 3  |
| 1.3.1  | Oculus Reparo .....                            | 3  |
| 1.3.2  | Alohomora .....                                | 4  |
| 1.3.3  | Expelliarmus .....                             | 5  |
| 1.3.4  | Lumos .....                                    | 6  |
| 1.3.5  | Rictumsempra .....                             | 7  |
| 1.3.6  | Bombarda .....                                 | 7  |
| 1.3.7  | Cruciatus .....                                | 8  |
| 1.3.8  | Avada Kedavra .....                            | 9  |
| 1.3.9  | Imperius .....                                 | 10 |
| 1.3.10 | Finite Incantatum .....                        | 12 |
| 1.3.11 | Immobulus .....                                | 13 |
| 1.3.12 | Expecto patronum .....                         | 14 |
| 1.4    | Používané metodiky .....                       | 14 |
| 1.5    | Retrospektíva .....                            | 16 |
| 1.5.1  | Oculus reparo .....                            | 16 |
| 1.5.2  | Expelliarmus .....                             | 16 |
| 1.5.3  | Lumos .....                                    | 16 |
| 1.5.4  | Rictumsempra .....                             | 16 |
| 1.5.5  | Bombarda .....                                 | 17 |
| 1.5.6  | Cruciatus .....                                | 17 |
| 1.5.7  | Avada Kedavra .....                            | 17 |
| 1.5.8  | Imperius .....                                 | 17 |
| 1.5.9  | Finite Incantatum .....                        | 17 |
| 1.5.10 | Immobulus .....                                | 17 |

|        |  |    |
|--------|--|----|
| 1.6    | Zoznam kompetencií členov tímu .....   | 18 |
| 1.7    | Motivácia jednotlivých trojíc pre výber témy .....   | 19 |
| 1.7.1  | Trojica č. 20.....   | 19 |
| 1.7.2  | Trojica č. 25.....   | 20 |
| 1.7.3  | Usporiadané zoznamy tém podľa priorít.....   | 22 |
| 2      | Metodiky .....   | 24 |
| 2.1    | Metodika na tvorbu metodiky.....   | 24 |
| 2.1.1  | Role a zodpovednosti .....   | 24 |
| 2.1.2  | Proces splnenia úlohy .....  | 24 |
| 2.2    | Metodika ukladania súborov v zdieľanom priečinku OneDrive .....  | 26 |
| 2.2.1  | Pojmy .....  | 26 |
| 2.2.2  | Role a zodpovednosti .....   | 26 |
| 2.2.3  | Onedrive .....   | 26 |
| 2.2.4  | Stromová štruktúra .....   | 26 |
| 2.3    | Názvy šprintov .....   | 27 |
| 2.3.1  | Šprinty .....  | 27 |
| 2.4    | Metodika na tvorbu grafických prvkov do hry .....  | 28 |
| 2.4.1  | Pojmy .....  | 28 |
| 2.4.2  | Role a zodpovednosti .....   | 28 |
| 2.4.3  | Procesy splnenia úlohy .....   | 28 |
| 2.5    | Metodika na prácu s úlohami.....   | 30 |
| 2.5.1  | Pojmy .....  | 30 |
| 2.5.2  | Role a zodpovednosti .....   | 30 |
| 2.5.3  | Proces splnenia úlohy .....  | 30 |
| 2.6    | Metodika na tvorbu zápisníc.....   | 32 |
| 2.6.1  | Role a zodpovednosti .....   | 32 |
| 2.6.2  | Proces splnenia úlohy .....  | 32 |
| 2.7    | Pravidlá tvorby skíc na animovanie.....  | 34 |
| 2.8    | Metodika na tvorbu zdrojových kódov pre skripty a programové komponenty vytvárané v programovacom jazyku C#..... | 35 |
| 2.8.1  | Pojmy .....  | 35 |
| 2.8.2  | Role a zodpovednosti .....   | 35 |
| 2.8.3  | Procesy .....  | 35 |
| 2.10   | Metodika verziovania kódu .....  | 37 |
| 2.10.1 | Pojmy .....  | 37 |

|        |  |    |
|--------|--|----|
| 2.10.2 | Role a zodpovednosti .....                                 | 37 |
| 2.10.3 | Procesy .....  | 37 |
| 2.11   | Súborová štruktúra Unity projektu .....                    | 41 |
| 2.11.1 | Pojmy .....  | 41 |
| 2.11.2 | Role a zodpovednosti .....                                 | 41 |
| 2.11.3 | Priečinky.....   | 41 |
| 2.12   | Metodika na tvorbu testov - UnityTestTools.....            | 42 |
| 2.12.1 | Nadväzujúce metodiky a dokumenty .....                     | 42 |
| 2.12.2 | Role a zodpovednosti .....                                 | 42 |
| 2.12.3 | Procesy .....  | 42 |
| 2.13   | Metodika rizík.....  | 44 |
| 2.13.1 | Identifikované riziká .....                                | 44 |
| 2.13.2 | Dodržiavané zásady na všeobecné predchádzanie rizikám..... | 44 |
| 2.14   | Metodika komunikácie .....                                 | 45 |
| 2.14.1 | Pojmy .....  | 45 |
| 2.14.2 | Role a zodpovednosti .....                                 | 45 |
| 2.14.3 | Procesy .....  | 45 |
| 3      | Export z TFS .....   | 48 |
| 4      | Prílohy .....  | 56 |
| A.     | Integrácia TFS a Unity .....                               | 56 |
| A-1    | Install .....  | 56 |
| A-2    | Configure .....  | 56 |
| A-3    | Pripojenie sa na aktuálny projekt.....                     | 57 |
| B.     | Scénar pre prvú scénu .....                                | 58 |
| C.     | Analýza hry The Stanley Parable .....                      | 60 |
| D.     | Prvotné nápady na príbeh.....                              | 62 |
| E.     | Popis dialógov v hre .....                                 | 64 |
| E-1    | Scéna 1.....   | 64 |
| E-2    | Scéna 1.....   | 65 |
| E-3    | Scéna 2.....   | 65 |
| E-4    | Scéna 2.....   | 66 |
| E-5    | Scéna 3.....   | 66 |
| E-6    | Scéna 4.....   | 68 |
| E-7    | Scéna 4.....   | 68 |
| E-8    | Scéna 5.....   | 69 |



|      |   |     |
|------|---|-----|
| E-9  | Scéna 6.....                                      | 69  |
| E-10 | Scéna 6.....                                      | 70  |
| F.   | Rozhodnutia .....                                 | 71  |
| G.   | Plagát TP Cup.....                                | 74  |
| H.   | Schémy Xml súborov .....                          | 75  |
| H-1  | Model interakcií.....                             | 75  |
| H-2  | Model dialógov .....                              | 83  |
| H-3  | Model konverzácií .....                           | 86  |
| I.   | Zápisnice zo stretnutí tímového projektu .....    | 90  |
| I-1  | Zápisnica z 1. stretnutia tímového projektu.....  | 90  |
| I-2  | Zápisnica z 2. stretnutia tímového projektu.....  | 92  |
| I-3  | Zápisnica z 3. stretnutia tímového projektu.....  | 95  |
| I-4  | Zápisnica z 4. stretnutia tímového projektu.....  | 100 |
| I-5  | Zápisnica z 5. stretnutia tímového projektu.....  | 103 |
| I-6  | Zápisnica z 6. stretnutia tímového projektu.....  | 106 |
| I-7  | Zápisnica z 7. stretnutia tímového projektu.....  | 109 |
| I-8  | Zápisnica z 8. stretnutia tímového projektu.....  | 112 |
| I-9  | Zápisnica z 9. stretnutia tímového projektu.....  | 114 |
| I-10 | Zápisnica z 10. stretnutia tímového projektu..... | 116 |
| I-11 | Zápisnica z 11. stretnutia tímového projektu..... | 118 |
| I-12 | Zápisnica z 12. stretnutia tímového projektu..... | 120 |

# 1 Riadenie projektu

---

Dokumentácia riadenia projektu obsahuje podrobný opis spôsobu riadenia tímového projektu. Uvádza všetkých členov, ich pridelené úlohy a roly v tíme, venuje sa manažmentom projektu, ktoré sú aplikované pre správne riadenie. Popisuje všetky existujúce šprinty, ich sumarizáciu a používané metodiky.

## 1.1 Role členov tímu a podiel práce

Náš tím sa skladá zo šiestich členov a na čele vedúceho je Ing. Eduard Kuric. Každý člen zastáva určitú úlohu v tíme, na základe ktorej vykonáva a plní svoje povinnosti. Mnohé, napríklad dokumentovanie tímových stretnutí v podobe zápisníc alebo rôzne kreatívne úlohy sme zastávali všetci spoločne alebo sa zodpovednosť za ne striedala.

Ďalej uvádzame zoznam členov tímu aj s ich príslušnou rolou v tíme a niekoľkými dlhodobými alebo krátkodobými manažérskymi úlohami, za ktoré sú zodpovední.

### **Bc. Filip Šandor**

- Vedúci tímu
- Podnecovanie a motivovanie členov tímu.
- Školenie tímu v práci s potrebnými nástrojmi.
- Tvorba grafických reprezentácií skíc v počítači (scény a postava).

### **Bc. Zora Moravčíková**

- Manažér plánovania
- Sledovanie termínov a dohľad nad plnením úloh.
- Špecialista na vývojový game engine Unity.
- Riešenie source control a s tým spojených problémov.

### **Bc. Lukáš Miškovský**

- Manažér rizík
- Identifikácia rizík.
- Špecialista na prácu s TFS nástrojom.
- Riešenie animácie pohybov charakterov.

### **Bc. Aleš Mäsiar**

- Manažér dokumentovania
- Dohliada na plnenie úloh spojených s dokumentáciou.
- Tvorba a správa webovej stránky tímového projektu.

### **Bc. Lukáš Marták**

- Manažér kvality
- Návrh softvérovej architektúry.
- Kontrola kvality výsledkov úloh
- Špecifikuje sa na oblasť umelej inteligencie a jej využitie v našom projekte

### **Bc. Daniela Hajdu**

- Manažér komunikácie

- Komunikácia pri dohadovaní termínov stretnutí tímového projektu.
- Grafické návrhy scén, postav a predmetov do hry.
- Dohľad na konzistenciu grafickej stránky projektu.

Tabuľka 2. Rozdelenie prínosu členov do jednotlivých častí dokumentu

| Dokument          | Kapitola     | Filip Šandor | Zora Moravčíková | Lukáš Miškovský | Aleš Mäsiar | Lukáš Marták | Daniela Hajdu |
|-------------------|--------------|--------------|------------------|-----------------|-------------|--------------|---------------|
| Riadenie          | Kapitola 1.1 | 10%          | 40%              | 10%             | 0%          | 0%           | 40%           |
|                   | Kapitola 1.2 | 5%           | 40%              | 5%              | 5%          | 5%           | 40%           |
|                   | Kapitola 1.3 | 20%          | 20%              | 20%             | 10%         | 10%          | 20%           |
|                   | Kapitola 1.4 | 16,6%        | 16,6%            | 16,6%           | 16,6%       | 16,6%        | 16,6%         |
|                   | Kapitola 1.5 | 12.5%        | 30%              | 12.5%           | 12.5%       | 12.5%        | 20%           |
|                   | Kapitola 1.6 | 16,6%        | 16,6%            | 16,6%           | 16,6%       | 16,6%        | 16,6%         |
|                   | Kapitola 1.7 | 16,6%        | 16,6%            | 16,6%           | 16,6%       | 16,6%        | 16,6%         |
|                   | Kapitola 1.5 | 12.5%        | 30%              | 12.5%           | 12.5%       | 12.5%        | 20%           |
|                   | Kapitola 1.6 | 16,6%        | 16,6%            | 16,6%           | 16,6%       | 16,6%        | 16,6%         |
|                   | Kapitola 1.7 | 16,6%        | 16,6%            | 16,6%           | 16,6%       | 16,6%        | 16,6%         |
|                   | Kapitola 2   | 16,6%        | 16,6%            | 16,6%           | 16,6%       | 16,6%        | 16,6%         |
|                   | Kapitola 3   | 0%           | 0%               | 50%             | 50%         | 0%           | 0%            |
|                   | Príloha A    | 0            | 50%              | 0               | 0           | 50%          | 0             |
|                   | Príloha B    | 16,6%        | 16,6%            | 16,6%           | 16,6%       | 16,6%        | 16,6%         |
|                   | Príloha C    | 100%         | 0                | 0               | 0           | 0            | 0             |
|                   | Príloha D    | 16,6%        | 16,6%            | 16,6%           | 16,6%       | 16,6%        | 16,6%         |
|                   | Príloha E    | 16,6%        | 16,6%            | 16,6%           | 16,6%       | 16,6%        | 16,6%         |
| Inžinierske dielo | Kapitola 1   | 16,6%        | 16,6%            | 16,6%           | 16,6%       | 16,6%        | 16,6%         |
|                   | Kapitola 2   | 10%          | 20%              | 20%             | 20%         | 20%          | 10%           |
|                   | Kapitola 3   | 16,6%        | 16,6%            | 16,6%           | 16,6%       | 16,6%        | 16,6%         |
|                   | Kapitola 4   | 15           | 18               | 16,5            | 18          | 15,5         | 17            |

## 1.2 Aplikácie podporujúce manažment projektu

### 1.2.1 Zdieľanie dokumentov

Ako prvý krok sme si zvolili OneDrive ako náš zdieľaný priečinok na dokumenty, najmä preto, že všetci používame OS Windows a poznáme výhody synchronizácie pomocou Microsoft OneDrive. V tomto priečinku sú uložené všetky dokumenty týkajúce sa projektu, okrem samotného kódu aplikácie.

## 1.2.2 Komunikácia

Na vzájomnú komunikáciu medzi členmi tímu sme zvolili Slack. Jednotlivé diskusie sú rozdelené tematicky a každý člen tímu sa vyjadruje k danej problematike do príslušného kanála. Keď je to potrebné členovia môžu medzi sebou komunikovať aj jednotlivo. Môžeme si nahrávať súbory, riešiť problémy, ktoré nepočkajú do stretnutia, plánovať udalosti pomocou kalendára.

Pre konferenčné hovory používame skupinu vytvorenú na Skype.

Taktiež máme založený spoločný mail, ktorý slúži najmä na komunikáciu s vedúcim, zákazníkom z Nitry, s verejnosťou alebo preposielanie informačných mailov medzi členmi tímu. Mailovú skupinu máme založenú na google groups.

## 1.2.3 Evidencia úloh

TFS zároveň podporuje source control projektu pomocou Git-u. A tak máme na jednom mieste samotný projekt, jeho kód verziovaný pomocou Git a zoznam úloh, ktoré treba spraviť. Dohodli sme sa vyvíjať našu hru v Unity Engine s podporou c# skriptov, ktoré budeme upravovať v Microsoft Visual Studiu (VS). Na manažment úloh preto využívame Team Foundation Server (TFS) hlavne kvôli jeho integrácie do VS.

## 1.2.4 Verziovanie kódu a projektu

Git je postačujúci nástroj na zdieľanie a verziovanie kódu, avšak je potrebné zdieľať celý projekt v Unity, ktorý obsahuje aj iné ako textové súbory. S takýmito konfliktmi si samotný Git nedokáže poradiť. Preto sme v nastaveniach Unity projektu nastavili, aby sa scény ukladali v textovom formáte namiesto binárneho. A platí dohoda, že každý upravuje assets, ktoré si vytvoril, aby nevznikali zbytočné konflikty. Avšak niektorým konfliktom sa nebude dať vyhnúť, preto používame aj Unity SmartMerge Tool, čo je nástroj na riešenie konfliktov v Unity súboroch.

## 1.2.5 Nastavenie projektu

Každý člen tímu ma k dispozícii návod ako správne nastaviť VS pre komunikáciu s TFS serverom. Tento návod obsahuje postup ako klonovať Git repozitár a vytvoriť z neho Unity projekt (Príloha A).

## 1.3 Sumarizácia šprintov

Následná časť dokumentu opisuje jednotlivé šprinty, ktoré sú momentálne dokončené alebo v prograse. Dĺžka jedného šprintu je dva týždne.

### 1.3.1 Oculus Reparo

Tabuľka 3: Zoznam úloh v prvom šprinte

| User Story                         | Úloha  | Člen tímu        |
|------------------------------------|--|------------------|
| -                                  | Dokument na gameplay brainstorming             | Filip Šandor     |
|                                    | Materiály a tutoriály k vývoju pomocou Unity3D | Lukáš Marták     |
|                                    | Spísať ohraničenia a otvorené otázky           | Daniela Hajdu    |
| <b>Navrhnutie Architektúry</b>     | Navrhnuť model architektúry                    | Lukáš Marták     |
| <b>Analýza herných mechanizmov</b> | Analýza Stanley Parable                        | Filip Šandor     |
|                                    | Analýza Unity Web Player                       | Zora Moravčíková |
|                                    | Skúmať motivačné faktory v hrách               | Aleš Mäsiar      |

|                                  |   |                  |
|----------------------------------|---|------------------|
|                                  |   | Lukáš Miškovský  |
| <b>Správa webu</b>               | Grafika webu                            | Zora Moravčíková |
|                                  | Vytvoriť základnú štruktúru webu        | Aleš Mäsiar      |
|                                  | Navrhnuť dizajn                         | Zora Moravčíková |
|                                  | Nasadiť web na server                   | Aleš Mäsiar      |
| <b>Úvodný manažment</b>          | Vymyslieť logo tímu                     | Zora Moravčíková |
|                                  | Zabezpečiť tímovú komunikáciu cez Slack | Aleš Mäsiar      |
|                                  | Zápisnice zo stretnutí                  | Daniela Hajdu    |
|                                  | Vymyslieť meno tímu                     | Filip Šandor     |
| <b>Tvorba príbehu</b>            | Vymyslieť záhadu                        | Aleš Mäsiar      |
|                                  | Napísať scenár                          | Filip Šandor     |
|                                  | Vymyslieť mená postáv                   | Filip Šandor     |
|                                  | Vymyslieť mini hry                      | Lukáš Miškovský  |
| <b>Návrh produkčného systému</b> | Analýza produkčných systémov a možností | Daniela Hajdu    |

V prvom šprinte sme sa dohodli aké procesy na manažovanie tímu budeme používať. Vytvorili sme si tímový mail, cloud na súbory, slack na tímovú komunikáciu. Ďalej sme vymysleli logo nášho tímu, názov tímu a vytvorili sme šablóny pre zápisnice. Aleš Mäsiar spolu s Zorou Moravčíkovou vytvorili web a nahrali ho na server. Prebehla analýza herných mechanizmov aby sme zistili atraktívne prvky na hrách a mohli ich zakomponovať do nášho projektu. Bližšie sme preskúmali hru The Stanley Parable. Taktiež sme zistili možnosti nahratia hotovej hry na web player.

V rámci tohto šprintu sme sa intenzívne venovali zostavovaniu konceptu a príbehu hry. Vytvorili sme záhadu ktorá bude jadrom nášho príbehu, do príbehu sme pridali mini hry ktoré hráča zabavia počas experimentu. Napísali sme scenár s opisom prostredia pre prvú kapitolu hry. Vytvorili sme animáciu pre vytvorenú hlavnú postavu. Scénu s animovanou postavou sme nahrali do Unity a vytvorili prvú scénu, po ktorej sa môže hráč pohybovať. V tomto šprinte sa nám podarilo úspešne splniť všetky stanovené úlohy. V tabuľke č. 2 sa nachádza zoznam všetkých úloh v rámci šprintu. Ku každej úlohe je uvedený aj člen tímu, ktorý ju mal vykonať alebo bol za jej úspešné vykonanie zodpovedný (v tomto šprinte sa vyskytlo mnoho úloh, ktoré sme riešili ako tím).

### 1.3.2 Alohomora

V tomto šprinte bolo našim cieľom navrhnuť grafickú reprezentáciu scény, preniesť ju do digitálnej podoby a vytvoriť Unity projekt, v ktorom budeme spoločne vytvárať hru. Bolo potrebné navrhnuť a aplikovať do scény aj hlavnú postavu. Funkčnosť výsledného produktu po tomto šprinte má byť možnosť pohybu hlavnej postavy po scéne.

Preto sme nakreslili prvú scénu najskôr na papier a potom aj do počítača, taktiež sme nakreslili hlavnú postavu. Po neúspešnom pokuse vytvoriť 3D postavičku sme sa rozhodli že hra bude celá v 2D.

Prebehlo školenie kde sa ostatní členovia tímu naučili pracovať s nástrojom RealDraw na kreslenie skíc.

Začali sme pracovať na abstraktnej reprezentácii scény a tiež source control na projekte. Úloha vytvorenia abstraktnej reprezentácie scény bola príťažká na tento šprint a tak sa preniesla do ďalšieho šprintu.

Okrem tejto úlohy sa nám ešte nepodarilo splniť úlohu vytvorenia modelu postavy podľa náčrtu a tá sa teda v zmenenej podobe tiež preniesie do nasledujúceho šprintu. Ostatné úlohy boli úspešne splnené.

V tabuľke č. 3 sa nachádza zoznam všetkých úloh v rámci šprintu.

*Tabuľka 4. Zoznam úloh v druhom šprinte*

| User Story                         | Úloha  | Člen tímu                        |
|------------------------------------|--|----------------------------------|
| -                                  | Testovanie v Unity   | Aleš Mäsiar                      |
| <b>Vedieť sa pohybovať v scéne</b> | Nakresliť prvú scénu na papier                             | Daniela Hajdu                    |
|                                    | Nakresliť postavy Jozefa na papier                         | Daniela Hajdu                    |
|                                    | Prekresliť scénu do digitálnej podoby                      | Filip Šandor                     |
|                                    | Vytvoriť model postavy podľa náčrtu                        | Filip Šandor                     |
|                                    | Vytvoriť projekt a zabezpečiť source control projektu      | Zora Moravčíková                 |
|                                    | Vytvoriť prvú scénu z vytvorených modelov a obrázkov scény | Aleš Mäsiar                      |
|                                    | Vytvoriť pohyb hráča po scéne                              | Aleš Mäsiar                      |
|                                    | Vytvoriť animáciu pohybu postavy                           | Lukáš Miškovský                  |
|                                    | Vymyslieť abstraktnú reprezentáciu scény                   | Lukáš Marták                     |
|                                    | Zaškoliť tím do RealDraw                                   | Filip Šandor                     |
|                                    | Vytvoriť model postavy podľa náčrtu                        | Daniela Hajdu<br>Filip Šandor    |
|                                    | Vyladiť funkčnosť source control                           | Zora Moravčíková<br>Lukáš Marták |
|                                    | Vymyslieť abstraktnú reprezentáciu scény                   | Zora Moravčíková<br>Lukáš Marták |

### 1.3.3 Expelliarmus

Jedným z našich hlavných cieľov v tomto šprinte je dokončiť úlohu prenesenú zo šprintu predchádzajúceho a to vymyslenie abstraktnej reprezentácie scény, ktorú sme v tomto šprinte vhodne premenovali. Tento krát sme touto náročnou úlohou nepoverili len jedného člena tímu, ale štyroch.

Ďalej, keďže v minulom šprinte sme sa rozhodli zmeniť náš prístup k vytváraniu grafickej reprezentácie postáv v hre bolo nutné venovať sa tejto úlohe aj naďalej. Je potrebné vytvoriť novú skicu, prepracovať dizajn postavy, tak aby bol vhodný pre 2D grafiku a nielen ako podklad pre vytvorenie 3D modelu. Ďalej túto skicu treba samozrejme spracovať, previesť do digitálnej podoby a následne prepracovať animáciu pohybu hlavnej postavy za použitia tejto novej grafickej reprezentácie.

Okrem úloh súvisiacich s vývojom produktu samotného, sa budú všetci členovia tímu v tomto šprinte venovať aj vypracovaniu dokumentácie, konkrétne treba dokumentovať riadenie projektu a inžinierske dielo.

*Tabuľka 5: Zoznam úloh v tretom šprinte*

| User Story   | Úloha   | Člen tímu                   |
|--|---|-----------------------------|
| <b>Špecifikácia interakcie s objektom a vylepšenie</b> | Vymyslieť reprezentáciu objektov a interakcie | Lukáš Marták<br>Aleš Mäsiar |

|         |  |   |
|---------|--|---|
| grafiky |  | Lukáš Miškovský<br>Zora Moravčíková   |
|         | Graficky zdokonaľiť hlavnú postavu                       | Filip Šandor  |
|         | Zdokonaľiť animáciu hlavnej postavy                      | Lukáš Miškovský   |
|         | Zdokonaľiť skicu hlavnej postavy                         | Daniela Hajdu   |
|         | Manažment projektu                                       | Daniela Hajdu<br>Lukáš Marták<br>Aleš Mäsiar<br>Lukáš Miškovský<br>Zora Moravčíková<br>Filip Šandor |
|         | Dokončiť grafickú reprezentáciu Jozefa                   | Daniela Hajdu   |
|         | Vytvoriť model scény a interakcií medzi hernými entitami | Lukáš Marták  |
|         | Vytvoriť vhodnú reprezentáciu plavidlového systému       | Aleš Mäsiar   |
|         | Grafická reprezentácia scén                              | Filip Šandor  |
|         | Vytvoriť stavový diagram hry                             | Zora Moravčíková  |

### 1.3.4 Lumos

V predchádzajúcich šprintoch sme vytvorili abstraktnú reprezentáciu našej hry na úrovni niekoľkých diagramov. Úlohou tohto šprintu ich bolo vhodne preniesť do existujúcej hry.

Cieľom je teda implementovať navrhnutý model vzájomnej interakcie objektov v scéne, zadefinovať na úrovni dát (použiť vhodnú reprezentáciu napr. xml súbor) jednoduchú interakciu s generickým objektom v prvej scéne. Napríklad po kliknutí na objekt by sa vyvolá interakcia a odpoveď na ňu bude zadefinovaná v tomto súbore a hráčovi sa v hre prejaví nejakou akciou napr. že sa objekt "predstaví". Teda v hre sa vypíše "ja som lampa".

Tabuľka 6: Zoznam úloh v štvrtom šprinte

| User Story                               | Úloha  | Člen tímu        |
|--|--|------------------|
| <b>Implementácia interakcie objektom</b> | Otestovať a spísať metodiku source control           | Zora Moravčíková |
|  | Upraviť diagram stavového priestoru 1. kapitoly      | Zora Moravčíková |
|  | Spísať metodiku pre notáciu stavového priestoru scén | Zora Moravčíková |
|  | Spísať postup pre tvorbu xml súborov                 | Aleš Mäsiar      |
|  | Definovať štruktúru aplikácie                        | Zora Moravčíková |
|  | Zadefinovať metodiku tvorby zdrojového kódu          | Lukáš Marták     |
|  | Implementácia herného modelu                         | Lukáš Marták     |
|  | Implementácia manažéra interakcií                    | Aleš Mäsiar      |
|  | Implementácia interpretera akcií                     | Lukáš Marták     |
|  | Implementácia herného kontrolera                     | Zora Moravčíková |
|  | Grafika scén   | Filip Šandor     |
|  | Animácia postavy                                     | Lukáš Miškovský  |
|  | Grafika hlavnej postavy                              | Daniela Hajdu    |
|  | Spísať postup pre tvorbu grafiky postáv              | Filip Šandor     |
|  | Špecifikácia presunu medzi scénami                   | Filip Šandor     |
|  | Implementácia interpretera akcií                     | Lukáš Miškovský  |

### 1.3.5 Rictumsempra

Hlavnou úlohou tohto šprintu je dokončiť jednoduché úlohy, ktoré nám ostali z predchádzajúceho šprintu a to konkrétne implementovať zobrazenie odpovede objektu na interakciu.

Tento šprint je zameraný na komunikáciu a interakciu hráča s avatarmi. Táto funkcionálnosť v našej hre je veľmi dôležitá a hlasový modul, ktorý pre nás vytvára SAV je už hotový – môžeme začať s implementáciou. Preto sme si vybrali niekoľko user stories týkajúcich sa tohto problému.

V šprinte budeme graficky navrhovať a vytvárať avatarov, ktorý budú v hre vystupovať ako jednoduchá ikona, implementujeme a doplníme dáta pre interakciu s avatarom nakoniec integrujeme komunikáciu so SAV. Výstupom tohto šprintu bude hra, v ktorej bude možné po kliknutí na ikonu avatarov vypočítať si predpísaný dialóg (avatari majú definovaný text, ktorý povedia).

V rámci tohto šprintu sa tím pripravuje aj na druhý kontrolný bod tímového projektu a to finalizáciou dokumentácie, preto sme si aj zadefinovali úlohy, ktoré túto činnosť pokrývajú.

Tabuľka 7: Zoznam úloh v piatom šprinte

| User Story                                  | Úloha  | Člen tímu  |
|---|--|--|
| <b>Implementácia interakcie s objektom</b>  | Definovať metodiku na testovanie a testovať                        | Lukáš Miškovský  |
|   | Grafika hádankárky   | Daniela Hajdu  |
|   | Implementácia dialógového okna                                     | Daniela Hajdu  |
|   | Implementácia dialógového okna                                     | Filip Šandor   |
|   | Grafický návrh avatara   | Daniela Hajdu  |
| <b>Inicializácia komunikácie s avatarmi</b> | Prekreslenie skice do RealDraw                                     | Daniela Hajdu  |
|   | Vytvorenie skriptu na nahrávanie zvuku                             | Lukáš Miškovský  |
|   | Analýza možností REST komunikácie v Unity                          | Zora Moravčíková   |
|   | Implementácia REST komunikácie so SAV                              | Lukáš Marták   |
| <b>Grafická reprezentácia avatarov</b>      | Implementácia znemožnenia iných akcií pri prebiehajúcej interakcii | Lukáš Marták   |
|   | Vytvorenie dialógovača   | Aleš Mäsiar  |
|   | Vytvorenie skriptu na zadávanie textu                              | Filip Šandor   |
| <b>Komunikácia s avatarom - predpísaná</b>  | Rozšírenie interpretera akcií                                      | Lukáš Marták   |
|   | Vylepšenie skriptu na zobrazovanie textu                           | Filip Šandor   |
| -   | Manažment projektu   | Lukáš Marták<br>Lukáš Miškovský<br>Filip Šandor<br>Zora Moravčíková<br>Daniela Hajdu |

### 1.3.6 Bombarda

V tomto šprinte sme sa zamerali na validáciu kódu a na rozšírenie funkcionality hry.



Pridali sme do hry inventár a implementovali nad ním funkcie ako pridávanie, odoberanie a posúvanie elementov. Spolu s tým bola vytvorená databáza objektov na uchovanie objektov z hry.

Tiež sme rozšírili dialógy avatarov o nové konverzácie a pridali ich do XML súborov. Ďalej sme pridali prenášanie vedomostí medzi scénami aby sa hráč mohol voľne prechádzať.

Rozšírili sme aj grafiku hry o novú postavu, vtipkárku.

Kvôli blížiacemu sa TP-Cup-u sme zhotovili abstract.

Tabuľka 8: Zoznam úloh v šiestom šprinte

| User Story                                 | Úloha  | Člen tímu        |
|--|--|------------------|
| -  | Reeviadacia a prezretie funkčnosti kódu      | Lukáš Miškovský  |
|  | Gramatická korekcia abstraktu TP-cup         | Daniela Hajdu    |
|  | Obsahová korekcia abstraktu TP-cup           | Aleš Mäsiar      |
| <b>Prechod medzi scénami</b>               | Doplniť XML                                  | Aleš Mäsiar      |
|  | Prenášanie informácií medzi scénami          | Zora Moravčíková |
|  | Pripraviť scény v Unity                      | Filip Šandor     |
|  | Prepínanie medzi scénami                     | Zora Moravčíková |
| <b>Komunikácia s Avatarom - predpísaná</b> | Rozšírenie dialógovača pre Avatárov          | Aleš Mäsiar      |
|  | Pridať dialógy s Avatarmi do dialógových XML | Aleš Mäsiar      |
| <b>Reprezentácia vtipkárky</b>             | Prekreslenie vtipkárky do digitálnej podoby  | Daniela Hajdu    |
| <b>Reprezentácia hádankárky</b>            |  | Daniela Hajdu    |
| <b>Inventár</b>                            | Pri zatváraní/otváraní inputu sa vypisuje    | Filip Šandor     |
|  | Vytvoriť grafickú reprezentáciu inventára    | Filip Šandor     |
|  | Vytvorenie kľúčových funkcií pre inventár    | Filip Šandor     |
|  | Ikony predmetov inventára                    | Daniela Hajdu    |
| <b>Obchod</b>                              | Vytvoriť prototyp obchodu                    | Lukáš Miškovský  |
| <b>Logovanie stavov hry</b>                |  | Lukáš Miškovský  |

### 1.3.7 Cruciatus

V tomto šprinte sme vylepšili grafiku scén a objektov v nej. Vytvorili sme základnú animáciu pre vtipkárku a hádankárku.

Vytvorili sme úvodné menu v ktorom sa dá nastaviť hlasitosť, zvuky a hudba.

Pridali sme otváranie inventáru a obchodu cez tlačidlá na obrazovke, takisto ikonka s avatarmi slúži na ich vyvolanie.

Navrhli a implementovali sme kľúčovp komunikáciu hry s modulov od SAV na preklad textu na reč a reči na text.

Rozšírili sme XML súbory o stavy a prechody medzi nimi.

Navrhli sme minihru ktorá bude slúžiť na získavanie peňazí za ktoré si bude môcť hráč niečo kúpiť, a tiež na pobavenie a odpútanie pozornosti od avatarov. Model pre túto hru sme neskôr aj implementovali a nakreslili pixelovú grafiku pre túto hru nech pripomína automatové hry z 80.-tych rokov.

Tabuľka 9: Zoznam úloh v siedmom šprinte

| User Story                               | Úloha                                     | Člen tímu        |
|--|---|------------------|
| -  | Prepracovať grafiku hry                   | Filip Šandor     |
|  | Vytvorenie animácie pre vtipkárku         | Daniela Hajdu    |
|  | Vytvorenie animácie pre avatara           | Daniela Hajdu    |
|  | Nastavenia                                | Lukáš Miškovský  |
|  | Vyvolanie inventáru cez ikony             | Filip Šandor     |
|  | Napísať SAV ohľadom prekladačov           | Zora Moravčíková |
| <b>Komunikácia s Avatarom - prototyp</b> | Návrh základného algoritmu chatbota       | Zora Moravčíková |
|  | Implementovanie vzorového dialógu         | Zora Moravčíková |
| <b>Správanie vtipkárky</b>               | Pridať vtipy do scén                      | Zora Moravčíková |
|  | Správanie vtipkárky                       | Zora Moravčíková |
|  | Aktualizácia SAV služieb                  | Zora Moravčíková |
| <b>Vyvolanie minihry cez automat</b>     | grafika                                   | Daniela Hajdu    |
| <b>Minihra - FlappyJoe</b>               | Minihra - model                           | Lukáš Marták     |
|  | Minihra - obsah                           | Lukáš Marták     |
|  | Minihra - grafika                         | Daniela Hajdu    |
| <b>Použitie predmetu v scéne</b>         | Implementácia stavov a akcií do inventáru | Aleš Mäsiar      |
|  | Pridanie stavov a prechodov do XML        | Aleš Mäsiar      |

### 1.3.8 Avada Kedavra

Pracovalo sa hlavne na správaní a dialógoch avatarov, ale aj hádankárky a magických hovoriacich dverí. Rozšírené boli viaceré moduly, a to interpreter akcií a dialógový manažér. Viaceré nové dialógy a interakcie boli zadefinované v rámci XML súborov. Tie bolo nutné rozšíriť, navrhnuť novú štruktúru a spôsob akým sa do nich budú zapisovať konverzácie.

Aby sme plne využili možnosti hlasového modulu poskytovaného SAV a zároveň spríjemnili henry zážitok rozhodli sme sa, že by bolo najlepšie aby každá postava, ktorá komunikuje verbálne mala vlastný špecifický hlas. Vykonávali sa aj úlohy potrebné na zabezpečenie postupu hráča do ďalších častí hry, konkrétne na zobrazenie tajného hesla a spustenie minihry.

Tabuľka 10: Zoznam úloh v ôsmom šprinte

| User Story | Úloha   | Člen tímu                         |
|------------|---|-----------------------------------|
| -          | Zobrazenie stavu peňazí                         | Lukáš Miškovský                   |
|            | Príchod avatara                                 | Lukáš Miškovský                   |
|            | Idle animácia – hádankárka, vtipkárka           | Lukáš Miškovský                   |
|            | Nemožné otvoriť inventár                        | Lukáš Miškovský                   |
|            | Posunutie obsahu inventára                      | Filip Šandor                      |
|            | Spojzdniť prechod medzi hlavnou hrou a minihrou | Lukáš Marták                      |
|            | Nepoužitý item sa má vrátiť do inventára        | Filip Šandor                      |
|            | Zpracovanie zmien v grafike                     | Filip Šandor                      |
|            | Animácia hádankárky                             |                                   |
|            | Generické postavy                               | Daniela Hajdu                     |
|            | Výmena postavy v minihre                        | Daniela Hajdu                     |
|            | <b>Správanie dverí</b>                          | Vloženie konverzácie s dverami do |

|  |  |                  |
|--|--|------------------|
|  | XML  |                  |
|  | Automatické prihovorenie po interakcií       | Aleš Mäsiar      |
| <b>Správanie hádankárky</b>                    | Rozšírenie dialógového manažéra              | Aleš Mäsiar      |
|  | Tadefinovanie prvej hádanky v XML            | Aleš Mäsiar      |
| <b>Animácia škriekajúcich dverí</b>            | Pripraviť animáciu dverí, ktoré pýtajú heslo | Filip Šandor     |
| <b>Komunikácia s Avatarmi – plná funkčnosť</b> | Rozlišovanie druhu odpovede                  | Zora Moravčíková |
|  | Odovzdanie rady                              | Zora Moravčíková |
|  | Zrušenie dialógu                             | Zora Moravčíková |
|  | Návrh a implementácia XML pre konverzácie    | Zora Moravčíková |
|  | Zmena hlasu hovoriacej postavy v konverzácií | Zora Moravčíková |
|  | Vloženie dialógu pre scénu 1 do XML          | Aleš Mäsiar      |
|  | Vyvolanie avatarov cez ikonu                 | Filip Šandor     |
|  | Doplnenie všetkých konverzácií               | Zora Moravčíková |
| <b>Zobrazenie tajného hesla</b>                | Rozšírenie interpretéra akcií                | Aleš Mäsiar      |
|  | Vloženie potrebných údajov do XML            | Aleš Mäsiar      |
| <b>Vyvolanie minihry cez automat</b>           |  | Aleš Mäsiar      |

### 1.3.9 Imperius

V rámci tohto šprintu bolo našim cieľom hlavne zostavenie funkčného a hrateľného prototypu hry. Z tohto dôvodu sme sa hlavne venovali odstráneniu pretrvávajúcej problémov. Šprint je preto zameraný primárne na opravovanie chýb a vylešovanie častí zaradených do prototypu.

Ďalej sa upravovali i rozširovali dialógy, prebehla všeobecná normalizácia zvukov v hre i pridanie nových zvukových efektov. Dialógy boli editované, správanie niektorých postáv aj naďalej vylepšované aby sa hráčovi zdali zaujímavejšie a ich hlasy boli zmenené.

Tiež sme sa venovali práci na ovládaní hry. To by malo byť jednoduché a umožniť hráčov vykonať presne tie akcie, ktoré potrebuje rýchlo a efektívne, preto bola obmedzená funkcionalita a možnosti interakcie hráča, zmenené klávesové skratky a vykonaných niekoľko ďalších úprav spojených s ovládaním. Boli napísané dva úvodné texty, a to úvod, ktorý slúži na uvedenie hráča do deja, a tiež prísaha, ktorá vopred zabezpečí získanie hlasovej vzorky hráča.

Zároveň sme sa však snažili tento prototyp aj rozšíriť pridaním rozhodnutí, z ktorých jedno predstavuje aj pomerne závažnú voľbu fakulty. Z implementáciou tohto rozhodnutia súvisela príprava potrebných grafických prvkov ako sú erby týchto dvoch fakúlt. Taktiež sa začali prípravy na druhú kapitolu, konkrétne vo forme náčrtu postavy Vlada, ktorý v nej bude predstavený.

Tabuľka 11: Zoznam úloh v deviatom šprinte

| User Story  | Úloha                                  | Člen tímu       |
|---|--|-----------------|
| <b>Oprava bugov a vylepšovanie do prototypovej verzie</b> | Normalizácia zvukov, pozadia, avatarov | Lukáš Miškovský |
|   | Pridanie zvukov do hry                 |                 |
|   | Slidre v nastaveniach                  | Filip Šandor    |
|   | Dokončenie logovania                   | Lukáš Miškovský |

|  |                  |
|--|------------------|
| Prísaha na začiatku  | Filip Šandor     |
| Zmena vš. klávesových skratiek a obmedzenie funkčnosti                           | Zora Moravčíková |
| Inventár v dialógoch zmeniť na batoh   | Zora Moravčíková |
| Blikanie Avatarov podľa toho, ktorý rozpráva                                     | Aleš Mäsiar      |
| Ikonka mikrofónu keď sa nahráva zvuk   | Lukáš Miškovský  |
| Poslovenčiť texty  | Zora Moravčíková |
| V prvej scéne dorobiť rozhodnutie s kľúčami                                      | Lukáš Miškovský  |
| Zmena hlasov rozprávajúcich postáv   | Zora Moravčíková |
| Úprava úvodného textu  | Filip Šandor     |
| Prílet avatarov vylepšiť   | Aleš Mäsiar      |
| Pripraviť základnú sadu otázok od hráča pre Avatara                              |                  |
| Odstránenie Cancel tlačidla pri dialógu  | Lukáš Miškovský  |
| Scéna 3: Obmedzenie vstupu do dverí  | Zora Moravčíková |
| Scéna 4: Úprava dialógov   | Zora Moravčíková |
| Scéna 4: Odstrániť samo rozprávanie Jozefa                                       | Zora Moravčíková |
| Scéna 4: Inventár úprava   | Lukáš Miškovský  |
| Scéna 5: Žaba výzor a rozhodnutie  | Zora Moravčíková |
| Prechod medzi scénami hore dole  | Zora Moravčíková |
| Scéna 6: Rozhodnutie na výber fakulty  |                  |
| Pridanie náповedy na klávesové skratky   | Aleš Mäsiar      |
| Scéna 2: Hádankárky – nemôže dávať 2-krát tú istú hádanku a zakaždým dať disketu | Lukáš Miškovský  |
| Všetky scény: poposúvať clickable areas  | Filip Šandor     |
| Pridať animáciu rozprávania dverí v scéne 3                                      | Filip Šandor     |
| Vylepšenie obchodu (otvoriť aj batoh)  | Filip Šandor     |
| Blikanie avatarov – pokazené   | Zora Moravčíková |
| Klikanie na UI elementy prechádza do vrstvy AdventureCreator-u                   | Aleš Mäsiar      |
| Minihra: Dorazil si na koniec aj keď si nedorazil                                | Zora Moravčíková |
| Scéna 1: Počas úvodného textu je povolený AC                                     | Zora Moravčíková |
| Otváranie inventára zároveň so shopom  | Zora Moravčíková |
| Minihra – pridanie zvukov k udalostiam   | Lukáš Marták     |
| Scéna 1: Problém pri získavaní predmetu z truhlice                               | Zora Moravčíková |
| Pridanie finálneho rozhodnutia   | Lukáš Marták     |
| Scéna 1: Na “rozumiem“ odpoveď “no nič, pochopíš neskôr“                         | Filip Šandor     |

|  |                         |               |
|--|-------------------------|---------------|
|  | Text prísahy            | Daniela Hajdu |
|  | Text intra              | Daniela Hajdu |
|  | Erby                    | Daniela Hajdu |
|  | Úprava Jozefovho plášťa | Daniela Hajdu |
|  | Náčrt Vlada             | Daniela Hajdu |

### 1.3.10 Finite Incantatum

V tomto šprinte sme aj naďalej vylepšovali prototyp vytvorený v šprinte predchádzajúcom. Veľká pozornosť sa venovala oprave dialógov a ovládania hry. Hrateľnosť aj naďalej zvyšujeme opravovaním chýb, ktoré sa prejavili pri opakovanom hraní aj pridávaním istých navigačných indikátorov. Napríklad meniaci sa kurzor myši hráčovi napovie, že kliknutím na danú oblasť sa presunie do ďalšej scény. Okrem týchto úloh sa neustále pracuje na vylepšovaní grafickej stránky hry.

V rámci príprav na implementáciu nasledujúcej kapitoly bola prerobená predošlá verzia grafickej reprezentácie zápornej postavy Vlada, tak aby bolo možné túto postavu animovať. Taktiež bol vytvorený náčrt a následne aj kompletná grafika jedného z profesorov školy, ktorý Jozefa učí.

Tabuľka 12: Zoznam úloh v desiatom šprinte

| User Story                                | Úloha  | Člen tímu        |
|---|--|------------------|
| <b>Oprava bugov z prototypovej verzie</b> | Zmeniť kurzor na šípku keď sa prechádza medzi scénami          | Aleš Mäsiar      |
|   | Flappy: - Skončiť hru niečím lepším – napríklad Enter          | Lukáš Miškovský  |
|   | Tutoriál chyby   | Aleš Mäsiar      |
|   | Dialógy: Nedovoliť pokračovať v dialógu dokým nedokecá         | Zora Moravčíková |
|   | Dialógy: texty   | Zora Moravčíková |
|   | Dialógy: ostatné bugy  | Zora Moravčíková |
|   | Scéna 1: Truhlica a kľúče                                      | Lukáš Marták     |
|   | Scéna 1: Úvodný text – posuň šípkami dať nižšie                | Filip Šandor     |
|   | Scéna 2: Dialógy texty lepšie                                  | Zora Moravčíková |
|   | Scéna 2: Podlaha má inú blbú perspektívu                       | Filip Šandor     |
|   | Scéna 3: Stredné dvere majú názov „Center door“ – niečo lepšie | Zora Moravčíková |
|   | Scéna 4: texty   | Zora Moravčíková |
|   | Scéna 6: návrat z 5ky  | Zora Moravčíková |
|   | Jozef chodí na mieste po začatí konverzácie                    | Aleš Mäsiar      |
|   | Ikona myši prekrýva output                                     | Lukáš Miškovský  |
|   | Prerobenie Vlada   | Daniela Hajdu    |
|   | Nakresliť profesora  | Daniela Hajdu    |

### 1.3.11 Immobulus

Okrem vykonávania posledných úprav a odstraňovania chýb z prvej kapitoly, sme v tomto šprinte začali aktívne pracovať aj na implementácii druhej kapitoly. Vytvorila sa grafika nových scén, zákutí školy, ktoré pred tým neboli hráčovi prístupné ako učebňa a Jozefova internátna izba. Rozšírený bol aj scenár a dialógy, tak aby zodpovedali pred tým načrtnutému príbehu. Medzi nové prvky tejto kapitoly patrí aj rušička, ktorá predstavuje jeden z kľúčových objektov príbehu. S implementáciou zápornej postavy Vlada je do hry vnesený konflikt.

Pokračovalo sa aj v práci na odstraňovaní chýb vyskytujúcich sa v prvej kapitole.

Tabuľka 13: Zoznam úloh v jedenástom šprinte

| User Story   | Úloha   | Člen tímu        |
|--|---|------------------|
| <b>Kapitola 2</b>                                    | Kapitola 1: ubytovanie dokončiť                           | Zora Moravčíková |
|  | Kapitola 2: izby kreslenie                                | Filip Šandor     |
|  | Kapitola 2: generické osoby                               |                  |
|  | Kapitola 2: Dialógy                                       | Zora Moravčíková |
|  | Prechod do izby   | Zora Moravčíková |
|  | Chodba do učebne/rušebne                                  | Filip Šandor     |
|  | Učebňa: keď mešká, tak trest                              |                  |
|  | Rozhodnutie s Vladom: Avatari mu poradia či ísť, či neísť | Zora Moravčíková |
|  | Kapitola 2 – nakreslenie učebne                           | Filip Šandor     |
|  | Rušička   | Daniela Hajdu    |
|  | Portál  | Daniela Hajdu    |
|  | <b>Oprava chýb hlásených od Števa</b>                     | Úvodný text      |
| Tutoriál: bug pri text input                         |   | Aleš Mäsiar      |
| Tutoriál: buggy                                      |   | Aleš Mäsiar      |
| Nahrávanie odpovede: bez kliku možnosť stlačiť R     |   | Zora Moravčíková |
| Blurovy Jozef a ostatné postavy                      |   | Filip Šandor     |
| ZVUKY  |   |                  |
| X tlačítko na skončenie                              |   | Filip Šandor     |
| Scéna 3: ústa a dvere                                |   | Filip Šandor     |
| Scéna 3: rozmazaný kľúč                              |   | Filip Šandor     |
| Scéna 2: Rečnicke otázky pri vtipkárke a hádankárke  |   | Zora Moravčíková |
| Normálne rady od avatarov striedať                   |   | Lukáš Miškovský  |
| Scéna 4: rada o lampášoch                            |   |                  |
| Scéna 5: rada žaba                                   |   | Zora Moravčíková |
| Prísaha: text  |   |                  |
| Scéna 2: dialóg s avatarmi                           |   | Aleš Mäsiar      |
| Scéna 3: animácia na kľúče                           |   | Lukáš Miškovský  |
| Dialógy: upraviť a pripraviť jeden zdieľaný dokument |   | Filip Šandor     |
| Nastavenia a úvodná scéna                            |   | Zora Moravčíková |
| ASR output okno                                      |   | Lukáš Marták     |
| Vylepšiť logovanie TTS a ASR                         |   | Aleš Mäsiar      |
| Pridať šípky na hotspots                             |   | Aleš Mäsiar      |
| Scéna 1: Keď chce odísť a nemá zviera v batohu       |   | Zora Moravčíková |

|  |  |                               |
|--|--|-------------------------------|
|  | MonWalk je iba konverzáciach                                     | Lukáš Miškovský               |
|  | Scéna 2: Porozprávaj sa s dievčatami aj keď sa rozpráva          | Zora Moravčíková              |
|  | Dialógy s avatarmi: podopíňať striedanie avatarov, ktorí hovoria | Lukáš Miškovský               |
|  | Scéna 3: Keď sa spýtajú na kľúče použiť text input               | Lukáš Miškovský               |
|  | Scéna 4: čo máš v batohu = otvoriť inventár                      | Lukáš Miškovský               |
|  | Mnohonásobné kliknutie na avatara                                |                               |
|  | Inventár: drag and click? ... nízka priorita                     | Filip Šandor                  |
|  | Scéna 4: vždy keď sa pýta na niečo v batohu alebo mince          |                               |
|  | Scéna 4: Neprispôsobený avatar pri rade                          | Zora Moravčíková              |
|  | Scéna 4: červený a zelený lampáš a farba textu                   | Lukáš Marták                  |
|  | Scéna 5: Náznak dverí keď zmiznú Singleton                       | Filip Šandor                  |
|  | Kapitola 1: Ubytovanie text                                      | Filip Šandor                  |
|  | Úvodný text kratšie  | Daniela Hajdu                 |
|  | Log: Párovanie rád na akcie                                      | Aleš Mäsiar                   |
|  | POSTER Priorita 1  | Zora Moravčíková              |
|  | Avatari – používať my  |                               |
|  | Odstrániť čiary v scénach  | Filip Šandor<br>Daniela Hajdu |
|  | Vypínanie/zapínanie interakcií AC                                | Aleš Mäsiar                   |
|  | Dve úrovne logovania   |                               |

### 1.3.12 Expecto patronum

Tento šprint práve prebieha, no plánujeme v rámci neho pracovať na dialógoch a príbehu druhej kapitoly, a tiež sa vysporiadať s novými chybami, ktoré vyplývajú z rozširovania príbehu. Taktiež sa venujeme zabezpečeniu kvalitného logovania užitočných dát, ktoré poslúžia ako vstup pre výskum.

Tabuľka 14: Zoznam úloh v dvanástom šprinte

| User Story                             | Úloha  | Člen tímu   |
|--|--|-------------|
| <b>Oprava chýb po druhom prototypu</b> | Logovanie: Vylepšiť pre Števa                                    | Aleš Mäsiar |
|  | Animácie: Znížiť rozlíšenia obrázkov, nech zaberajú menej miesta |             |
|  | Výber fakulty je možné urobiť viackrát                           |             |
|  | Text input: Ostáva palička v poli                                |             |

## 1.4 Používané metodiky

Pre riadenie tímu je dôležité mať zadefinované všetky postupy a metodiky, ktorými sa majú členovia riadiť pri riešení určitých úloh. Metodiky môžu pribúdať s časom riešenia projektu.

Každá metodika je vo vlastnom súbore a obsahuje stručne popísaný postup vykonania úlohy, všetky metodiky sú uložené v jednom priečinku. Presné znenie metodík sa nachádza v prílohe.

### **Metodika na tvorbu metodiky**

Stručný návod a niekoľko pravidiel na tvorbu metodiky.

### **Metodika ukladania súborov v zdieľanom priečinku OneDrive**

Popisuje hierarchiu úložiska, charakterizuje obsah každého priečinku. Definuje, kde ktoré dokumenty patria, pokyny k archivácii dokumentov.

### **Metodika na tvorbu grafických prvkov v hre**

Popisuje ako sa vyrábajú grafické prvky v hre. Najprv sa nakreslí náčrt na papier a keď je schválený, tak sa prekreslí do počítačovej podoby.

### **Metodika na prácu s úlohami**

Popisuje formálnu prácu s úlohami v rámci projektu. Obsahuje opis ich vytvárania a zadeľovania.

Zapísanie úloh zo zápisnice a ich vloženie do TFS. Následné pridelenie úloh členom tímu, odhad času a vedenie stavu úloh, až po ukončenie a zhodnotenie na stretnutí.

### **Metodika na tvorbu zápisníc**

Opisuje ako sa majú písať zápisnice a kde ich ukladať.

### **Metodika na tvorbu zdrojových kódov**

Stručne zhrňuje ako písať kód do projektu.

### **Metodika verziovania kódu**

Metodika, ktorá opisuje postup ako udržiavať svoj projekt. Obsahuje presný postup ako aktualizovať projekt, potvrdiť a nahráť lokálne zmeny.

### **Súborová štruktúra Unity projektu**

Projekt v nástroji Unity sa skladá z niekoľkých priečinkov, ktoré sú dostupné priamo z rozhrania nástroja. Sú to štandardné priečinky, ale aj také, ktoré vývojár sám vytvorí, z tohto dôvodu máme zadanú túto metodiku. Jej úlohou je pomôcť programátorovi identifikovať kde má vkladať svoje súbory tak, aby ich ďalší kolegovia mohli nájsť.

### **Metodika na tvorbu testov - UnityTestTools**

Opisuje ako pripraviť a vykonávať testy v Unity pomocou UnityTestTools.

### **Metodika rizík**

Popisuje hlavné identifikované riziká a takisto aj možnosti ako sa s týmito rizikami vysporiadať



## 1.5 Retrospektíva

### 1.5.1 Oculus reparo

Vzhľadom na typ nášho projektu sme nerobili kompletnú retrospektívu prvého šprintu, pretože úlohy, ktoré sme v ňom vykonávali neboli časovo odhadované a často sme ich vykonávali spoločne. Bolo potrebné vymyslieť a navrhnuť princíp, príbeh a scenár hry.

### 1.5.2 Expelliarmus

Druhý šprint už obsahoval aj implementačné úlohy a tieto mali svoje časové ohodnotenie. Na konci šprintu sme si spravili rýchlu retrospektívu.

Zistili sme, že odhad jednotlivých úloh bol často príliš optimistický najmä preto, že veľa vecí, s ktorými sme robili bolo pre nás úplne nových. Našli sa aj úlohy, ktoré sme museli preniesť do ďalšieho šprintu, pretože pri ich vykonávaní sa vyskytli problémy, ktoré bolo treba prekonzultovať so zvyškom tímu.

Aby sme v ďalších šprintoch predišli takýmto problémom rozhodli sme, aby úlohy, ktoré obsahujú prácu s novými nástrojmi boli ohodnotené ako zložitejšie a člen tímu, ktorý ich má riešiť tento fakt bral do úvahy a podľa toho nastavil svoj odhadovaný čas aspoň dvojnásobok pôvodného odhadu.

V prípade, ak sa prenášajú úlohy medzi šprintami z dôvodu, že sa nestihli vykomunikovať medzi členmi tímu, rozhodli sme sa takto prenesené úlohy prideliť vždy minimálne aspoň dvom členom, aby mohli medzi sebou následne diskutovať.

### 1.5.3 Lumos

Tretí šprint bol zameraný hlavne na implementáciu navrhutej hry a vytvorenie jednoduchej interakcie medzi hlavnou postavou a objektom. Zadefinované úlohy boli takmer všetky splnené okrem jednoduchých úloh, ako je vypísanie textu (odpovede od objektu) na obrazovku. Tento problém vznikol kvôli nedokonalému odhadu zaneprázdnenosti člena tímu, ktorý ho mal pridelený.

V rámci šprintu sme identifikovali aj problém, keď úlohy medzi sebou na seba nadväzujú, nemali sme prioritne dané poradie úloh v akom sa majú urobiť, preto niektorí členovia nemohli vykonávať svoju úlohu, kým čakali na dokončenie inej. S týmto problémom sme bojovali v strede šprintu, kedy sa nedokončili úlohy medzi stretnutiami presne kvôli tomuto problému.

Pre predídanie takýchto zaseknutí členov pri plnení ich úloh, sme sa rozhodli si explicitne definovať presný čas, kedy bude úloha dokončená v rámci týždňa. Tento čas si určí na stretnutí jej riešiteľ podľa vlastného zváženia osobného harmonogramu a ak s tým ostatní členovia súhlasia, zapíše sa do opisu úlohy v TFS – tento termín musí byť následne dodržaný.

### 1.5.4 Rictumsempra

Zistili sme že robenie unit testov v unity nieje možné v spojení s doplnkom Adventure Creator a rozhodli sme sa že budeme pravidelne posielat' export hry zadávateľovi na schválenie a spätnú väzbu čo treba zmeniť/doplniť.

Veľa času nám zobrala aj finalizácia dokumentácie, zhodli sme sa na tom že budeme dokumentáciu tvoriť postupne aby sme sa tomuto vyvarovali.

Úspešne sa nám podarilo implementovať okno na dynamický výpis textu na obrazovku. Ďalej sme nakreslili nové postavy do hry a graficky reprezentovali avatara.

Implementovali sme aj komunikačný model zo SAV na preklad textu na reč a reči na text. Pri tomto bode sme sa dohodli že bude dôležitá vzájomná komunikácia medzi nami a SAV aby sme daný modul mohli spoločne prispôbovať a vylepšovať.

### 1.5.5 Bombarda

Tento šprint prebiehal hladko, všetky úlohy čo sme si zadefinovali sa podarilo úspešne splniť.

Zároveň sme začali pracovať na logovaní stavov hry. Tu sme si definovali že bude potrebné dohodnúť sa zo zadávateľom v akom formáte potrebuje výsledné dáta.

### 1.5.6 Cruciatu

Zistili sme že rozpoznávanie reči nefunguje správne, a treba do modulu rozpoznávača doplniť niektoré slová. Stanovili sme člena tímu ktorí bude komunikovať zo SAV ohľadom vylepšovania a dopĺňovania hlasového modelu.

### 1.5.7 Avada Kedavra

V tomto šprinte sme zistili že je veľmi dôležité mať predchádzajúce skúsenosti pred odhadovaním časovej zložitosti úloh. Úloha s grafickým animovaním dverí ktorá sa najskôr javila na max 30 minút nakoniec zabrala niekoľkonásobne viac.

Ostatné úlohy boli spravené bez väčších komplikácií.

Identifikovali sme riziko že kľúčový aspekt ktorým je konverzácia a dialógy ktoré sú reprezentované v XML súboroch ovláda len jeden až dvaja členovia tímu. To by mohlo spôsobiť problémy a zdržať nás v postupe. Rozhodli sme sa teda že budeme mať školenie kde sa všetci zaškolia do toho ako používať XML súbory s dialógmi.

### 1.5.8 Imperius

V tomto šprinte sme hru pripravovali na prototypové odovzdanie zadávateľovi. Snažili sme sa ukončiť prvú kapitolu hry a dorobiť všetky chýbajúce aspekty.

Zároveň sa začali prípravy na druhú kapitolu ako predbežné nákresy novej postavy a náčrt dejovej línie.

### 1.5.9 Finite Incantatum

Primárne sme sa v tomto šprinte venovali zdokonaľovaniu prototypu, odstraňovaniu pretrvávajúcich i nových problémov s dialógmi, grafikou či ovládaním. Paralelne prebiehali aj prípravy ďalších materiálov pre druhú kapitolu, ako vytvorenie ďalšej postavy a nová iterácia charakteru z predošlého šprintu. Pri opakovanom hraní hry neustále objavujeme nové chyby, ktoré je nutné riešiť, aj keď ich výskyt nie je častý, no často spôsobí nemožnosť hry, a teda aj hráča, pokračovať ďalej.

Po odovzdaní prototypovej verzii sme dostali od zadávateľa pripomienky a nahlásené chyby ktoré sme v tomto šprinte odstraňovali.

Súbežne prebiehala príprava scén a postáv pre druhú kapitolu

### 1.5.10 Immobulus

Po konferencii IITSRC sme dostali kopec spätnej väzby a námetov na vylepšenie. Tieto pripomienky sme si prešli a kľúčové prvky sme sa rozhodli implementovať do našej hry. Zároveň nám prišli aj nejaké pripomienky od zadávateľa.

Taktiež sa menila komunikácia s hlasovým modelom, pribudli parametre ako rýchlosť reči, hlasitosť a iné. Tieto zmeny sme úspešne implementovali do našej hry a otestovali.

V rámci tohto šprintu sme začali venovať implementácií druhej kapitoly zvýšenú pozornosť. Implementovali sme viaceré predpripravené prvky tejto kapitoly a zároveň bolo nutné zabezpečiť jednoduchý a zreteľný prechod medzi existujúcimi a novými lokalitami.

Počas celého šprintu tiež prebiehalo aktívne zapracovávanie návrhov a oprava chýb hlásených externým zadávateľom projektu, ktorému bol v minulých šprintoch odovzdaný pototyp na testovanie. Táto skutočnosť náš pokrok pri vytváraní druhej kapitoly aj naďalej spomaľuje.

## 1.6 Zoznam kompetencií členov tímu

Každý člen tímu má bohaté skúsenosti s tvorbou softvéru a navrhovaním architektúry, ktoré získal buď počas školy alebo v rámci zamestnania. Okrem toho máme všetci skúsenosti s prácou nad databázami, s procedurálnym programovaním a taktiež aj s návrhom a implementáciou softvéru v objektovo-orientovaných jazykoch. Avšak rôzni členovia majú rôzne skúsenosti a tak sa navzájom dopĺňame.

### **Bc. Daniela Hajdu**

Manažér komunikácie

Vo voľnom čase, ale aj v práci sa venuje tvorbe web stránok (WordPress) a zároveň skvele ovláda anglický jazyk a to aj vďaka ročnému študijnému pobytu v USA. Mimo iné ovláda aj C, Java, Ruby a assembly.

### **Bc. Lukáš Marták**

Manažér kvality

Zaujíma sa a špecializuje na aplikáciu metód a prostriedkov umelej inteligencie v robotike, automatizácii, predikovaní či rozhodovaní. Z praxe má skúsenosti s agilným vývojom a prácou v tíme. Medzi jeho záujmy patrí umelá inteligencia a taktiež hudba. Pracoval už s nástrojmi ako java, ruby on rails, c, postgresql, julia.

### **Bc. Aleš Mäsiar**

Manažér dokumentácie

Potrpí si na tom, aby všetko bolo zaznamenané a malo správnu formu. Zároveň sa vo voľnom čase venuje umeniu, takže do tímu prináša trochu odlišný a inak zameraný pohľad na viaceré problémy. Praktické skúsenosti má s technológiami java, c#, ruby on rails, c, html, css, javascript, postgresql, taktiež s vývojom webovej hry.

### **Bc. Lukáš Miškovský**

Manažér rizík

Dokázal svoju šikovnosť na výskumne orientovanom seminári, kde sa v relatívne krátkom čase naučil Ruby on Rails, z práce ma praktické skúsenosti s písaním odbornej dokumentácie a testovania. Mimo iné ovláda aj C, Java, Ruby, postgresql a assembly.

### **Bc. Zora Moravčíková**

Manažér plánovania

Má skúsenosti v oblasti vývoja hier pre špecifického používateľa konkrétne v Unity frameworku a prácou s pohybovým senzorom Kinect. Pracovala s technológiami ako java, c, c#, html, css, javascript - angular framework, postgresql, sqlite.

### **Bc. Filip Šandor**

Vedúci tímu

Má skúsenosti s 3D modelovaním, o ktorom aj vedie workshopy. Vo voľnom čase za zaujíma o vývojový engine Unity 3D, v ktorom vytvoril aj jednoduchú hru. Znalosť C, Javy, Ruby a assemblerov je pre neho samozrejmosťou.

## 1.7 Motivácia jednotlivých trojíc pre výber témy

### 1.7.1 Trojica č. 20

Bc. Daniela Hajdu, Bc. Lukáš Miskovský, Bc. Filip Šandor

#### **Trojica -- Prečo my? Lebo...**

“Stretnutie je začiatok, súdržnosť je pokrok a spolupráca je úspech.” - Henry Ford

Každý z nás sa venuje rôznym odvetviam v rámci informatiky a to nás paradoxne spája. Poháňa nás chuť naučiť sa niečo nové, pričom sme si pri tom vzájomne nápomocní. Filip má skúsenosti s 3D modelovaním, o ktorom aj vedie workshop-y, animáciami a vývojovým game engine-om Unity 3D, v ktorom vytvoril aj jednoduchú first-person shooter hru. Lukáš dokázal svoju šikovnosť aj na výskumne orientovanom seminári, kde sa v relatívne krátkom čase naučil Ruby on Rails, z práce ma praktické skúsenosti s písaním odbornej dokumentácie a testovania. Daniela sa vo voľnom čase, ale aj v práci venuje tvorbe web stránok (WordPress) a zároveň skvele ovláda anglický jazyk a to aj vďaka ročnému študijnému pobytu v USA. Znalosť C, Javy, Ruby a assemblerov je pre nás samozrejmosťou. Naše znalosti C++ sú na základnej úrovni, avšak spoločnými silami sa snažíme ďalej v tomto smere vzdelávať.

Učíme sa pre život, a preto...

Vybrali sme si predmety, ktoré nás nie len zaujímajú, ale nám aj pomôžu pri tímovom projekte či budúcom zamestnaní a naučíme sa pri nich niečo nové. V súvislosti s tímovým projektom - Počítačová hra: Adventure and Trust in Speech už máme naštudovanú Umelú inteligenciu a Princípy počítačovej grafiky a spracovania obrazu. Okrem hore spomínaných predmetov máme zapísanú Vizualizáciu dát a Počítačové videnie, ktoré nám môžu taktiež pomôcť pri projektoch: Vizualizácia informácií v obohatenej realite a Interakcia a kolaborácia vo virtuálnej realite. Taktiež sme si už stihli osvojiť teoretickú stránku technológie Kinect senzor, avšak v najbližších dňoch sa chceme s touto technológiou bližšie zoznámiť prostredníctvom praktického testovania.

#### Motivácia -- Počítačová hra: Adventure and Trust in Speech [TraSpi]

K tomuto projektu nás na počiatku samozrejme pritiahol najmä fakt, že sa jedná o počítačovú hru. Tie sú nám totiž všetkým dôverne blízke. Po podrobnejšom preskúmaní zadania sa však náš záujem ešte viac prehĺbil. Jednak by sme mohli pri práci uplatniť naše rozsiahle znalosti jazyka Java, ale taktiež využiť a prehĺbiť si vedomosti z oblasti umelej inteligencie, o ktorú sa aktívne zaujímame (bakalárske práce dvoch členov nášho tímu sa týkali práve oblasti umelej inteligencie).

Radi by sme sa tiež bližšie oboznámili s technikami analýzy hlasu, keďže je to progresívna oblasť so širokou škálou uplatnení. Tento projekt nás oslovil taktiež z dôvodu, že v porovnaní s mnohými inými, poskytuje veľký priestor pre sebarealizáciu a rozvoj kreatívnych nápadov, napríklad prostredníctvom tvorby avatarov, s ktorými bude hráč prichádzať do kontaktu.

Jednou z nesporných výhod nášho tímu je, že jeden z členov, Filip, sa na tvorbe počítačovej hry už v minulosti podieľal, a preto má zručnosti spojené s grafikou a modelovaním. Vzhľadom na to, že sa jedná o medzinárodný projekt, dôležitú rolu určite zohráva aj komunikácia so zainteresovanými zahraničnými inštitúciami. Tento aspekt pre nás

nepredstavuje problém, keďže v rámci tímu aktívne ovládame angličtinu technickú aj hovorovú, a to na vysokej úrovni.

Motivácia -- Vizualizácia informácií v obohatenej realite [AugReality]

Téma vizualizácie informácií v obohatenej realite nás upútala z viacerých dôvodov. Hlavne by sme radi pracovali práve na nových, inovatívnych projektoch, ktoré posúvajú možnosti informačných technológií.

Vo voľnom čase sa niektorí z členov nášho tímu venujú modelovaniu 3D objektov a medzi naše záujmy taktiež patrí sledovanie vývoja nových technológií, ako napríklad Microsoft Hololens, Intel RealSense a Leap Motion senzor. Aj vďaka tomuto aktívnemu záujmu už máme utvorené základné predstavy o princípoch, na ktorých sú takéto technológie založené, o možných spôsoboch interakcie, ktoré poskytujú, a tiež o vzťahoch charakteristických pre 3D priestor a ich transformáciu do priestoru počítača, čiže priestoru 2D.

Súčasťou nášho tímu je aj študent, ktorý absolvoval predmet Výskumne orientovaný seminár a preto veríme, že výzvu, ktorú prináša výskumný charakter tohto projektu, dokážeme spolu bez problémov zvládnuť. Lukáš sa tiež zúčastnil prezentácie jedného zo študentov našej fakulty, ktorý sa zaoberal podobnou tematikou. Tam mal možnosť si v praxi vyskúšať manipuláciu s Kinect senzorom alebo aj mobilnú aplikáciu využívajúcu práve obohatenú realitu na poskytovanie informácií o snímaných obrazoch.

Motivácia -- Interakcia a kolaborácia vo virtuálnej realite [VR-Collab]

Virtuálna realita je v súčasnej dobe nie len populárna, ale aj aktuálna. Komunikácia je však rovnako dôležitá vo virtuálnom priestore ako v reálnom živote, a preto nás zaujal práve tento projekt zameraný na interakciu. Máme možnosť využiť naše skúsenosti z oblasti počítačovej grafiky a modelovania objektov. V blízkej dobe by sme chceli otestovať možnosti Kinect senzoru na skenovanie a tvorbu 3D modelov postavy, a zaujímavá by bola možnosť takto vytvoriť avatara.

## 1.7.2 Trojica č. 25

Zora Moravčíková, Aleš Mäsiar, Lukáš Marták – Manažér kvality

### **Predstavenie tímu**

Tímový projekt si predstavujeme ako spoluprácu niekoľkých členov tímu, pričom každý z nich zastáva iné rôzne funkcie, ktoré sú v projekte potrebné. Princíp tímovej práce podľa nás spočíva v rozdelení úloh medzi členov tak, aby po splnení úloh boli výsledky jednotným prínosom a zároveň želaným výstupom projektu. Preto je dôležité mať v tíme na každý typ úlohy člena tímu, ktorý vyniká v plnení úloh daného typu. Správne rozdelenie funkcií podľa zručností členov je podľa nás nutnou podmienkou existencie pre funkčný a efektívny tím.

Sila nášho tímu spočíva v rozmanitosti schopností a skúseností jeho členov. Zvládame rozličné technologické prostriedky a rôzne spôsoby a metódy práce, no zároveň sme všetci tvoriví a máme rovnako perfekcionistický a systematický prístup k práci.

Časť tímu uprednostňuje webové technológie pre implementáciu používateľského rozhrania najmä pre rýchlu a univerzálnu dostupnosť obsahu. Iná časť tímu má väčšie skúsenosti s vývojom používateľského prostredia na úrovni mobilných aplikácií. Okrem toho máme všetci skúsenosti s prácou nad databázami, s procedurálnym programovaním a taktiež aj s návrhom a implementáciou softvéru v objektovo-orientovaných jazykoch.

Zatiaľ čo časť tímu má špecifickejšie znalosti v oblasti psychológie, použiteľnosti a vývoja používateľského prostredia a hier, iná časť sa viac zaujíma a špecializuje na aplikáciu

metód a prostriedkov umelej inteligencie v robotike, automatizácii, predikovaní či rozhodovaní.

Členovia nášho tímu už konkrétne pracovali s technológiami java, c#, ruby on rails, c, html, css, javascript - angular framework, postgresql, sqlite, oracle sql, pl/sql, julia. Máme skúsenosti s vývojom hier, 2D aj 3D v hernom engine Unity3D s použitím senzoru Kinect. Predmety, ktoré máme zapísané v prvom ročníku korešpondujú zameraniu na oblasti umelej inteligencie, počítačovej grafiky, strojového videnia a spracovania obrazu.

Naším spoločným menovateľom je práve prístup k práci. Keď niečo robíme, snažíme sa to robiť správne, udržateľne a efektívne. Písať čitateľný a prehľadný kód aj dokumentáciu a robiť veci modulárne. Ak zistíme, že náš postup v riešení problému je nesprávny alebo neefektívny, snažíme sa poučiť z našich doterajších chýb a nájsť inú, lepšiu cestu k riešeniu. Kladieme pri práci dôraz na detail a spoľahlivosť, pretože ak chceme očakávať, že sa môžeme spoľahnúť na prácu iných, musíme aj my pracovať tak, aby sa ostatní mohli spoľahnúť na tú našu.

Za prínosnú považujeme aj skutočnosť, že všetci členovia tímu sa dobre poznajú, preto nemajú problém s komunikáciou týkajúcou sa pracovných, ale aj iných tém a taktiež počas doterajšieho štúdia už spolupracovali na viacerých projektoch, ktoré mali tímový charakter, takže už majú určité skúsenosti v tímovej práci medzi sebou. Zároveň, okrem pracovných kvalít a podobného prístupu k riešeniu problémov a študijných záležitostí, nás spájajú aj spoločné záľuby, z ktorých vyniká napríklad aktívny záujem všetkých členov o umenie a kultúru. Vďaka tomu, no aj vďaka nášmu spoločnému rozhodnutiu a zdravému zápalu k práci teda očakávame, že náš tím bude úspešne fungovať ako na úrovni jednotlivcov, tak aj ako celok na úrovni vzájomnej spolupráce.

### **[TraSpi] Počítačová hra: Adventure and Trust in Speech**

Táto téma nás zaujala najmä tým, aký obrovský priestor ponecháva našej fantázií a tvorivosti. Všetci členovia tímu sú umelecky nadaní a ich tvorivosť a kreativita sa okrem iného prejavuje aj v iných disciplínach ako je IT (konkrétne: hudba 2ks, tanec 1ks). Naša kreativita by sa tým pádom mohla a chcela prejaviť napríklad aj pri tvorbe príbehu a iného netechnického obsahu projektu.

K projektu vieme poskytnúť skúsenosti s tvorbou počítačovej hry. Dvaja z troch členov tímu v rámci bakalárskeho projektu vyvíjali hru a naučili sa okrem iného aj ako pracovať s používateľmi, čo je to použiteľnosť hry, ako motivovať hráča, majú nastudované rôzne materiály ohľadom vývoja hier. Muzikantsky orientovaná časť tímu je taktiež otvorená myšlienke tvorby soundtracku ku hre. Máme tiež skúsenosti a dostupné prostriedky na tvorbu hudobných nahrávok štúdiovej kvality.

Tretí z troch členov nášho tímu má už od čias predmetu Umelá Inteligencia veľký záujem o oblasť a metódy umelej inteligencie a rád investuje čas a energiu do implementácie jej prostriedkov v rámci tohto projektu. Ak je práca na projekte zaujímavá, potreba spánku väčšinou prehráva súboj s potrebou tvoriť.

Hlavná myšlienka hry a nápad pracovať s intonáciou ľudskej reči nám prídu veľmi zaujímavé. Vývoj hry považujeme za zaujímavý, tvorivý a motivujúci proces.

### **[Chamelleon] Chamelleon**

V dnešnom svete, kde je všetkého dostatok, majú ľudia potrebu vystrčiť sa z davu a všetko čo využívajú si prispôbiť svojim potrebám. Myšlienka dynamického prispôbovania dizajnu

stránky je zaujímavá a keďže sme už viackrát narazili na problémy s rozličným vnímaním použiteľnosti webových stránok, myslíme si že má tento nápad veľký potenciál.

Táto téma je pre náš tím zaujímavá aj vďaka tomu, že existujú už mnohé výskumy na základe, ktorých sa dá rýchlejšie analyzovať problematika a navrhnúť vhodné riešenie s konečnou podobou vo forme ľahko použiteľnej aplikácie. Dvaja z členov tímu majú záujem o tvorbu webov, či webových aplikácií aj najmä preto, že je to nevyhnutná súčasť dnešného života, oceňujú jednoduchú dostupnosť obsahu na Webe a majú, s takýmto vývojom skúsenosti. Zatiaľ čo tretí člen vie poskytnúť oporu v spracovaní veľkého množstva údajov a dát, s ktorým sa počas vývoja určite stretne.

V rámci bakalárskych prác sme sa stretli s rôznymi používateľmi našich aplikácií pri testovaní a nazbierané poznatky vieme využiť počas testovania Chamelleonu, čo bude veľmi dôležité, pretože budeme musieť zväziť používateľov rôznych skupín.

Zmazanie bariéry medzi skúseným a neskúseným používateľom poskytuje výzvu, do ktorej sa radi pustíme.

### **[DronSim] Simulácia správania UAV v roji**

Téma tohto projektu je pre náš tím atraktívna najmä kvôli vzťahu k použiteľnosti v praxi. Tím dronov, ktorý spolupracuje a komunikuje pri plnení misie je použiteľný pre množstvo rôznych úloh a situácií.

Prostredie simulácie poskytuje možnosti implementácie rôznych zaujímavých algoritmov, či prvkov umelej inteligencie ako strojové učenie, rozoznávanie objektov a situácií (UNS), alebo rozhodovanie sa a implikovanie nových informácií v reálnom čase s použitím expertných systémov.

Náš tím vie projektu poskytnúť skúsenosti s vývojom vo frameworku Unity3D a s definovaním správania virtuálnych objektov vo virtuálnom prostredí. Tieto skúsenosti plynú z práce na bakalárskom projekte a z tvorby definícií správania virtuálnych entít v rôznych PC hrách, ktoré túto možnosť definície poskytovali.

### **1.7.3 Usporiadané zoznamy tém podľa priorít**

V rámci motivačných dokumentov oboch tímov musela byť na konci dokumentu aj tabuľka s usporiadaným zoznamom podľa priority všetkých tém. Tento zoznam pre oba tímy sa nachádza v tabuľke č. 7.

Tabuľka 15: Usporiadané zoznamy tém podľa priorit

| Číslo<br>témy | Poradie trojice č. 20 | Poradie trojice č. 25 |
|---------------|-----------------------|-----------------------|
| 1.            | TraSpi                | TraSpi                |
| 2.            | AugReality            | Chamelleon            |
| 3.            | VR-Collab             | DronSim               |
| 4.            | VisitorTrack          | VisitorTrack          |
| 5.            | WebTest               | UX-WEB                |
| 6.            | DronSim               | OpenScience           |
| 7.            | OpenScience           | WebTest               |
| 8.            | 3D-recon              | Askalot2edX           |
| 9.            | DevAct                | VR-Collab             |
| 10.           | Askalot2edX           | AugReality            |
| 11.           | Chamelleon            | DevAct                |
| 12.           | IndoorNav             | 3D-recon              |
| 13.           | LinkedResearch        | LinkedResearch        |
| 14.           | UX-WEB                | IndoorNav             |
| 15.           | 3D-UML                | FunCan                |
| 16.           | FunCan                | 3D-UML                |
| 17.           | 3D-Futbal             | 3D-Futbal             |
| 18.           | VirtNET               | InvisibleWifi         |
| 19.           | InvisibleWifi         | LTE2VoIP              |
| 20.           | LTE2VoIP              | VirtNET               |



## 2 Metodiky

---

Kapitola obsahuje presné znenie metodík, ktoré boli stručne opísané v prvej kapitole.

### 2.1 Metodika na tvorbu metodiky

#### 2.1.1 Role a zodpovednosti

|             |                                |
|-------------|--------------------------------|
| Rola        | Zodpovednosť                   |
| Zapisovateľ | Zapíše metodiku                |
| Tím         | Identifikácia potreby metodiky |

#### 2.1.2 Proces splnenia úlohy

|    |                                      |
|----|--------------------------------------|
| 1. | Identifikácia potreby metodiky       |
| 2. | Identifikácia účastníkov na metodike |
| 3. | Identifikácia jednotlivých procesov  |
| 4. | Opis procesov                        |
| 5. | Zverejnenie                          |

##### 2.1.2.1 Identifikácia potreby metodiky

Vstup: Činnosť

Výstup: Identifikovaná úloha

Rola: Tím

Tím svojou aktivitou identifikuje úlohu/činnosť, ktorá sa vykonáva častejšie rôznymi členmi tímu. Preto sa dohodne na konkrétnom postupe ako sa bude táto úloha bude riešiť a určí zapisovateľ'a.

##### 2.1.2.2 Identifikácia účastníkov na metodike

Vstup: Identifikovaná úloha

Výstup: Účastníci

Rola: Tím, zapisovateľ

Tím spolu so zapisovateľom určí kto sa zúčastňuje na danej úlohe a akú zodpovednosť má.

##### 2.1.2.3 Identifikácia jednotlivých procesov

Vstup: Identifikovaná úloha

Výstup: Jednotlivé procesy

Rola: Tím

Tím spolu so zapisovateľom určí jednotlivé kroky a procesy v danej úlohe.

##### 2.1.2.4 Opis procesov

Vstup: Jednotlivé procesy

Výstup: Opis konkrétnych procesov

Rola: Tím

Identifikované procesy treba konkrétne a jasne opísať. Kto kedy a prečo vykonáva daný proces.

##### 2.1.2.5 Zverejnenie

Vstup: Opis úlohy

Výstup: Dokument s metodikou

Rola: Zapisovateľ

Následne sa všetky časti úlohy ako sú účastníci, procesy a ich opis spíšu do jedného dokumentu a uložia/ zverejnia na určené miesto na úložisku OneDrive.

## 2.2 Metodika ukladania súborov v zdieľanom priečinku OneDrive

### 2.2.1 Pojmy

- OneDrive - používaný cloud priestor od Microsoftu

### 2.2.2 Role a zodpovednosti

- Metodika je platná pre všetkých členov tímu

### 2.2.3 Onedrive

Na zdieľanie dokumentov, iných ako zdrojové kódy, používame jeden zdieľaný priečinok na OneDrive serveri. Každý člen tímu ma práva upravovať priečinok (pozvánka bola poslaná na mail člena)

V priečinku je povinné dodržiavať základnú stromovú štruktúru uloženia dokumentov. Hlavný priečinok sa volá Patronus, tento je zdieľaný a upravovaný. Pre zjednodušenie používame na názvy súborov a priečinkov slovenský jazyk.

### 2.2.4 Stromová štruktúra

- Patronus
  - o Dokumenty
    - **Architektúra** – obsahuje vytvorené diagramy na popis architektúry hry
    - **Dokumentácia** – obsahuje výsledný dokument Dokumentácia inžinierskeho diela, Dokumentácia riadenia, obsahuje aj dočasné súbory na poznámky týkajúce sa dokumentácie
      - **Metodiky** – priečinok so všetkými používanými metodikami, každá metodika je vo vlastnom Word dokumente
    - **Hra,scenare,nápady** – obsahuje všetky súbory týkajúce sa vymýšľania príbehu, návrhu scenára, deja hry, dialógy
    - **TP Cup** – všetky dokumenty týkajúce sa súťaže TP Cup
    - **TFS & Unity, návody** – obsahuje návod ako správne nainštalovať a spustiť Unity projekt, ako nastaviť TFS na zdieľanie kódu, prípadne iné návody
    - **Zápisnice** – všetky zápisy stretnutí oficiálnych aj neoficiálnych (skype hovory)
  - o Fotografie
    - **Projekt** – fotky odfotené počas stretnutí, napr. náčrty na tabuli
    - **My** – dokumentačné fotky nás pri tvorení
    -
  - o Grafika
    - **Logo** – Návrhy, exportované obrázky a rozpracované projekty s logom
    - **Modely** – Model hlavnej postavy 3D aj 2D, rôzne objekty používané v hre, exportované modely ale aj projekty
    - **Scény** – Nákresy scén použitých ako pozadie v hre
    - **Skice** – všetko čo bolo pred kreslené na papier do hry, v digitálnej forme (sken)
  - o **Programy** a pluginy – programy a pluginy potrebné alebo voliteľné pre projekt

- **Temp** – Dočasné súbory, nemusia priamo súvisieť s projektom
- **Web** – Súbory potrebné pre chod stránky

## 2.3 Názvy šprintov

### 2.3.1 Šprinty

V rámci tímu pracujeme na dvojtýždňových šprintoch.

Každý šprint má jednoznačný názov (zaklínadlo) zo zoznamu vypísaného nižšie

| <b>Názov</b>      | <b>Vysvetlenie</b> |
|-------------------|--------------------|
| Accio             | privolá            |
| Alohomora         | odomkne            |
| Expelliarmus      | odzbrojí           |
| Lumos             | rozsvieti          |
| Rictusempra       | rozosmeje          |
| Bombarda          | zničí              |
| Cruciatus         | umučí              |
| Avada Kedavra     | zabije             |
| Imperius          | ovláadne           |
| Finite Incantatum | zruší kúzla        |
| Immobulus         | znehyní            |
| Expecto Patronum  | patronus           |

## 2.4 Metodika na tvorbu grafických prvkov do hry

### 2.4.1 Pojmy

- .rwd – RealDraw súbor

### 2.4.2 Role a zodpovednosti

| rola                     | zodpovednosť   |
|--------------------------|--|
| grafik papierovej skice  | - nakreslenie skice na papier  |
| pozorovateľ              | - posúdi kvalitu danej skice   |
| grafik počítačovej skice | - prekreslenie skice z náčrtku do počítača<br>- pri postavách oddeliť pohyblivé končatiny do samostatných balíkov aby sa tieto dali ďalej animovať |

### 2.4.3 Procesy splnenia úlohy

|    | Krok                          |
|----|-------------------------------|
| 1. | Náčrt                         |
| 2. | Posúdenie kvality skice       |
| 3. | Dokreslenie náčrtov           |
| 4. | Vytvorenie počítačovej skice  |
| 5. | Dokreslenie počítačovej skice |

#### 2.4.3.1 Náčrt

Vstup: opis scény zo scenára

Výstup: papierový náčrt scény

Zodpovedný: grafik papierovej skice

Grafik papierovej skice pripraví niekoľko náčrtov podľa opisu v scenári.

#### 2.4.3.2 Posúdenie kvality skice

Vstup: náčrty scény

Výstup: schválenie / pripomienkovanie náčrtu

Zodpovedný: pozorovateľ

Pozorovateľ skontroluje skice, prípadne vyberie najvhodnejší variant. Skontroluje či bol dodržaný globálny štýl hry. Následne schváli alebo pripomienkuje niektoré časti náčrtu.

#### 2.4.3.3 Dokreslenie náčrtu

Vstup: pripomienky k náčrtu skice

Výstup: opravený náčrt skice

Zodpovedný: grafik papierovej skice

Grafik papierovej skice prekreslí náčrt podľa pripomienok od pozorovateľa.

#### 2.4.3.4 Vytvorenie počítačovej skice

Vstup: papierový náčrt skice

Výstup: rdw a png súbor z prekreslenou skicou

Zodpovedný: grafik počítačovej skice

Grafik počítačovej skice prekreslí náčrt v programe RealDrow do počítačovej podoby. Pri kreslení postáv ukladá jednotlivé končatiny do zvlášť balíkov aby bola možná neskoršia animácia. Finálny produkt uloží vo formáte .rwd a .png.

2.4.3.5 Dokreslenie počítačovej skice

Vstup: pripomienky k náčrtu skice

Výstup: opravený náčrt skice

Zodpovedný: grafik počítačovej skice

Grafik počítačovej skice prekreslí náčrt podľa pripomienok od pozorovateľa.

## 2.5 Metodika na prácu s úlohami

Táto metodika popisuje formálnu prácu s úlohami v rámci tímového projektu. Zahŕňa ich vytváranie, zadeľovanie a kritéria splnenia.

### 2.5.1 Pojmy

- TFS – team foundation server, manažovací systém

### 2.5.2 Role a zodpovednosti

| Rola               | Zodpovednosť   |
|--------------------|--|
| Zadávatel'         | <ul style="list-style-type: none"><li>• určenie úlohy</li><li>• vysvetlenie úlohy</li></ul>  |
| Zapisovateľ/Tvorca | <ul style="list-style-type: none"><li>• vytvorenie úlohy v TFS</li><li>• doplnenie popisu</li><li>• priradenie riešiteľovi</li></ul>                                       |
| Riešiteľ           | <ul style="list-style-type: none"><li>• doplnenie časového odhadu</li><li>• zmena stavu pri riešení</li><li>• evidencia stráveného času</li><li>• splnenie úlohy</li></ul> |
| Kontrolujúci       | <ul style="list-style-type: none"><li>• zhodnotenie výstupu</li><li>• uzatvorenie úlohy</li></ul>  |
| Prezentátor        | <ul style="list-style-type: none"><li>• prezentácia výsledkov úlohy v tíme</li><li>• vysvetlenie výstupov</li><li>• zodpovedanie otázok ostatných členov tímu</li></ul>    |

### 2.5.3 Proces splnenia úlohy

|    | Krok          |
|----|---------------|
| 1. | Naplánovanie  |
| 2. | Vytvorenie    |
| 3. | Riešenie      |
| 4. | Kontrola      |
| 5. | Doplnenie     |
| 6. | Prezentovanie |

#### 2.5.3.1 Naplánovanie

Vstup : požiadavka

Výstup : evidencia úlohy v dokumente

Zodpovedný : zadávatel'

Zadávatel' identifikuje potrebu splnenia vybranej úlohy. Úlohu aj jej opodstatnenie opíše členom tímu. Zapisovateľ úlohu zaeviduje do zápisnice. V rámci tímu sa dohodne riešiteľ tejto úlohy.

#### 2.5.3.2 Vytvorenie

Vstup : dokument

Výstup : úloha v TFS nástroji

Zodpovedný : zapisovateľ/tvorca

Zapisovateľ prepíše úlohu do nástroja na evidenciu úloh a pridá k nej podrobný popis obsahujúci rozsah riešenej úlohy a kritéria na jej splnenie. Následne priradí vytvorenú úlohu dohodnutému členovi tímu.

#### 2.5.3.3 Riešenie

Vstup : priradená úloha v TFS

Výstup : reprezentácia splnenej úlohy

Zodpovedný : riešiteľ

Riešiteľ zmení stav úlohy na aktívna a snaží sa splniť kritéria na jej splnenie. Počas práce si eviduje strávený čas.

#### 2.5.3.4 Kontrola

Vstup : reprezentácia splnenej úlohy

Výstup : zhodnotenie úlohy

Zodpovedný : kontrolujúci

Riešiteľ požiada člena tímu o skontrolovanie úlohy. Kontrolujúci zhodnotí či úloha spĺňa dohodnuté kritéria a spíše si prípadné nedostatky. Ak sa nedostatky nenachádzajú tak zmení stav úlohy na splnená.

#### 2.5.3.5 Doplnenie

Vstup : reprezentácia úlohy a pripomienky

Výstup : opravená úloha

Zodpovedný : riešiteľ

Ak kontrolujúci nájde nedostatky, skontaktuje riešiteľa a ten následne opraví zistené nedostatky. Proces pokračuje predchádzajúcim krokom.

#### 2.5.3.6 Presentovanie

Vstup : prezentácia

Výstup : nové poznatky v rámci tímu

Zodpovedný : prezentátor

Na stretnutie si riešiteľ pripraví krátku prezentáciu. Oboznámi členov tímu so svojou úlohou a spôsobom jej vyriešenia. Odpovie na prípadné otázky a vysvetlí použité nástroje a techniky.



## 2.6 Metodika na tvorbu zápisníc

Na každom stretnutí sa píše zápisnice, ktoré sa následne formalizujú do jednotného formátu.

### 2.6.1 Role a zodpovednosti

| Rola           | Zodpovednosť                               |
|----------------|--|
| Zapisovateľ    | Zapisovanie na stretnutí<br>Konečná úprava |
| Vedúci         | Určenie zapisovateľa                       |
| Webový správca | Zverejnenie zápisnice na webe              |

### 2.6.2 Proces splnenia úlohy

|    |                            |
|----|----------------------------|
| 1. | Určenie zapisovateľa       |
| 2. | Zapisovanie                |
| 3. | Určenie úloh zo stretnutia |
| 4. | Formálna úprava            |
| 5. | Preklad                    |
| 6. | Zverejnenie                |

#### 2.6.2.1 Určenie zapisovateľa

Vstup: Stretnutie

Výstup: Meno zapisovateľa

Rola: Vedúci

Vedúci na začiatku stretnutia určí zapisovateľa, ktorý bude mať na starosti zápisnicu z aktuálneho stretnutia.

#### 2.6.2.2 Zapisovanie

Vstup: Meno zapisovateľa

Výstup: Neformálna zápisnica

Rola: Zapisovateľ

Počas celého stretnutia musí zapisovateľ zapisovať všetky dôležité informácie spomenuté na stretnutí. Má za úlohu z rozhovoru aj identifikovať úlohy, ktoré bude treba riešiť do ďalšieho stretnutia.

#### 2.6.2.3 Určenie úloh zo stretnutia

Vstup: Stretnutie

Výstup: Zoznam úloh

Rola: Vedúci, zapisovateľ

V druhej časti stretnutia treba identifikovať a pridelit' úlohy členom, ktoré vznikli na základe diskusie. Na toto slúži zápisnica, z ktorej zapisovateľ vyčíta úlohy a spolu so zapisovateľom určia ich potrebu riešenia-

#### 2.6.2.4 Formálna úprava

Vstup: Neformálna zápisnica

Výstup: Finálna verzia dokumentu v slovenčine

Rola: Zapisovateľ

Po stretnutí najneskôr do 24 hodín zapisovateľ upraví zápisnicu do finálnej podoby a zverejní ju na zdieľanom úložisku.

#### 2.6.2.5 Preklad

Vstup: Finálna verzia dokumentu v slovenčine

Výstup: Finálna verzia dokumentu v angličtine

Rola: Zapisovateľ

Následne treba preložiť zápisnicu aj do angličtiny, v prípade, že sa stretnutie určí ako menej dôležité pre medzinárodné potreby preklad nie je potrebný.

#### 2.6.2.6 Zverejnenie

Vstup: Finálne verzie dokumentov

Výstup: Dokumenty zverejnené na stránke

Rola: Webový správca

Ako náhle zapisovateľ zverejní dokumenty na OneDrive oznámi to webovému správcovi, ktorý dokumenty ihneď nahrá na internetovú stránku projektu.

## 2.7 Pravidlá tvorby skíc na animovanie

Skice určené na animáciu sa musia kresliť tak aby obsahovali balíky predstavujúce jednotlivé končatiny. Projekt musí obsahovať balík s zvlášť nakreslenou:

- ľavou rukou,
- pravou rukou
- ľavou nohou
- pravou nohou
- hlavou
- trupom

eventuálne časti skice ako palica, plášť a iné ktoré sa majú hýbať nezávisle na ostatných častiach teľa sa ukladajú tiež do osobitných balíkov.

Pri úpravách skíc sa vytvorí nový súbor s názvom názov x+1, kde x je číslo ktoré mal obrázok predtým. Prvý obrázok sa uloží ako názov 00.

Každá skica nejakého objektu alebo postavy sa uloží do nezávislého súboru ktorí bude obsahovať všetky verzie skíc, vyexportovaný bitmap obrázok skice v primeranej kvalite a všetky externé súbory ktoré sa v projekte použili(ako textúry a iné).

### Skice do animácií

Postup konečného výstupného formátu skice určenej na animáciu

1. Ak kreslíme postavu z boku tak musí existovať aj predná aj zadná noha.
  2. Predné a zadné končatiny nemôžu byť priamo cez seba – s kosťami sa potom zle hýbe.  
– Jednu ruku pootočiť smerom dopredu a pod. (nemusí byť realisticky)
  3. V real draw spojíme krivky tvoriace jeden celok postavy do spoločného package-u (celok je vždy časť ktorá sa hýbe vid' doc. skice do animácií) Ak je oblečenie z dvoch vrstiev medzi ktorými je ruka, treba ich exportovať samostatne (snažíme sa o čo najmenej vrstiev/celkov, ale musia existovať)
  4. Pre každý celok otvoríme nové plátno, vložíme celok, ctrl+q na orezanie platná na obrázok
  5. Exportneme použiteľný obrázok (Bitmap Export → Transparent alpha)
  6. Všetky obrázky dáme na Onedrive do jedného priečinku (postavaPoloha)
- 

iné zásady:

- kresby všetky robiť v rovnakej veľkosti, aby sa dodržala veľkosť animácie

## 2.8 Metodika na tvorbu zdrojových kódov pre skripty a programové komponenty vytvárané v programovacom jazyku C#

Uvedené konvencie pre pomenovávanie tried, ich atribútov a metód sa vzťahujú na zdrojové kódy v jazyku C#

### 2.8.1 Pojmy

PascalCasing - každé slovo v názve začína veľkým písmenom

camelCasing - každé slovo v názve okrem prvého začína veľkým písmenom

CONSTANT\_CASING - slová v názve sú oddelené podtržníkmi a

### 2.8.2 Role a zodpovednosti

| Rola        | Zodpovednosť                    |
|-------------|---------------------------------|
| Programátor | Vytvára a upravuje zdrojový kód |

### 2.8.3 Procesy

#### Tvorba zdrojového kódu

- Pri tvorbe zdrojového kódu je potrebné dodržiavať konvencie pomenovania, aby bol kód čitateľný a čo možno najviac samo vysvetľujúci.
- Názvy tried, metód, atribútov(objektov/premenných) a konštant by mali vysvetľovať vysvetľovať význam tak, aby okrem miesta ich definície/deklarácie nebol nikde inde potrebný komentár vysvetľujúci, čo daná metóda robí resp. akú informáciu daný atribút nesie.
- Odporúčané je použitie nástroja [ReSharper](#) pre kontrolu konvencií pomenovania, komentovania a formátovania pre uľahčenie písania, revidovania a refaktoringu zdrojového kódu

#### Pravidlá

- [Kapitalizácia identifikátorov](#)

Konštanty sú označené štýlom CONSTANT\_CASING.

Argumenty funkcií sú označené štýlom camelCasing.

Všetky ostatné identifikátory sú označené štýlom PascalCasing.

- [Konvencie písania kódu](#)

Konvencie písania kódu sú opísané na uvedených stránkach MSDN. Používame oficiálne C# konvencie pre zachovanie kompatibility s možnosťami nástroja ReSharper a pre minimalizáciu odchýlky v konvenciách od konvencií skriptovania definovanými komunitou Unity.

- [Konvencie komentovania kódu pre generovanie dokumentácie \(XML\)](#)

XML značky komentárov pre generovanie dokumentácie sú na príkladoch demonštrované na uvedených stránkach MSDN. Základné značky, ktoré je potrebné používať pre opis tried, atribútov a metód sú uvedené aj s príkladmi použitia.

```
///
```

```
/// Brief description of SomeClass, SomeMethod or SomeAttribute.
```

```
///</summary>
```

```
///<value>
```

```
/// Value tag is used to describe the property value, i.e. what does this object contain.
```

```
///</value>
```

```
///<returns>
```

```
/// Return value's description goes here.
```

```
///</returns>
```

```
///<param name="inputValue">
```

```
/// Parameter inputValue is described here.
```

```
///</param>
```

## 2.10 Metodika verziovania kódu

### 2.10.1 Pojmy

|                   |   |  |
|-------------------|---|--|
| Git               | - | verziovací systém  |
| Visual Studio TFS | - | Team Foundation Server od Microsoftu – slúži na evidenciu úloh a podporu verziovania |
| Commit            | - | Potvrdenie lokálnej zmeny v kóde   |
| Fetch             | - | Zobrazenie dostupných zmien na serveri   |
| Push              | - | Nahratie lokálnych zmien na server   |
| Pull              | - | Stiahnutie najnovšej verzie projektu zo servera                                      |

### 2.10.2 Role a zodpovednosti

| Rola        | Zodpovednosť  |
|-------------|---|
| Programátor | Vytvára a upravuje zdrojový kód<br>Udržiava si lokálnu verziu projektu aktuálnu a je zodpovedný za pravidelnú aktualizáciu a aktualizáciu lokálneho projektu s projektom na serveri |

### 2.10.3 Procesy

Kód sa píše v spoločnom projekte zdieľanom cez git a visual studio TFS.

Každá zmena kódu musí byť zdieľaná medzi ostatných členov tímu, aby mohli priebežne sledovať zmeny a vyhli sa tak konfliktom

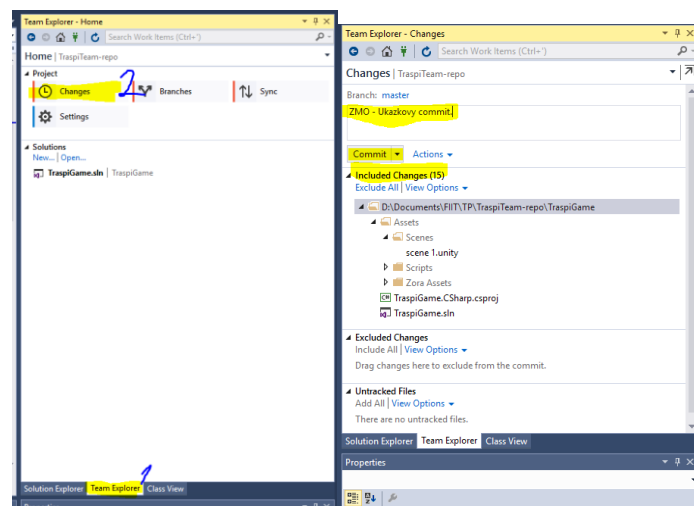
**Je dôležité, aby si každý pred začatím práce na projekte stiahol najnovšiu verziu projektu - PULL a svoje zmeny čo najčastejšie commitoval – po rôznych zmenách a pushoval vtedy keď si je istý svojou zmenou a má to otestované.**

Postup pushovania zmien:

1. Included changes – upraviť súbory, ktoré sa budú potvrdzovať podľa nižšie opísaných pravidiel
2. Commit – treba potvrdiť zmeny, ktoré sme urobili naposledy, správa pri commite musí na začiatku obsahovať skratku mena toho, kto zmenu potvrdzuje – (prvé písmeno mena a dva prvé písmená priezviska – ZMO, LMA, LMI, AMA, DHA, FSA)
3. Fetch – zobrazí všetky dostupne pull-y
4. Pull – ak sa nachádzajú nejaké nové zmeny, treba si ich stiahnuť pull príkazom
  - a. V prípade, že nastanú konflikty, tieto sa dajú vyriešiť len keď sú všetky lokálne zmeny commitnuté (bod 2)
  - b. Potom Visual Studio ponúkne možnosť resolve conflicts
  - c. Takto sa dá každý jeden súbor, ktorý je v konflikte prejsť a rozhodnúť sa, či zanechať naše zmeny, zmeny zo servera alebo ich pomocou merge toolu spojiť podľa svojho uváženia
  - d. Následne treba toto riešenie commitnúť a tým sa vyriešia konflikty
5. Sync / Push – sync urobí aj pull aj push, push uploaduje zmeny na server – všetky naše lokálne commity pošle na server a ostatní ich tak môžu vidieť

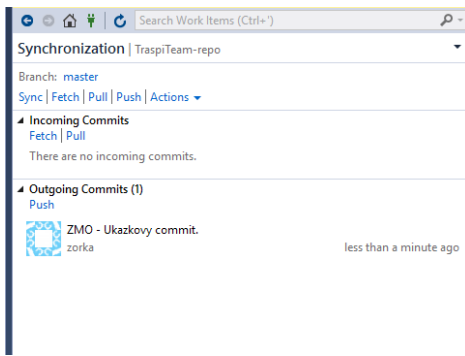
## Commit:

- Visual Studio -> Team Explorer -> Changes
  - Includes changes – Obsahuje súbory, v ktorých sa vykonali zmeny
    - .cs kódy
    - .meta súbory tých súborov, ktoré sme menili (ak sme v scéne vykonávali zmeny, musíme pridať súbor .unity.meta, ktorý obsahuje referencie na objekty v scéne)
    - ak sme pridali nový súbor (priečink, obrázok, asset) pridáme aj prislúchajúci .meta súbor
    - ak vieme, že sme nejaký súbor nemenili a máme v included changes jeho .meta súbor, excludneme ho – môže to byť súbor, ktorý si náš Unity lokálne a následne by mohol prepísať niekoho .meta file, ktorý s ním naozaj robil
    - Ak následne vznikne konflikt na takomto .meta súbore, preberieme ten, ktorý bol na serveri (ak vieme, že my zmeny lokálne nemáme)
    - .csproj, .sln – pridajú sa do included vtedy, keď sa mení štruktúra Visual studio projektu (premiestňovanie, pridanie skriptov), taktiež preberáme – konflikty si vie git dobre vyriešiť
  - Excluded changes
    - Vylúčime veci, čo vieme, že sme nemenili, hlavne .meta súbory, ktoré si samo Unity vytvorili a my sme ich nemenili
    - Treba to občas prejsť, môže sa stať, že .gitignore vylúči .meta súbory, ktoré chceme include
  - Untracked files
    - Väčšinou si navšívame, jedine keď nám vypíše, že untracked files prevent pull – vtedy treba nájsť problémové súbory, commitnúť a riešiť konflikty



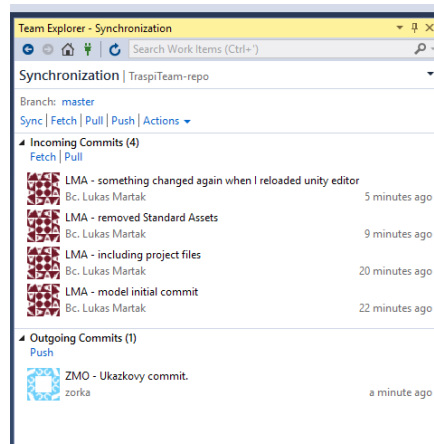
Obrázok 38: Zobrazenie si lokálnych zmien

- Sync
  - Ako je spomenuté v poradí krokov, po tom ako commit-neme naše zmeny musíme ich poslať na server – Sync príkaz
  - Predpokladáme, že sme pull-y najnovšie zmeny a vyriešili prípadne konflikt



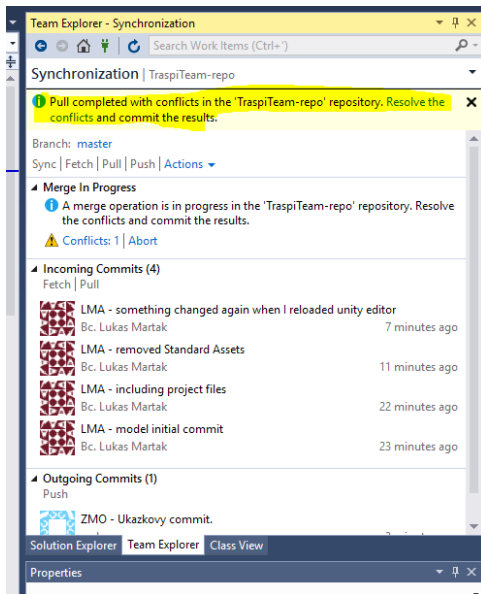
Fetch ->

Obrázok 39: Hneď po novom commit-e

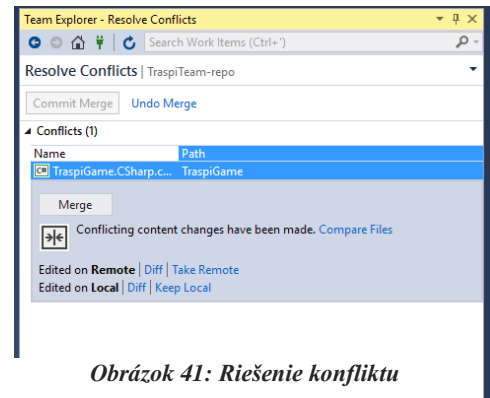


Obrázok 40: Po Fetch príkaze - pribudnú najnovšie pull-y zo servera

Pull ->



Resolve conflicts ->

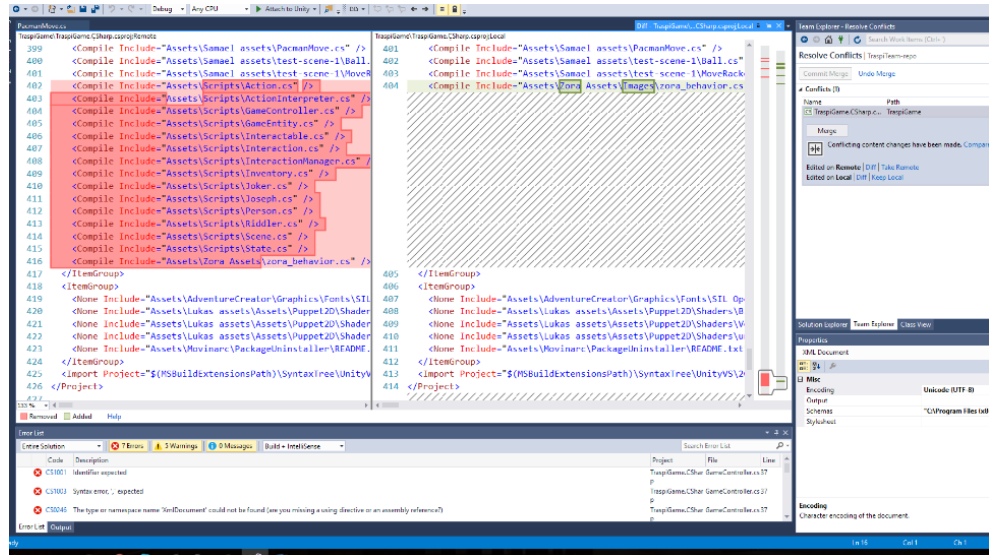


Obrázok 41: Riešenie konfliktu

Obrázok 42: Pull príkaz skončil s niekoľkými konfliktami



Compare files ->



Obrázok 43: Vidíme zmeny medzi našou verziou a konfliktnou zo serveru, na základe tohto sa rozhodneme či ponecháme našu verziiu alebo serverovú, alebo vytvoríme novú pomocou Merge nástroja

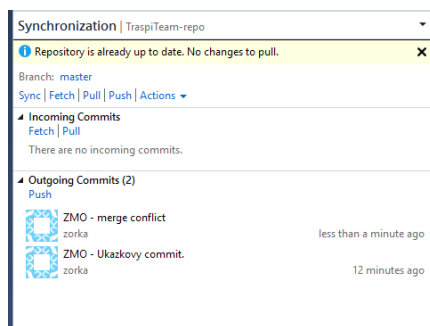
Take remote / Take Local



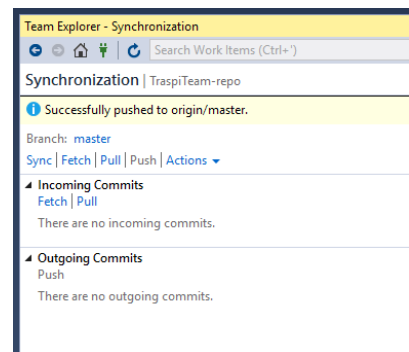
-> Commit merge ->

Obrázok 44: Potvrdíme našu zmenu (Commit merge)

Commit->



Push/Sync->



Obrázok 45: Vyriešený konflikt, najnovšia verzia projektu je up to date

Obrázok 46: Úspešne dokončený commit a push

## 2.11 Súborová štruktúra Unity projektu

### 2.11.1 Pojmy

- Unity projekt - projekt v nástroji Unity, v ktorom sa vyvíja produkt
- Jozef - hlavná postava v kre
- Hádankárka - jedna z vedľajších postáv

### 2.11.2 Role a zodpovednosti

| Rola        | Zodpovednosť  |
|-------------|---|
| Programátor | Musí udržiavať dohodnutú štruktúru projektu<br>Je zodpovedný za to, že súbory, ktoré do projektu vloží budú na správnom mieste a jednoducho vyhľadateľné iným programátorom |

### 2.11.3 Priečinky

- Adventure creator
  - o Súbory adventure creator assetu
- Animations
  - o animácie objektov, rozdelené do pod priečinkov buď podľa druhu pohybu alebo druhu postavy (podľa toho ako sa to bude robiť)
- Models
  - o obsahuje obrazové reprezentácie objektov v scéne
  - o Characters
    - Postavy ako Jozef, hádankárka
  - o Images
    - Všetky obrázkové súbory, ktoré sa používajú len ako sprity
    - Napríklad pozadie, neklikateľné objekty
  - o Items
    - Objekty, s ktorými sa dá interagovať
    - Napríklad dvere, alebo váza, ktorá sa da rozbiť
- Prefabs
  - o
- Scenes
  - o Súbory s príponou .unity, scény
  - o prvá scéna scena 1 – spoločná
- Scripts
  - o všetky scripty, ktoré napíšeme
  - o môže sa deliť na pod priečinky podľa logiky
- Sounds
  - o zvukové súbory použité v hre, môže sa následne deliť na pod priečinky
- Tests
  - o scény obsahujúce integration testy a iné podklady k testovaniu
  -

Na vlastné potreby má každý vytvorený priečinok so skúšobnou scénou kde si môže skúšať svoje veci – nemusia sa pushovať kompletne medzi ostatných

## 2.12 Metodika na tvorbu testov - UnityTestTools

Metodika bližšie popisuje využitie UnityTestTools na testovanie pridávaných častí funkcionality.

### 2.12.1 Nadväzujúce metodiky a dokumenty

Dokumentácia UnityTestTools –

<https://bitbucket.org/Unity-Technologies/unitytesttools/wiki/Home>

### 2.12.2 Role a zodpovednosti

| Rola        | Zodpovednosť   |
|-------------|--|
| Programátor | Vytvára a upravuje testovacie prípady                  |
| Tester      | Spúšťa testy, zapisuje výsledky a rieši objavené chyby |

### 2.12.3 Procesy

Tvorba testovacích UnityTestTools prípadov

- AssertionTesty – testovanie na báze jednoduchých podmienok. Vyberieme konkrétny objekt, konkrétnu hodnotu a aký má byť ich stav (porovnanie)
- Potrebne zdefinovať vhodný invariant a kedy sa má kontrolovať. Možné určiť na konkrétnu udalosť, časový interval či každý frame hry.
- IntegrationTesty – používať na vytvorených prefab objektoch. Testujeme ich správanie.
- Po vytvorení integration testu ešte treba zdefinovať kedy sa zavolá metóda plnenia testu a kedy metóda nesplnenia testu, aby sa testy dali vyhodnotiť
- Script je nazvaný call testing a pridáva sa na konkrétne prefab objekty
- V scéne sa vytvorí test do ktorého sa dajú objekty, ktoré testujeme a ktorým sa definujú testovacie skripty. Tieto objekty sú viditeľné len v rámci konkrétneho testu.
- Testy spúšťame po pridaní novej scény, po pridaní obsiahlejšej funkcionality a po zmene prefab objektu.

### Tvorba unit testov

Unity test tools v sebe integruje NUnit framework, ktorý slúži na tvorbu automatizovaných unit testov.

- Nové unit testy ukladáme do priečinka UnityTestTools\Examples\UnitTestExamples\Editor – v rámci tohto priečinka sú definované všetky potrebné referencie na používané knižnice
- Novú triedu obsahujúcu testovacie metódy označíme anotáciou [[TestFixture](#)]
- Každú metódu predstavujúcu unit test označíme anotáciou [[Test](#)]
- Testovaciu metódu kvôli čitateľnosti delíme na tri časti

- Prichystanie – zadefinujeme objekty, ktoré sa chystáme použiť a pripravíme im vhodný stav. Používame náhradné objekty určené na efektívnejšie testovanie.
- Akcia – zavolanie správania ktoré testujeme
- Vyhodnotenie – zhodnotenie stavu či správania, podľa tejto časti sa určí či bol test úspešný
- Pri písaní testovacích prípadov takisto dodržíjeme konvencie písania kódu popísané v metodike SourceCode

## 2.13 Metodika rizík

### 2.13.1 Identifikované riziká

#### **Zlé pochopenie úlohy**

Môže sa stať, že člen tímu nepochopí úlohu, ktorá sa mu prideli prípadne ju pochopí inak. Pri prideliovaní úloh je preto potrebná dôsledná komunikácia. Dôležitý je aj popis jednotlivých úloh v manažovacom nástroji. Dodržíjeme nasledovnú štruktúru:

- popis – bližšie rozobratie úlohy
- nadväznosť – ak je potrebné dokončenie iných úloh, treba spomenúť
- čas dokončenia – ak od tejto úlohy závisia iné, treba zadefinovať kedy bude úloha hotová
- výstup – čo sa očakáva od hotovej úlohy, aby sa vedelo vyhodnotiť či je úloha splnená
- riziká – ak má táto úloha špecifické riziká, treba ich uviesť a aj možnosť riešenia, užitočné pre riešiteľa

#### **Absencia člena tímu**

V prípade choroby či iných neočakávaných povinností je možné, že počas šprintu bude niektorý člen neschopný dokončiť svoje úlohy. Podľa charakteru úloh sa dá toto riziko vyriešiť rôzne:

- úloha nie je kľúčová – riziko ignorujeme, dá sa akceptovať nedokončenie úlohy, úloha sa splní pri najbližšej novej príležitosti
- na úlohu nadväzujú iné – ak člen tímu z vážnych príčin nestíha splniť task okamžite dá vedieť všetkým členom tímu. Po dohode si task zoberie iný člen tímu

#### **Zlá granularita úlohy (príliš veľká náročnosť)**

Pri nedostatočnej analýze prideliovaných úloh sa môže stať, že jedna z úloh má v sebe obsiahnuté väčšie množstvo práce ako sa predpokladalo. Tomuto riziku sa vyhneme konzultáciou v tíme no ak predsa nastane tak sa úloha rozdelí na dve menšie ak ju riešiteľ nestíha spraviť celú. Druhá časť sa prideli inému členovi, ktorý má menej práce.

#### **Zlé odhadnutie potrebného času**

Toto riziko čiastočne nadväzuje na predchádzajúce. Hlavný faktor tohto je však skutočnosť či máme referenčnú hodnotu s ktorou sa dá úloha porovnať. V začiatkoch nášho projektu sa oboznamujeme s väčším množstvom technológií a preto môžeme čakať častejšie zlé odhady. V budúcnosti sa im vieme vyhnúť detailným dokumentovaním terajších úloh.

#### **Nesplnenie úlohy členom tímu (nadväznosť iných úloh)**

Ak člen tímu z nejakého dôvodu nesplní svoju úlohu, na ktorú nadväzuje iná, hrozí neúspešné skončenie šprintu. Z tohto dôvodu je potrebné dohliadať priebežne na určené termíny a zisťovať v akom stave sú konkrétne úlohy. Tím sa tak vyhne väčším meškaniam.

### 2.13.2 Dodržiavané zásady na všeobecné predchádzanie rizikám

- Opis userstory v TFS musí obsahovať okrem zoznamu úloh aj identifikované špecifické riziká. Riešiteľovi to pomôže lepšie si riziká uvedomiť a pokiaľ je to možné vyhnúť sa im  
Členom tímu je odporúčané viesť si týždenný plánovač – mimo TFS nástroja na rozvrhnutie práce aby sa zabezpečilo plnenie úloh.

## 2.14 Metodika komunikácie

Tento dokument pojednáva o procesoch v rámci internej komunikácie tímu ale aj o komunikácii tímu so zadávateľom projektu a pedagogickým vedúcim. Metodika sa venuje práci s tromi hlavnými oficiálnymi komunikačnými nástrojmi, a to Slack a email v službe Gmail a Skype. Spomenutá je aj telefonická komunikácia.

### 2.14.1 Pojmy

| Pojem                      | Vysvetlenie  |
|----------------------------|--|
| Slack                      | nástroj pre tímovú komunikáciu                                 |
| Kanál na Slack-u (Channel) | tématický chat v službe Slack                                  |
| Gmail                      | služba firmy Google, poskytujúca emailovú schránku             |
| Skupinový email            | email, s ktorého chodia prijaté správy celému tímu             |
| Skype                      | nástroj pre uskutočnenie telefonátov prostredníctvom internetu |

### 2.14.2 Role a zodpovednosti

| Rola               | Zodpovednosť   |
|--------------------|--|
| Vysielateľ správy  | posiela správu   |
| Príjemca správy    | je adresátom správy, reaguje na správu   |
| Tím                | komunikuje interne aj externe  |
| Člen tímu          | komunikuje s ostatnými členmi tímu, vedúcim a v prípadoch, ktoré si to vyžadujú aj so zadávateľom projektu |
| Vedúci tímu        | komunikuje s tímom a zadávateľom projektu  |
| Zadávateľ projektu | komunikuje s vedúcim a členmi tímu keď je to potrebné, hlavne pri validácii požiadaviek                    |
| Externá osoba      | komunikuje s tímom, či jeho konkrétnym členom o záležitostiach súvisiacich s projektom                     |

### 2.14.3 Procesy

#### Komunikácia medzi členmi tímu

Vstup: potreba kontaktovať celý tím alebo jeho konkrétneho člena, správa

Výstup: doručená správa

Rola: vysielateľ, príjemca

Proces sa začne rozhodnutím vysielateľa o tom aká urgentná je správa, ktorú potrebuje príjemcovi poslať. Urgentnosť správy závisí na tom, ako rýchlo vysielateľ potrebuje na ňu dostať odpoveď. Pri najnižšej úrovni urgentnosti je ako prostriedok komunikácie zvolený tímový email. Pokiaľ správa nevyžaduje okamžitú reakciu, no jej urgentnosť je vyššia ako pri využití mailu, je ako komunikačný kanál zvolený Slack. Ak je okamžitá reakcia nevyhnutná vysielateľ môže využiť telefonický kontakt.

Pokiaľ sa vysielateľ rozhodne využiť email, a zároveň sa správa týka celého tímu je potrebné ju poslať na adresu skupinového mailu tímu. Správa bude takto doručená všetkým

členom tímu. Email je však častejšie používaný pri komunikácii s vedúcim tímu alebo externými osobami než medzi členmi tímu.

Pri využívaní Slack-u je ďalej potrebné rozhodnúť sa, či je správa striktné adresovaná jednému príjemcovi, alebo je vhodné, či dokonca žiadané aby ju dostali viacerí príjemcovia. Na základe tohto rozhodnutia odosielateľ vyšle správu privátne, použitím Direct Messages, alebo využije jeden z dostupných Channels na Slack-u, ku ktorým má prístup celý tím, alebo len jeho vybraní členovia. Pri odosielaní správy viacerým príjemcom je potrebné vybrať si za týmto účelom jeden z kanálov. Toto rozhodnutie vykoná vysielateľ na základe obsahu správy a názvov jednotlivých kanálov, ktoré opisujú predmet diskusie v danom kanály.

### **Založenie nového komunikačného kanála na Slack-u**

Vstup: potreba kontaktovať všetkých členov tímu a zároveň adresovať subjekt, ktorý sa tematicky nehodí do žiadneho z existujúcich komunikačných kanálov na Slack-u  
Výstup: nový Channel na Slack-u  
Rola: člen tímu

Je potrebné udržiavať komunikáciu na Slack-u prehľadnú a dobre kategorizovanú pre účely neskoršieho navrátenia sa k informáciám a správam, ktoré sa jeho prostredníctvom zaslali. Z tohto dôvodu je potrebné aby každá preberaná téma mala svoj vlastný komunikačný kanál. Člen tímu, ktorý potrebuje adresovať subjekt, ktorý nesúvisí s témou žiadneho kanála, vytvorí kanál nový.

Vykoná tak kliknutím na tlačidlo „+“, ktoré sa nachádza na vrchu zoznamu existujúcich komunikačných kanálov, hneď vedľa nadpisu „Channels“ v bočnom menu Slack-u. Následne v zobrazenom formulári nastaví či je nový kanál súkromný alebo verejný, vyplní jeho názov, pozve členov tímu, ktorých sa komunikácia na danom kanály týka a môže vložiť krátky popis nového kanála. Na záver potvrdí tvorbu nového kanála kliknutím na tlačidlo „Create Channel“.

### **Rutinná kontrola komunikačných kanálov**

Vstup:  
Výstup: reakcia na prijaté správy  
Rola: člen tímu

Je nutné aby každý člen tímu kontroloval komunikačné kanály v pravidelných intervaloch. Členovia tímu sú povinní sa minimálne raz za 24 hodín pripojiť ku kanálom Slack a tímový email. Člen tímu skontroluje, či sa ho daným kanálom nepokúšal, v spojitosti s projektom, kontaktovať druhý člen tímu, či iná, externá, osoba.

### **Dohodnutie stretnutia tímového projektu**

Vstup: potreba dohodnutia termínu stretnutia tímového projektu  
Výstup: termín stretnutia potvrdený všetkými zúčastnenými stranami  
Rola: člen tímu, tím, vedúci tímu

Stretnutia tímu, ktoré sa odohrávajú každý týždeň semestra, je nutné dopredu dohodnúť s vedúcim tímu. Ak nebol dátum a čas nasledujúceho stretnutia spomínaný na stretnutí predchádzajúcom člen tímu, ktorý má túto úlohu na starosti, tradične manažér komunikácie, sa najprv dohodne s ostatnými členmi tímu na vhodnom termíne. Následne odošle vedúcemu

tímu email, s možnými termínmi stretnutia. Tento email sa zároveň posiala aj na adresu skupinového tímu, aby mal k diskusií prístup celý tím. Vedúci tímu môže dátum stretnutia potvrdiť alebo zamietnuť a navrhnúť iný. Táto komunikácia pokračuje až kým vedúci tímu nepotvrdí konkrétny termín.

### **Skupinový telefonát za použitia služby Skype**

Vstup: potreba vykonania skupinového telefonátu prostredníctvom služby Skype  
Výstup: úspešný telefonát  
Rola: vysielateľ, člen tímu, tím

V prípade, že ostatné komunikačné kanále nepokryjú potreby vysielateľa správy je možné uskutočniť skupinový skype telefonát. Zväčša sa tu jedná o komunikáciu, ktorá je dlhšia, informačne náročnejšia a vyžaduje si účasť viacerých členov tímu, či vedúceho tímu. Proces sa začne tým, že vysielateľ si dohodne termín konania telefonátu použitím niektorého iného komunikačného kanála. V dohodnutom termíne potom vysielateľ najprv potvrdí, že sú všetci prítomní a následne začne hovor. Ostatní účastníci sa naň napoja, aby sa tak vyhlo problémom s duplicitnými telefonátmi.

### **Komunikácia s externými osobami**

Vstup: potreba komunikácie s externou osobou  
Výstup: úspešná komunikácia s externou osobou, o ktorej je informovaný celý tím  
Rola: tím, externá osoba

Pri oficiálnej komunikácií tímu, či jeho konkrétneho člena, s externou osobou, napríklad zadávateľom projektu, je kľúčové aby sa relevantné informácie dostali ku všetkým členom tímu. Pokiaľ je táto komunikácia uskutočňovaná cez email, je potrebné aby boli tieto maily následne zaslané aj na adresu tímového projektu. Ak prebieha komunikácia iným spôsobom, je povinnosťou člena tímu, ktorý s externou osobou komunikuje, zabezpečiť, že tieto informácie budú zdieľané s tímom prostredníctvom jedného z oficiálnych tímových komunikačných kanálov.



### 3 Export z TFS

---

Typ nášho tímového projektu je dosť špecifický a nie veľmi vhodný na metodiky používanú v rámci predmetu. Preto sme pri aplikovaní určenej metodiky narazili na isté problémy. Jeden z týchto problémov bol, že úlohy, ktoré sme v rámci projektu vykonávali, hlavne v začiatkovej fáze, neboli veľmi vhodné na evidenciu poskytujúcu používaným nástrojom. Myslíme tým hlavne fázu v ktorej sme vymýšľali príbeh a charakter hry, jednotlivé zápletky a prostredie. Tieto úlohy boli často určené pre celý tím a určenie času na splnenie bolo vopred nemožné. Preto v evidencii nie sú uvedené, prípadne sú priradené len jednému človeku z tímu.

Takisto neuvádzame niektoré z úloh, ktoré neboli explicitne určené, no jednotliví členovia ich vykonali vo vlastnom záujme, pre uľahčenie ďalšieho vývoja. Ako napríklad prieskumy rôznych spôsobov vývoja hier, zaúčanie sa do Unity a iné.

Úlohy nie sú rozdelené podľa týždňov, uvádzame však v rámci ktorého šprintu boli riešené.

Project: Patronus Server: tfs.fiit.stuba.sk\StudentsProjects Query: AllTasks List type: Flat

| ID   | Iteration Path | Work Item Type | Title  | Assigned To          | State    |
|------|----------------|----------------|--|----------------------|----------|
| 2572 | \              | User Story     | Rozbehať TFS server  | Bc. Lukas Miskovsky  | Resolved |
| 2573 | \              | Task           | pushnut repozitar  | Bc. Lukas Miskovsky  | Closed   |
| 2791 | \              | User Story     | Správanie hádankárky                                       |                      | New      |
| 2792 | \              | User Story     | Správanie vtipkárky  |                      | New      |
| 2842 | \              | User Story     | Vyvolanie minihry cez automat                              |                      | New      |
| 2843 | \              | User Story     | Minihra - FlappyJoe  |                      | New      |
| 2844 | \              | User Story     | Inventár   |                      | New      |
| 2845 | \              | User Story     | Obchod   |                      | New      |
| 2846 | \              | User Story     | Použitie predmetu v scéne                                  |                      | New      |
| 2847 | \              | User Story     | Implementácia audio služieb od SAV                         |                      | New      |
| 2848 | \              | User Story     | Logovanie stavov hry                                       |                      | New      |
| 3226 | \              | User Story     | Komunikácia s avатарom - Interaktívna                      |                      | New      |
| 3228 | \              | User Story     | Reprezentácia vtipkárky                                    |                      | New      |
| 3229 | \              | User Story     | Reprezentácia hádankárky                                   |                      | New      |
| 2778 | \Alohomora     | User Story     | Pohyb po scéne   | Bc. Ales Masiar      | Resolved |
| 2779 | \Alohomora     | Task           | Nakreslit prvu scena na papier                             | Bc. Daniela Hajdu    | Closed   |
| 2780 | \Alohomora     | Task           | Nakreslit postavy Jozefa na papier                         | Bc. Daniela Hajdu    | Closed   |
| 2781 | \Alohomora     | Task           | Prekreslit scenu do digitalnej podoby                      | Bc. Filip Sandor     | Closed   |
| 2782 | \Alohomora     | Task           | Vytvorit model postavy podla nacrtnu                       | Bc. Filip Sandor     | Closed   |
| 2783 | \Alohomora     | Task           | Vytvorit projekt a zabezpecit source control projektu      | Bc. Zora Moravčíková | Closed   |
| 2784 | \Alohomora     | Task           | Vytvorit prvu scenu z vytvorených modelov a obrazkov sceny | Bc. Ales Masiar      | Closed   |
| 2785 | \Alohomora     | Task           | Vytvorit pohyb hraca po scene                              | Bc. Ales Masiar      | Closed   |
| 2786 | \Alohomora     | Task           | Vytvorit animáciu pohybu postavy                           | Bc. Lukas Miskovsky  | Closed   |
| 2841 | \Alohomora     | Task           | Vymyslieť abstraktnú reprezentáciu scény                   | Bc. Lukas Martak     | Closed   |
| 2850 | \Alohomora     | Task           | Zaškoliť tím do RealDraw                                   | Bc. Filip Sandor     | Closed   |
| 2851 | \Alohomora     | Task           | Vytvorit model postavy podla nacrtnu                       | Bc. Daniela Hajdu    | Closed   |
| 2852 | \Alohomora     | Task           | Vyladiť funkčnosť source control                           | Bc. Lukas Martak     | Closed   |

|      |               |            |  |                      |          |
|------|---------------|------------|--|----------------------|----------|
| 2853 | \Alohomora    | Task       | Testovanie v Unity                                       | Bc. Ales Masiar      | Active   |
| 3017 | \Alohomora    | Task       | Vyladiť funkčnosť source control                         | Bc. Zora Moravčíková | Closed   |
| 3018 | \Alohomora    | Task       | Vymyslieť abstraktnu reprezentáciu scény                 | Bc. Zora Moravčíková | Closed   |
| 2790 | \Expelliarmus | User Story | Špecifikácia interakcie s objektom a vylepšenie grafiky  | Bc. Ales Masiar      | Resolved |
| 3019 | \Expelliarmus | Task       | Vymyslieť reprezentáciu objektov a interakcie            | Bc. Ales Masiar      | Closed   |
| 3020 | \Expelliarmus | Task       | Vymyslieť reprezentáciu objektov a interakcie            | Bc. Lukas Martak     | Closed   |
| 3021 | \Expelliarmus | Task       | Vymyslieť reprezentáciu objektov a interakcie            | Bc. Lukas Miskovsky  | Closed   |
| 3022 | \Expelliarmus | Task       | Vymyslieť reprezentáciu objektov a interakcie            | Bc. Zora Moravčíková | Closed   |
| 3023 | \Expelliarmus | Task       | Graficky zdokonaľiť hlavnú postavu                       | Bc. Filip Sandor     | Closed   |
| 3024 | \Expelliarmus | Task       | Zdokonaľiť skicu hlavnej postavy                         | Bc. Daniela Hajdu    | Closed   |
| 3025 | \Expelliarmus | Task       | Zdokonaľiť animáciu hlavnej postavy                      | Bc. Lukas Miskovsky  | Closed   |
| 3026 | \Expelliarmus | Task       | Manažment projektu                                       | Bc. Daniela Hajdu    | Closed   |
| 3027 | \Expelliarmus | Task       | Manažment projektu                                       | Bc. Filip Sandor     | Closed   |
| 3028 | \Expelliarmus | Task       | Manažment projektu                                       | Bc. Lukas Martak     | Closed   |
| 3029 | \Expelliarmus | Task       | Manažment projektu                                       | Bc. Lukas Miskovsky  | Closed   |
| 3030 | \Expelliarmus | Task       | Manažment projektu                                       | Bc. Ales Masiar      | Closed   |
| 3031 | \Expelliarmus | Task       | Manažment projektu                                       | Bc. Zora Moravčíková | Closed   |
| 3114 | \Expelliarmus | Task       | Dokončiť grafickú reprezentáciu Jozefa                   | Bc. Daniela Hajdu    | Closed   |
| 3115 | \Expelliarmus | Task       | Vytvoriť model scény a interakcií medzi hernými entitami | Bc. Lukas Martak     | Closed   |
| 3116 | \Expelliarmus | Task       | Vytvoriť vhodnú reprezentáciu pravidlového systému       | Bc. Ales Masiar      | Closed   |
| 3117 | \Expelliarmus | Task       | Grafická reprezentácia scén                              | Bc. Filip Sandor     | Closed   |
| 3142 | \Expelliarmus | Task       | Vytvoriť stavový diagram hry                             | Bc. Zora Moravčíková | Closed   |
| 3149 | \Lumos        | User Story | Implementácia interakcie s objektom                      | Bc. Lukas Martak     | Active   |
| 3150 | \Lumos        | Task       | Otestovať a spísať metodiku source control               | Bc. Zora Moravčíková | Closed   |
| 3151 | \Lumos        | Task       | Upraviť diagram stavového priestoru 1. kapitoly          | Bc. Zora Moravčíková | Closed   |
| 3152 | \Lumos        | Task       | Spísať metodiku pre notáciu stavového priestoru scén     | Bc. Zora Moravčíková | Closed   |

|      |                |            |  |                      |          |
|------|----------------|------------|--|----------------------|----------|
| 3153 | \Lumos         | Task       | Spísať postup pre tvorbu xml súborov           | Bc. Ales Masiar      | Closed   |
| 3154 | \Lumos         | Task       | Definovať štruktúru aplikácie                  | Bc. Zora Moravčíková | Closed   |
| 3155 | \Lumos         | Task       | Špecifikácia dialogovača                       | Bc. Ales Masiar      | Active   |
| 3156 | \Lumos         | Task       | Zadeinovať metodiku tvorby zdrojového kódu     | Bc. Lukas Martak     | Closed   |
| 3157 | \Lumos         | Task       | Implementácia herného modelu                   | Bc. Lukas Martak     | Closed   |
| 3158 | \Lumos         | Task       | Implementácia manažéra interakcií              | Bc. Ales Masiar      | Closed   |
| 3160 | \Lumos         | Task       | Implementácia interpretéra akcií               | Bc. Lukas Martak     | Closed   |
| 3161 | \Lumos         | Task       | Implementácia herného kontrolera               | Bc. Zora Moravčíková | Closed   |
| 3162 | \Lumos         | Task       | Grafika scén                                   | Bc. Filip Sandor     | Closed   |
| 3163 | \Lumos         | Task       | Animácia postavy                               | Bc. Lukas Miskovsky  | Closed   |
| 3164 | \Lumos         | Task       | Grafika hlavnej postavy                        | Bc. Daniela Hajdu    | Closed   |
| 3165 | \Lumos         | Task       | Spísať postup pre tvorbu grafiky postáv        | Bc. Filip Sandor     | Closed   |
| 3167 | \Lumos         | Task       | Špecifikácia presunu medzi scénami             | Bc. Filip Sandor     | Closed   |
| 3183 | \Lumos         | Task       | Implementácia interpretera akcií               | Bc. Lukas Miskovsky  | Closed   |
| 2574 | \Oculus Reparó | Task       | Dokument na gameplay brainstorming             | Bc. Filip Sandor     | Closed   |
| 2575 | \Oculus Reparó | Task       | Zabezpečiť tímovú komunikáciu cez Slack        | Bc. Ales Masiar      | Closed   |
| 2576 | \Oculus Reparó | Task       | Grafika webu                                   | Bc. Zora Moravčíková | Closed   |
| 2589 | \Oculus Reparó | Task       | Zápisnice zo stretnutí                         | Bc. Daniela Hajdu    | Closed   |
| 2591 | \Oculus Reparó | Task       | Materiály a tutoriály k vývoju pomocou Unity3D | Bc. Lukas Martak     | Closed   |
| 2598 | \Oculus Reparó | Task       | Navrhnuť model architektúry                    | Bc. Lukas Martak     | Closed   |
| 2629 | \Oculus Reparó | User Story | Tvorba príbehu                                 | Bc. Filip Sandor     | Resolved |
| 2630 | \Oculus Reparó | Task       | Vymyslieť záhadu                               | Bc. Ales Masiar      | Closed   |
| 2631 | \Oculus Reparó | Task       | Napísať scénař                                 | Bc. Filip Sandor     | Closed   |
| 2632 | \Oculus Reparó | User Story | Návrh produkčného systému                      | Bc. Lukas Martak     | Resolved |
| 2633 | \Oculus Reparó | Task       | Analýza produkčných systémov a možností        | Bc. Daniela Hajdu    | Closed   |
| 2758 | \Oculus Reparó | User Story | Úvodný manažment                               | Bc. Filip Sandor     | Resolved |

|      |                |            |   |                      |          |
|------|----------------|------------|---|----------------------|----------|
| 2759 | \Oculus Reparó | Task       | Vymyslieť logo tímu                         | Bc. Zora Moravčíková | Closed   |
| 2760 | \Oculus Reparó | Task       | Vymyslieť meno tímu                         | Bc. Filip Sandor     | Closed   |
| 2761 | \Oculus Reparó | User Story | Správa webu                                 | Bc. Ales Masiar      | Resolved |
| 2762 | \Oculus Reparó | Task       | Vytvoriť základnú štruktúru webu            | Bc. Ales Masiar      | Closed   |
| 2763 | \Oculus Reparó | Task       | Navrhnuť dizajn                             | Bc. Zora Moravčíková | Closed   |
| 2764 | \Oculus Reparó | User Story | Analýza herných mechanizmov                 | Bc. Filip Sandor     | Resolved |
| 2765 | \Oculus Reparó | User Story | Navrhnutie architektúry                     | Bc. Lukas Martak     | Resolved |
| 2766 | \Oculus Reparó | Task       | Analýza Stanely Parable                     | Bc. Filip Sandor     | Closed   |
| 2767 | \Oculus Reparó | Task       | Analýza Unity Web Playera                   | Bc. Zora Moravčíková | Closed   |
| 2768 | \Oculus Reparó | Task       | Nasadiť web na server                       | Bc. Ales Masiar      | Closed   |
| 2770 | \Oculus Reparó | Task       | Skúmať motivačné faktory v hrách            | Bc. Ales Masiar      | Closed   |
| 2771 | \Oculus Reparó | Task       | Skúmať motivačné faktory v hrách            | Bc. Lukas Miskovsky  | Closed   |
| 2772 | \Oculus Reparó | Task       | Vymyslieť mená postáv                       | Bc. Filip Sandor     | Closed   |
| 2773 | \Oculus Reparó | Task       | Vymyslieť mini hry                          | Bc. Lukas Miskovsky  | Closed   |
| 2774 | \Oculus Reparó | Task       | Spísať ohraničenia a otvorené otázky        | Bc. Daniela Hajdu    | Closed   |
| 2787 | \Rictusempra   | User Story | Inicializácia komunikácie s Avatarmi        | Bc. Zora Moravčíková | Active   |
| 3159 | \Rictusempra   | Task       | Definovať metodiku na testovanie a testovať | Bc. Lukas Miskovsky  | Closed   |
| 3166 | \Rictusempra   | Task       | Grafika hádankárky                          | Bc. Daniela Hajdu    | Active   |
| 3222 | \Rictusempra   | Task       | Implementácia dialógového okna              | Bc. Filip Sandor     | Closed   |
| 3225 | \Rictusempra   | User Story | Komunikácia s avатарom - Predpísaná         | Bc. Zora Moravčíková | Active   |
| 3227 | \Rictusempra   | User Story | Grafická reprezentácia avatarov             | Bc. Zora Moravčíková | Active   |
| 3230 | \Rictusempra   | Task       | Grafický návrh avatara                      | Bc. Daniela Hajdu    | Active   |
| 3231 | \Rictusempra   | Task       | Prekreslenie skice do RealDraw              | Bc. Daniela Hajdu    | New      |
| 3232 | \Rictusempra   | Task       | Vytvorenie skriptu na nahrávanie zvuku      | Bc. Lukas Miskovsky  | New      |
| 3233 | \Rictusempra   | Task       | Analýza možností REST komunikácie v unity   | Bc. Zora Moravčíková | New      |
| 3234 | \Rictusempra   | Task       | Implementácia REST komunikácie so SAV       | Bc. Lukas Martak     | New      |

|      |              |      |  |                      |        |
|------|--------------|------|--|----------------------|--------|
| 3235 | \Rictusempra | Task | Implementácia znemožnenia iných akcií pri prebiehajúcej interakcii | Bc. Lukas Martak     | New    |
| 3236 | \Rictusempra | Task | Vytvorenie dialógovača   | Bc. Ales Masiar      | New    |
| 3237 | \Rictusempra | Task | Vytvorenie skriptu na zadávanie textu                              | Bc. Filip Sandor     | New    |
| 3238 | \Rictusempra | Task | Rozšírenie interpretéra akcií                                      | Bc. Lukas Martak     | New    |
| 3239 | \Rictusempra | Task | Vylepšenie skriptu na zobrazovanie textu                           | Bc. Filip Sandor     | Closed |
| 3240 | \Rictusempra | Task | Manažment projektu   | Bc. Lukas Martak     | Active |
| 3241 | \Rictusempra | Task | Manažment projektu   | Bc. Lukas Miskovsky  | Active |
| 3242 | \Rictusempra | Task | Manažment projektu   | Bc. Filip Sandor     | Active |
| 3244 | \Rictusempra | Task | Manažment projektu   | Bc. Zora Moravčíková | Active |
| 3245 | \Rictusempra | Task | Manažment projektu   | Bc. Daniela Hajdu    | Active |

Obrázok 47: Zoznam úloh ZS

| ID                   | Iteration Path       | Work Item Type | Title  | Assigned To        | State    |
|----------------------|----------------------|----------------|--|--------------------|----------|
| <a href="#">2787</a> | Patronus\Rictusempra | User Story     | Inicializácia komunikácie s Avatarmi                               | Bc. Zora Moravčík  | Resolved |
| <a href="#">2847</a> | Patronus\Rictusempra | User Story     | Implementácia audio služieb od SAV                                 |                    | Resolved |
| <a href="#">3159</a> | Patronus\Rictusempra | Task           | Definovať metodiku na testovanie a testovať                        | Bc. Lukas Miskovsi | Closed   |
| <a href="#">3166</a> | Patronus\Rictusempra | Task           | Grafika hádankárky   | Bc. Daniela Hajdu  | Closed   |
| <a href="#">3168</a> | Patronus\Rictusempra | Task           | Implementácia dialógového okna                                     | Bc. Daniela Hajdu  | Removed  |
| <a href="#">3222</a> | Patronus\Rictusempra | Task           | Implementácia dialógového okna                                     | Bc. Filip Sandor   | Closed   |
| <a href="#">3225</a> | Patronus\Rictusempra | User Story     | Komunikácia s objektom v hre                                       | Bc. Ales Masiar    | Resolved |
| <a href="#">3227</a> | Patronus\Rictusempra | User Story     | Grafická reprezentácia avatarov                                    | Bc. Daniela Hajdu  | Resolved |
| <a href="#">3230</a> | Patronus\Rictusempra | Task           | Grafický návrh avatara   | Bc. Daniela Hajdu  | Closed   |
| <a href="#">3231</a> | Patronus\Rictusempra | Task           | Prekreslenie skice do RealDraw                                     | Bc. Daniela Hajdu  | Closed   |
| <a href="#">3232</a> | Patronus\Rictusempra | Task           | Vytvorenie skriptu na nahrávanie zvuku                             | Bc. Lukas Miskovsi | Closed   |
| <a href="#">3233</a> | Patronus\Rictusempra | Task           | Analýza a implementácia komunikácie s hlasovým                     | Bc. Zora Moravčík  | Closed   |
| <a href="#">3235</a> | Patronus\Rictusempra | Task           | Implementácia znemožnenia iných akcií pri prebiehajúcej interakcii | Bc. Lukas Martak   | Closed   |
| <a href="#">3236</a> | Patronus\Rictusempra | Task           | Implementácia dialógovača  | Bc. Ales Masiar    | Closed   |
| <a href="#">3237</a> | Patronus\Rictusempra | Task           | Vytvorenie skriptu na zadávanie textu                              | Bc. Filip Sandor   | Closed   |
| <a href="#">3238</a> | Patronus\Rictusempra | Task           | Rozšírenie interpretéra akcií                                      | Bc. Lukas Martak   | Closed   |
| <a href="#">3239</a> | Patronus\Rictusempra | Task           | Vylepšenie skriptu na zobrazovanie textu                           | Bc. Filip Sandor   | Closed   |
| <a href="#">3240</a> | Patronus\Rictusempra | Task           | Manažment projektu   | Bc. Lukas Martak   | Closed   |
| <a href="#">3241</a> | Patronus\Rictusempra | Task           | Manažment projektu   | Bc. Lukas Miskovsi | Closed   |
| <a href="#">3242</a> | Patronus\Rictusempra | Task           | Manažment projektu   | Bc. Filip Sandor   | Closed   |
| <a href="#">3244</a> | Patronus\Rictusempra | Task           | Manažment projektu   | Bc. Zora Moravčík  | Closed   |
| <a href="#">3245</a> | Patronus\Rictusempra | Task           | Manažment projektu   | Bc. Daniela Hajdu  | Active   |
| <a href="#">3366</a> | Patronus\Rictusempra | Task           | Vytvorenie XML súborov so základnými dialógmi                      | Bc. Ales Masiar    | Closed   |
| <a href="#">3367</a> | Patronus\Rictusempra | Task           | Implementácia prevodníka zvuku do WAV                              | Bc. Ales Masiar    | Closed   |

| ID   | Iteration Path    | Work Item Type | Title                                      | Assigned To          | State    |
|------|-------------------|----------------|--|----------------------|----------|
| 2781 | Patronus/Bombarda | User Story     | Správanie hľadankárky                      |                      | Closed   |
| 2844 | Patronus/Bombarda | User Story     | Inventar Obchod                            | Bc. Filip Sander     | Resolved |
| 2845 | Patronus/Bombarda | User Story     | Obchod                                     | Bc. Lukas Miskovsky  | Resolved |
| 2848 | Patronus/Bombarda | User Story     | Lopovanie stavov hry                       | Bc. Lukas Miskovsky  | Resolved |
| 3128 | Patronus/Bombarda | User Story     | Reprezentácia vtipkárky                    | Bc. Daniela Hajdu    | Resolved |
| 3129 | Patronus/Bombarda | User Story     | Reprezentácia hľadankárky                  | Bc. Daniela Hajdu    | Closed   |
| 3248 | Patronus/Bombarda | Task           | Výbaň malovej škodlivosti                  | Bc. Zora Moravčíková | Removed  |
| 3250 | Patronus/Bombarda | Task           | Připravil scény v Unity                    | Bc. Filip Sander     | Closed   |
| 3251 | Patronus/Bombarda | Task           | Prepisovanie medií scenári                 | Bc. Zora Moravčíková | Closed   |
| 3252 | Patronus/Bombarda | Task           | Prenášanie informácií medzi scénami        | Bc. Zora Moravčíková | Closed   |
| 3253 | Patronus/Bombarda | Task           | Revidácia a opravenie funkčnosti kódu      | Bc. Lukas Miskovsky  | Closed   |
| 3254 | Patronus/Bombarda | Task           | Dojnosť XML - prídáv Alceoi                |                      | Closed   |
| 3255 | Patronus/Bombarda | Task           | Prebratanie vtipkárky do digitálnej podoby | Bc. Daniela Hajdu    | New      |
| 3258 | Patronus/Bombarda | Bug            | Prí zotavení / omdavení inputu sa vypíše   | Bc. Filip Sander     | Resolved |
| 3276 | Patronus/Bombarda | Task           | Inventar                                   | Bc. Filip Sander     | Removed  |
| 3278 | Patronus/Bombarda | Task           | Rozšírenie dialógovča pre Avatars          | Bc. Alex Maslar      | Closed   |
| 3280 | Patronus/Bombarda | Task           | vytvoril prototyp obchodu                  | Bc. Lukas Miskovsky  | Closed   |
| 3281 | Patronus/Bombarda | Task           | Gramatická kontrola abstraktných TP-cup    | Bc. Daniela Hajdu    | Closed   |
| 3282 | Patronus/Bombarda | Task           | Obsahová kontrola abstraktných TP-cup      | Bc. Alex Maslar      | Closed   |
| 3283 | Patronus/Bombarda | Task           | Vytvoril grafiku reprezentáciu inventára   | Bc. Filip Sander     | Closed   |
| 3284 | Patronus/Bombarda | Task           | Vytvorenie kľúčových funkcií pre inventár  | Bc. Filip Sander     | Closed   |

| ID   | Iteration Path     | Work Item Type | Title  | Assigned To          | State    |
|------|--------------------|----------------|--|----------------------|----------|
| 2792 | Patronus/Cruciatas | User Story     | Správanie vtipkárky                          |                      | Closed   |
| 2843 | Patronus/Cruciatas | User Story     | Minhra - Flappyvce                           | Bc. Lukas Martak     | Resolved |
| 2846 | Patronus/Cruciatas | User Story     | Použitie predmetu v záne                     | Bc. Alex Maslar      | Resolved |
| 3277 | Patronus/Cruciatas | Task           | Minhra - model                               | Bc. Lukas Martak     | Closed   |
| 3279 | Patronus/Cruciatas | Task           | grafika                                      | Bc. Daniela Hajdu    | New      |
| 3330 | Patronus/Cruciatas | Task           | Pridat vplyv do scemy                        | Bc. Zora Moravčíková | Closed   |
| 3331 | Patronus/Cruciatas | Task           | Správanie vtipkárky                          | Bc. Zora Moravčíková | Closed   |
| 3332 | Patronus/Cruciatas | Task           | Aktualizácia SAV služieb                     | Bc. Zora Moravčíková | Closed   |
| 3334 | Patronus/Cruciatas | Task           | Minhra - obsah                               | Bc. Lukas Martak     | Closed   |
| 3335 | Patronus/Cruciatas | Task           | Prepracoval grafiku hry                      | Bc. Filip Sander     | Closed   |
| 3336 | Patronus/Cruciatas | Task           | Implementácia stavov a akcií do inventára    | Bc. Alex Maslar      | Active   |
| 3337 | Patronus/Cruciatas | Task           | Vytvorenie animácie pre vtipkárku            | Bc. Daniela Hajdu    | New      |
| 3338 | Patronus/Cruciatas | Task           | vytvorenie animácie pre avatara              | Bc. Daniela Hajdu    | New      |
| 3339 | Patronus/Cruciatas | Task           | Nastavenia                                   | Bc. Lukas Miskovsky  | Closed   |
| 3370 | Patronus/Cruciatas | Task           | Pridanie stavov a prechodov do XML           | Bc. Alex Maslar      | Closed   |
| 3371 | Patronus/Cruciatas | Task           | Vytvorenie inventára cez baný                | Bc. Filip Sander     | Closed   |
| 3372 | Patronus/Cruciatas | Task           | Napísal SAV oHľadom prehrávaním              | Bc. Zora Moravčíková | Closed   |
| 3373 | Patronus/Cruciatas | Task           | Návrh základného algoritmu chatbota          | Bc. Zora Moravčíková | Closed   |
| 3374 | Patronus/Cruciatas | Task           | Implementovanie virového dialógu             | Bc. Zora Moravčíková | Closed   |
| 3375 | Patronus/Cruciatas | Task           | Minhra - grafika                             | Bc. Daniela Hajdu    | Closed   |
| 3285 | Patronus/Bombarda  | Task           | Vytvorenie kľúčových funkcií pre inventár    | Bc. Filip Sander     | Closed   |
| 3333 | Patronus/Bombarda  | User Story     | Prechod medzi scénami                        |                      | Resolved |
| 3289 | Patronus/Bombarda  | Task           | Pridal dialógy v Avatarsi do dialógových XML | Bc. Alex Maslar      | Closed   |

| ID   | Iteration Path         | Work Item Type | Title                                       | Assigned To          | State    |
|------|------------------------|----------------|---|----------------------|----------|
| 3322 | Patronus/Avada Kedavra | User Story     | Vytvorenie minihry cez automaty             | Bc. Alex Maslar      | Resolved |
| 3375 | Patronus/Avada Kedavra | Task           | Zrušenie dialógu                            | Bc. Zora Moravčíková | Closed   |
| 3376 | Patronus/Avada Kedavra | Task           | Odobranie rady                              | Bc. Zora Moravčíková | Closed   |
| 3377 | Patronus/Avada Kedavra | Task           | Rozšírenie druhu odpovede                   | Bc. Zora Moravčíková | Closed   |
| 3378 | Patronus/Avada Kedavra | User Story     | Komunikácia s Avatarsi - plná funkčnosť     | Bc. Zora Moravčíková | Resolved |
| 3380 | Patronus/Avada Kedavra | Task           | Zobrazenie stavu peňazí                     | Bc. Lukas Miskovsky  | Closed   |
| 3445 | Patronus/Avada Kedavra | Task           | Príchod avatara                             | Bc. Lukas Miskovsky  | Closed   |
| 3446 | Patronus/Avada Kedavra | Task           | Ida animácia - hľadankarka, vtipkárka       | Bc. Lukas Miskovsky  | Closed   |
| 3447 | Patronus/Avada Kedavra | Bug            | Nemožné otvoriť inventar                    | Bc. Lukas Miskovsky  | Closed   |
| 3448 | Patronus/Avada Kedavra | Bug            | Posunutie obsahu inventára                  | Bc. Filip Sander     | Active   |
| 3449 | Patronus/Avada Kedavra | Task           | Návrh a implementácia XML pre komerciu      | Bc. Zora Moravčíková | Closed   |
| 3450 | Patronus/Avada Kedavra | Task           | Zmena hlasy hovoriacich postavy v komentári | Bc. Zora Moravčíková | Closed   |
| 3451 | Patronus/Avada Kedavra | Task           | Spojenie prechod medzi hrami a minihrou     | Bc. Lukas Martak     | Closed   |
| 3452 | Patronus/Avada Kedavra | User Story     | Správanie hľadankárky                       | Bc. Alex Maslar      | Closed   |
| 3453 | Patronus/Avada Kedavra | Task           | Rozšírenie dialógového manabera             | Bc. Alex Maslar      | Closed   |
| 3454 | Patronus/Avada Kedavra | Task           | Zadefinovanie prvej hľadanky v XML          | Bc. Alex Maslar      | Closed   |
| 3455 | Patronus/Avada Kedavra | User Story     | Správanie dveri                             | Bc. Alex Maslar      | Closed   |
| 3456 | Patronus/Avada Kedavra | Task           | Vloženie konverzácie s divčami do XML       | Bc. Alex Maslar      | Closed   |
| 3457 | Patronus/Avada Kedavra | Task           | Vloženie dialógu pre scény 1 do XML         | Bc. Alex Maslar      | Closed   |
| 3458 | Patronus/Avada Kedavra | User Story     | Zobrazenie tapetného herla                  | Bc. Alex Maslar      | Resolved |
| 3459 | Patronus/Avada Kedavra | Task           | Rozšírenie interpretéra akcií               | Bc. Alex Maslar      | Closed   |



## 4 Prílohy

---

### A. Integrácia TFS a Unity

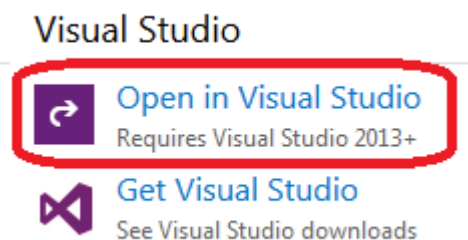
#### A-1 Install

You can download all installers [HERE](#).

1. Download & Install Unity 5 Personal Edition (preferable including Standard Assets)
2. Download & Install MS Visual Studio Community 2015 (or at least 2013 for TFS compatibility)
  - a. Free Student Licence is also available on [DreamSpark](#)
  - b. Custom Installation -> uncheck all Features (installable later) except **Common Tools**:
    - Git for Windows
    - GitHub Extensions for Visual Studio
    - Visual Studio Extensibility Tools
3. Download & Install FREE VS Tools for Unity

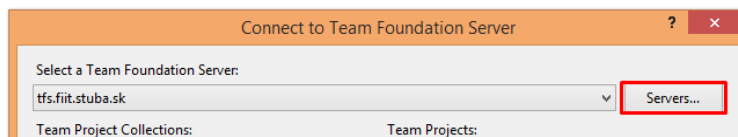
#### A-2 Configure

Open TFS project from [HERE](#)



Pridať nový server:

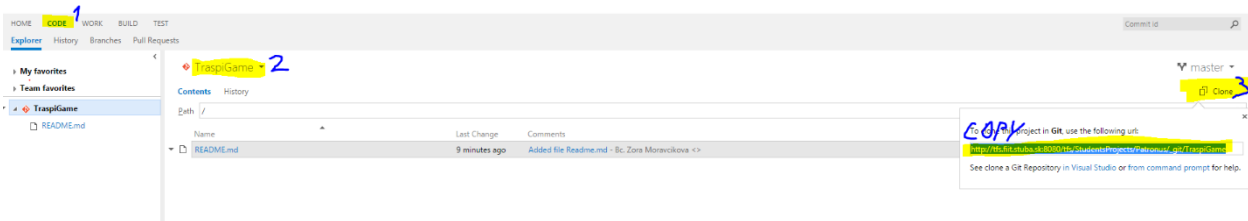
1. Manage connections
2. Connect to team project
- 3.



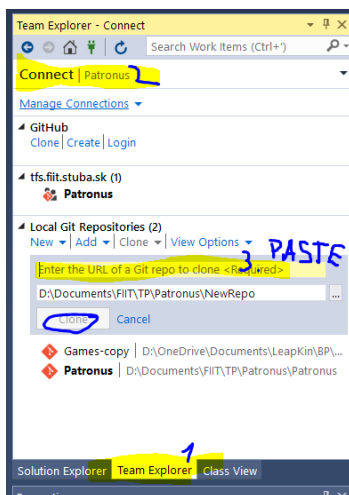
- 4.
5. Add...
6. Do name or URL napíšte: <http://tfs.fiit.stuba.sk:8080/tfs>

## A-3 Pripojenie sa na aktuálny projekt

1. Vytvor priečinok na lokály, kde budeš klonovať remote repository a budeš tam mať celý unity projekt (príklad D:\Documents\FIIT\TP\Patronus\TraspiTeam-repository)
2. Získaj clone adresu repozitáru TraspiGame



3. Kopíruj tú clone adresu do visuálka.
  - a. Otvor visuálka nie cez unity
  - b. Si pripojený na tfs server ako bolo už vyššie..
  - c. Team → manage connections, vpravo máš team explorer → connect
  - d. local git repositories → clone a tam vlož url z bodu 3 do priečinka, ktorý si vytvoril v prvom kroku
  - e. clone



4. Zapni Unity a manuálne otvor priečinok TraspiGame vnútri tvojho vytvoreného repo s klonovaným projektom – čakaj 😊
  - a. Projekt obsahoval len assets folder a project settings – ostatne si unity vytvorí samo
5. Otvor potom visuálka cez unity a môžeš pracovať
6. V assetoch má každý vlastný folder, kde si môže ukladať nejaké temp súbory, avšak ostatne ako obrázky, modely kódy, by sme mali riešiť spoločne
7. Každý má svoju skratku, prvé písmeno mena + dve písmena priezviska (zmo, fsa, lmi, lma...)
  - a. Toto používať na commit message (pr. ZMO – Pridanie scény do projektu)
  - b. Commitujeme a pushujeme len veci z assets

## B. Scénar pre prvú scénu

INTRO – je čarodejník, mágovia a muklovia, škola, robí sa na škole rušička – všetci o tom vedia

### KAPITOLA 1:

Sme na intráku – „na vrátnici“ – s Hugom a s deťmi, ktoré idú na intrák (dokopy Hugo, Jozef, jeho 2 kamaráti a nejaká skupinka detí (mrak alebo pár animovaných random detí – 3 extra deti))

Hugo: „Dobre, teraz vás všetkých odvediem do vašich izieb. Držte sa pokope, prvýkrát tam musíme ísť spolu, aby som vám ukázal ako to tu funguje.“

Jozef ide posledný – preletí okolo duch (popred neho) – resp. nie duch ale avatar (len ako záblesk) zľakne sa – spadne a pokecá s avatarom:

**Jozef:** Nadáva

**Avatar 1:** „Dávaj si pozor na jazyk!“

**Avatar 2:** „Inak ahoj Jozef“

Hráč by mal povedať „Ahoj“

**Avatar 2:** „Prepáč za to prudké uvítanie, potrebovali sme upútať tvoju pozornosť“

**Avatar 1:** „Vieš kto sme?“

**Hráč:** „Nie.“ / „Áno“

**Avatar 1:** „Dobre“

**Avatar 2:** „Dôležité je, že keby si čokoľvek potreboval, môžeme ti poradiť.“

**Avatar 1:** „Keď sme cez teba teraz preleteli vytvorili sme si s tebou spojenie.“

**Avatar 2:** „V podstate to funguje tak, že ak sa nám bude zdať, že potrebuješ poradiť, začuješ naše hlasy.“

**Avatar 1:** „Ak nás chceš zavolať ty, stačí aby si na nás pomyslel a my sa ti ozveme“

**Avatar 2:** „Všetkému rozumieš?“

**Hráč:** „Nie.“ / „Áno“

| Nie   | Áno                       |
|---|---------------------------|
| <b>Avatar 2:</b> „Nevadí, pochopíš neskôr.“ | <b>Avatar 2:</b> „Super.“ |

**Avatar 1:** „Dobre, už sme ťa zdržali viac než dost. Teraz sa musíš ísť ubytovať.“

\*odletia

**Jozef:** „No super ... a teraz sú všetci preč.“

skupinka odišla na 2. scénu on za nimi pobeží ale v 2. scéne už nie sú a je tam len hádankárka (Zuzka) a vtipkárka (Anča) (a automat na minihry (ktorý funguje len zato, že ten jeho otec to vyvíjal a jediné čo sa mu podarilo bolo hento sprovozniť) – toto bude neskôr)

**Avatar 2:** „Podľa mňa by si sa mal ísť porozprávať s tou slečnou vzadu pri dverách.“

**Avatar 1:** „Podľa mňa by si sa mal porozprávať s dievčaťom v rohu.“

avatar1 mu povie aby si šiel vypočuť vtip a avatar2 mu povedal aby šiel riešiť hádanku – za hádanku dostane disketu s novou mini-hrou do automatu

**Zuzka:** „Ahoj, ty si tu nový, že?“

**Jozef:** „Áno.“

**Zuzka:** „Vidím, že si celkom odvážny.“

**Jozef:** „Prosím?“

**Zuzka:** „Hneď v prvý deň sa oddeliť od skupinky, dokonca ešte pred ubytovaním.“

**Jozef:** „No hej ... ja som taký dobrodruh.“

**Zuzka:** „Tak vieš čo? Ak mi odpovieš správne na moju nasledujúcu otázku, dám ti niečo, čo by ti neskôr mohlo pomôcť, ok?“

**Jozef:** „Keď myslíš.“

**Zuzka:** „Tak počúvaj pozorne. Jedna z týchto vecí je iná ako ostatné. Kotlík, čarovná guľa, telefón a prútik. Čo myslíš?“

**Hráč:** „Kotlík!“

**Zuzka:** „Ehm ... tak ešte raz. Kotlík, čarovná guľa, TELEFÓN a prútik!“

**Hráč:** „Telefón?“

**Zuzka:** „Správne! Od začiatku som tušila, že si génius. No dobre, tu je moja odmena.“

\*daruje disketu

**Zuzka:** „Uvidíš, neskôr sa zide.“

**Anča:** „Vieš čo dostaneš keď dáš zmraziť jaternicu?“

**Jozef:** „???“

**Anča:** „SNEHURKU! Hahahahaha chápeš? Sne .... Hurku. Akože sneh a hurka.... No nič, pokračuj v ceste.“

**Jozef:** „?!?!?!“

**3. scéna** – ďalej po chodbe – hovoriace dvere čo chcú heslo – avatar1 hovorí aby sa pozrel doľava, avatar2 povie doprava

Dvere (najprotivnejším a najškreklavejším hlasom): „HEESLOO!“

**Avatar 1:** „Podľa mňa by si sa mal pozrieť doprava a možno tam nájdeš odpoveď.“

**Avatar 2:** „Podľa mňa by si sa mal pozrieť doľava, možno bude odpoveď tam.“

**4. scéna** – sklad upratovačky kde je automat – dá disketu zahrá si hru (flappy harry) – (ak chce ísť hráč von musí najskôr nájsť heslo – inak ho avatar vráti, že nemôže ísť nikam bez hesla) – získa mincu – otvorí sa mu shop (mačkne ikonku shopu) – avatar1 – kúpi si fialový lampáš, avatar2 – kúp si modrý lampáš – každý zasvieti na stene iné heslo (zemiaky, paradajky), výjde von – obe dvere (okrem tých hlavných) zmiznú, prejde cez dvere

**5. scéna** – klubovňa – tuli vaky - 2 dvere (chlapci/dievčatá - schodiská) – stretne toho zlého (hipster),

**Jozef:** „Nazdar, aj ty si tu nový?“

**Vlado pohrdavo:** „Hm?“

**Jozef:** „Ja som Jozef.“

**Vlado:** „Okej.“

\*Vlado odíde

## **KAPITOLA 2:**

hodina rušenia

**POZNÁMKY:**

obchod – george a fred – analógia – klikne sa na ikonku v rohu

ikonka avatarov – recyklujeme logo

15-rokov majú deti

minihry a miniúlohy – miniúlohy – puzzle (zvlášť obrazovka)

Kapitola 1. – ubytovanie na intráku – zoznámenie sa s postavami a hernou mechanikou

Kapitola 2. – chodíme na vyučovanie a tam zistí, že niekto marí výskum

Kapitola 3. – Zistíme, že Vlado marí výskum.

Kapitola 4. – Brutálne vyvrcholenie

Koniec:

## C. Analýza hry The Stanley Parable

Hra je založená na rozhodnutiach a na tom či hráč počúvne rady ktoré mu dáva hlas v jeho hlave.

Na zaujatie hráča je použitý moment prekvapenia, vtipné hlášky hlasu v jeho hlave ktorý reaguje na to čo hráč robí.

Pri pohybe nie je použitá žiadna animácia kolísania, takisto hráč nemá žiadne telo. Pohyb pôsobí celkom príjemným dojmom, chýbala mi len možnosť šprintu a skákania.

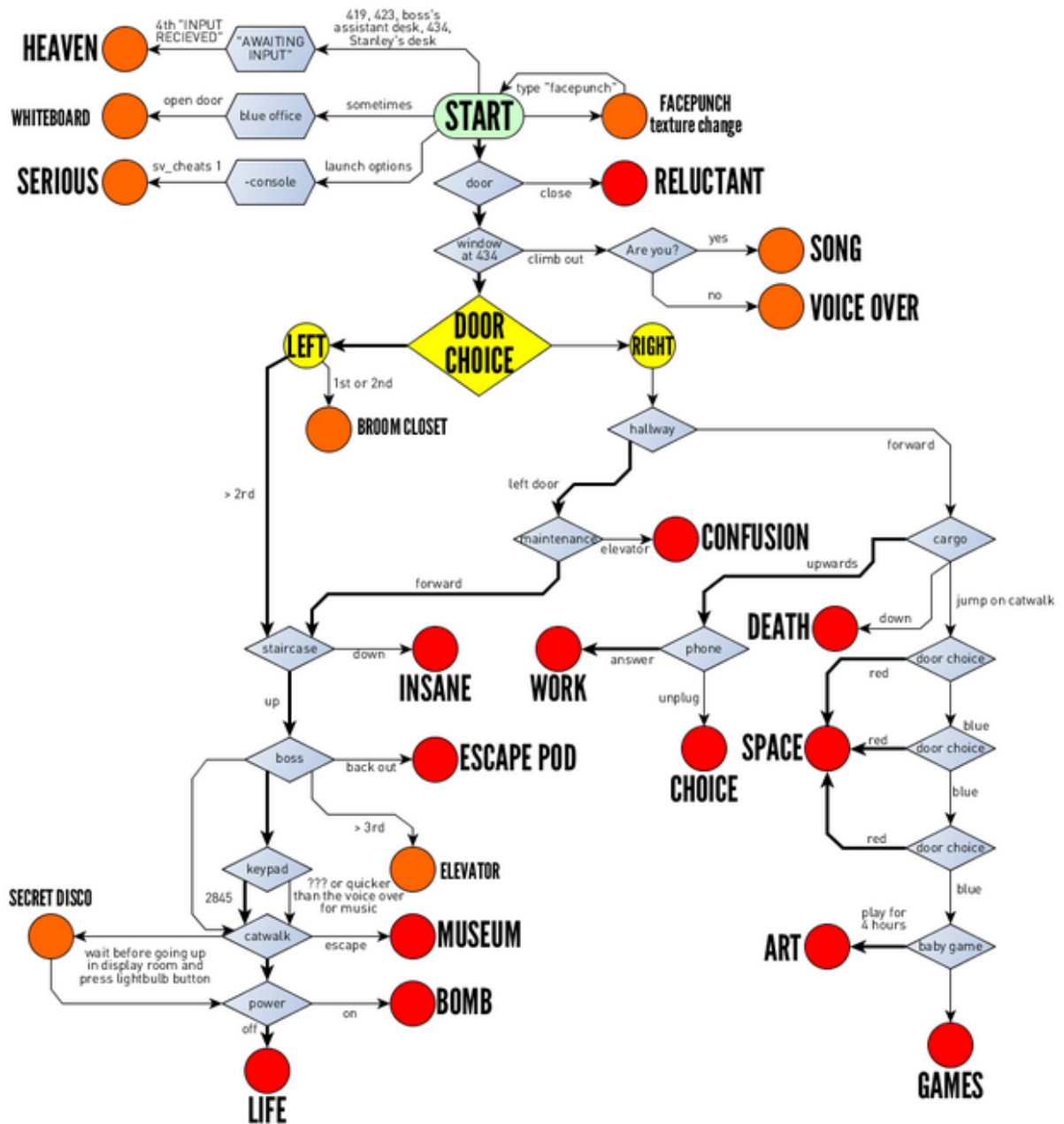
Celá hra je postavená na prekvapivých momentoch ako napríklad: [http://thestanleyparable.wikia.com/wiki/Special:NewFiles?file=2014-12-08\\_00001.jpg](http://thestanleyparable.wikia.com/wiki/Special:NewFiles?file=2014-12-08_00001.jpg)

Vtipne achievements. Hry v hre, rozprávač si s hracom robí čo chce.

Z analýzy herného videa a hry sme prišli na niektoré možné vylepšenia

Postrehy na zlepšenie:

- Malá voľnosť výberu
- Málo možností reagovania zo svojim okolím
- Aj keď má hra celkom jednoduchú grafiku pri najväčších detailoch seká
- Hráč má málo možností ovládania (okrem základného pohybu má možnosti len vykonať akciu a čupnúť si)
- Bolo by zaujímavé keby Stanley mal možnosť odpovedi, aby to nebol príbeh len rozprávača



Obrázok 48 Vývojový diagram hry Stanley Parable

## D. Prvotné nápady na príbeh

### **The neviem co:**

Ta budova je inteligentna a udržbar sposobil nejaky skrat a UI budovy mu vbehne do hlavy a zacne ho ovladat. Ludia nezmiznu len v gebeni su mu skryte. Ked zrazu vidi pizziera lebo mu to ta UI dovoli, poda si s nim ruky a ta UI prebehne do pizziera. A takto by to slo dalej a dalej. Ten ovladany clovek, by si myslel, ze mu nejaky vratnik co ho vidi cez kamery radi, ako ma ist von, lebo je nejaky poplach.

A koniec by sa vsetci zisli a kukali by si realne zabery z bezpecnostnych kamier ako tam bludili medzi obycajnymi ludmi jak taki oni.

### **The near dead ending:**

Príde starší clovek(elektrikár/ profesor.. Hoci čo) a stane sa mu nehoda (zošľahá ho prúd, strelia ho....) ako tak leží na zemi začne sa mu pred očami odohrávať celý život, obdobie keď roznášal pizzu, kopal kanále, robil na výskume.. Rôzne veci od teraz až po jeho detstvo. Konce by mohli byť 3, buď zomrie a ocitne sa v nebi.

Alebo sa rozhodne že chce žiť a vráti sa s5 do svojho tela.

Alebo si uvedomí že jeho život mohol byť iný a začne od detstva a postupne jak bude prechádzať cez miestnosti bude starnúť a vyvíjať sa.

V tomto budú vlastne miestnosti úseky z jeho života ktoré zažil, a prechodom do ďalšej "miestnosti" sa mu vyjaví iný úsek jeho života.

### **The government:**

**Veta:** Do budovy su zlakani vsetci ti typci lebo vlada s nejakym super novym advanced strojom zistila ze maju vynimocne geny, ktore mozu pomoct ludstvu.

**Detaily:** Potrebovali si ich vsak otestovat ci budu pasovat spolu alebo whatever a preto ich nahnali do jednej budovy kde mali riesit ulohy /\*akoze by sa v idealnom pripade rozhodli tak, ze by nevedomky pomohli niekomu inemu v tej budove\*/

Fungovalo im to spolu vsak tak dobre ze sa stretli (to sa nemalo stat) a odhalili tento plan a na konci sa teda rozhodnu ci sa nechaju zneuzit vladou (to by bol ten nas avatar/system)

### **The lunatic:**

**Veta:** Vsetkych ich tam nenapadne dostal nejaky psychicky narusený vrah, ktorý tuto budovu pouziva na zabavenie svojich obeti predtym nez ich zabije.

**Detaily:** Na konci vsetkych bud pozabija alebo sa im podari dostat sa dokopy a zneskodnit vraha (podla volby). Hrac by mohol aj jednu z postav hrat priamo za toho vraha ale tak ze by o tom vlastne nevedel (chcel by nastrazit pascu na niekoho lebo by si myslel ze ho tam niekto sleduje a pod. To by mu radil ten system/avatar)

### **Niečo nové, skôr psychologické**

**Veta:** Elektrikára, ktorý nie je práve spokojný so svojím životom, kopne el. prúd a kým je v bezvedomí jeho podvedomie mu v hlave vytvorí reprezentácie jobov, ktoré by aj chcel robiť a nechá ho „vyskúšať si to“. Všetko sa teda odohráva v jeho hlave a ostatné hrateľné postavy sú výplodom jeho mysle.

**Detaily:** Prestriedané postavy sa líšia len povoláním – reprezentujú zamestnania, ktoré v živote mohol robiť keby viac makal alebo to mal ako vysnívané povolanie keď bol dieťa alebo aj povolania, ktoré pred tým naozaj robil a bol neúspešný/vykašlal sa na to. Jeho podvedomie ich vytvorilo aby mu ukázalo čo všetko mohol v živote dosiahnuť (a ešte vlastne môže). Avatar je to jeho podvedomie (resp. ten hlások v hlave čo mu vždy vravel, že má naviac), ktoré sa ho snaží navigovať aby v tej budove splnil vždy konkrétny task pre jedno povolanie. Žiadny ľudia tu nie su lebo je to v jeho hlave a niečo v živote dosiahne len človek čo maká sám na sebe.



## E. Popis dialógov v hre

### E-1 Scéna 1



Ahoj Jozef. Prepáč za to prudké uvítanie, potrebovali sme upútať tvoju pozornosť. Vieš kto sme?

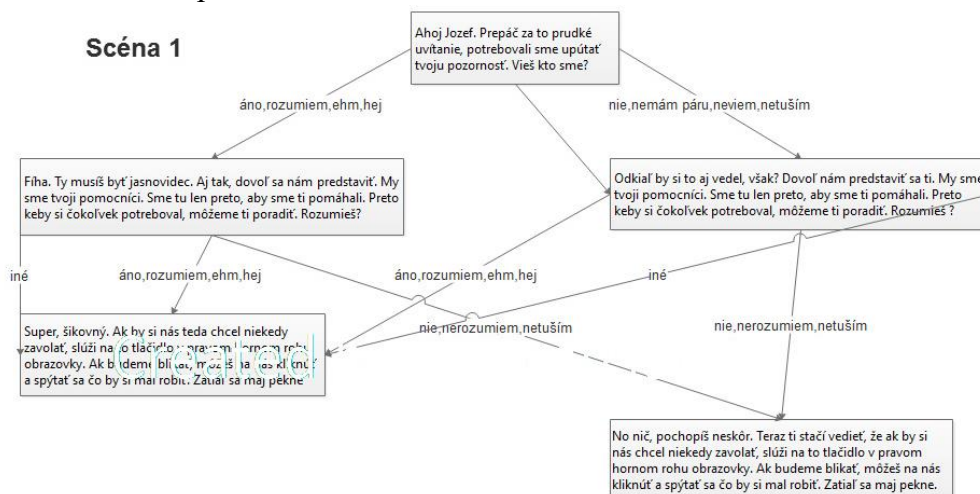
Odkiaľ by si to aj vedel, však? Dovoľ nám predstaviť sa ti. My sme tvoji pomocníci. Sme tu len preto, aby sme ti pomáhali. Preto keby si čokoľvek potreboval, môžeme ti poradiť. Rozumieš

Fíha. Ty musíš byť jasnovidec. Aj tak, dovoľ sa nám predstaviť. My sme tvoji pomocníci. Sme tu len preto, aby sme ti pomáhali. Preto keby si čokoľvek potreboval, môžeme ti poradiť. Rozumieš

Super, šikovný. Ak by si nás teda chcel niekedy zavolať, slúži na to tlačidlo v pravom hornom rohu obrazovky. Ak budeme blikať, môžeš na nás kliknúť a spýtať sa čo by si mal robiť. Zatiaľ sa maj pekne

No nič, pochopíš neskôr. Teraz ti stačí vedieť, že ak by si nás chcel niekedy zavolať, slúži na to tlačidlo v pravom hornom rohu obrazovky. Ak budeme blikať, môžeš na nás kliknúť a spýtať sa čo by si mal robiť. Zatiaľ sa maj pekne.

Počkaj, chvíľu som nepočúval, čo si hovoril.



## E-2 Scéna 1

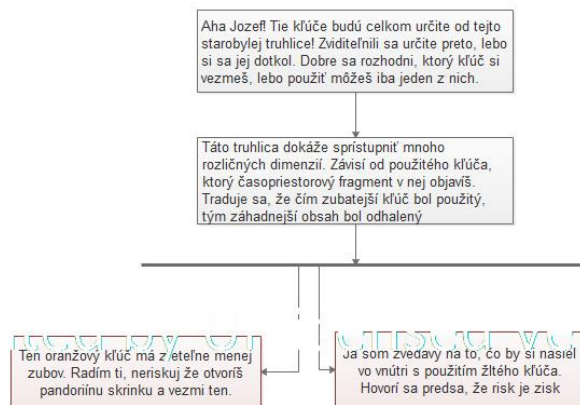
Aha Jozef! Tie kľúče budú celkom určite od tejto starobylej truhlice! Zviditeľnili sa určite preto, lebo si sa jej dotkol. Dobre sa rozhodni, ktorý kľúč si vezmeš, lebo použiť môžeš iba jeden z nich.

Táto truhlica dokáže sprístupniť mnoho rozličných dimenzií. Závisí od použitého kľúča, ktorý časopriestorový fragment v nej objavíš. Traduje sa, že čím zubatejší kľúč bol použitý, tým záhadnejší obsah bol odhalený.

Ten oranžový kľúč má zreteľne menej zubov. Radím ti, neriskuj že otvoríš pandoriínu skrinku a vezmi ten.

Ja som zvedavý na to, čo by si našiel vo vnútri s použitím žltého kľúča. Hovorí sa predsa, že risk je zisk.

### Scéna 1



## E-3 Scéna 2

Vidíš v miestnosti ešte niekoho

Opíš nám ako vyzerajú

Ako vyzerajú?

Si si istý, že v tejto miestnosti sa nenachádzajú iné postavy?

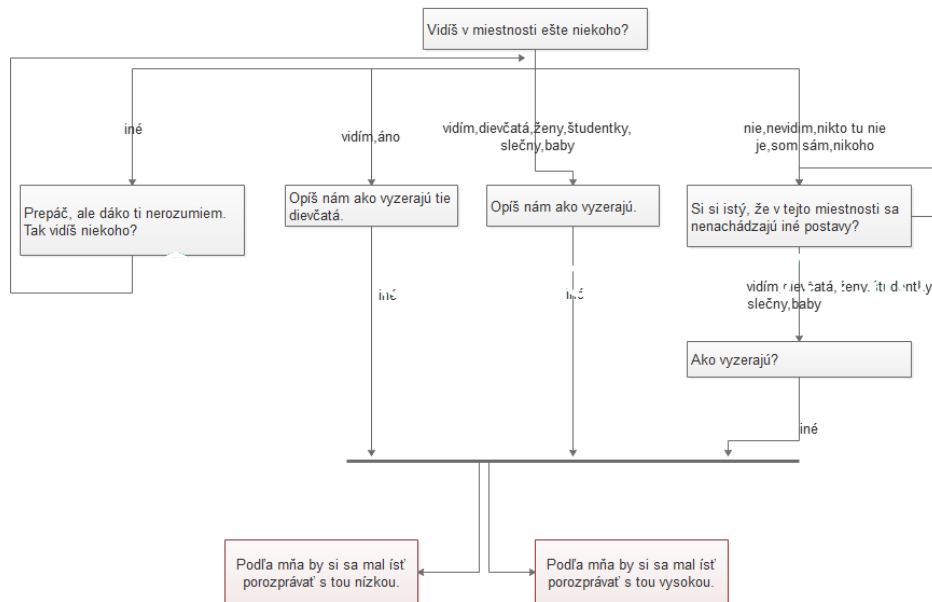
Prepáč, ale dáko ti nerozumiem. Tak vidíš niekoho

Šup šup rýchlo za nimi

Podľa mňa by si sa mal ísť porozprávať s tou nízkou

Podľa mňa by si sa mal ísť porozprávať s tou vysokou

## Scéna 2



## E-4 Scéna 2

Tak počúvaj pozorne. Ktorá z týchto vecí je iná ako ostatné? Kotlík, čarovná guľa, telefón a prútik.

No nič, tak ešte raz. Kotlík, čarovná guľa, telefón alebo prútik.

Spráávne! Od začiatku som tušila, že si génius. No dobre, tu je moja odmena.

Tak ešte raz. Kotlík, čarovná guľa, telefón alebo prútik.

## E-5 Scéna 3

Vieš čo by to heslo mohlo byť

Napadá ti niečo

Oukej, tak to skúsme inak

Čo vidíš okrem týchto škriekajúcich dverí?

Je medzi dvomi dverami nejaký rozdiel?

Môžeme ti dať kľúč od jednych zamknutých dverí. Musíš si ale dobre vybrať, lebo možno sa už naspäť nevrátiš. Chápeš?

Prepáč, chvíľu sme ťa nepočuli. Čo si povedal?

To tiež, ale všimli sme si tu aj iné dvere.

Myslíš, že je medzi tými dverami nejaký rozdiel?

Od ktorých dverí ti máme dať kľúče? Pravých alebo ľavých?

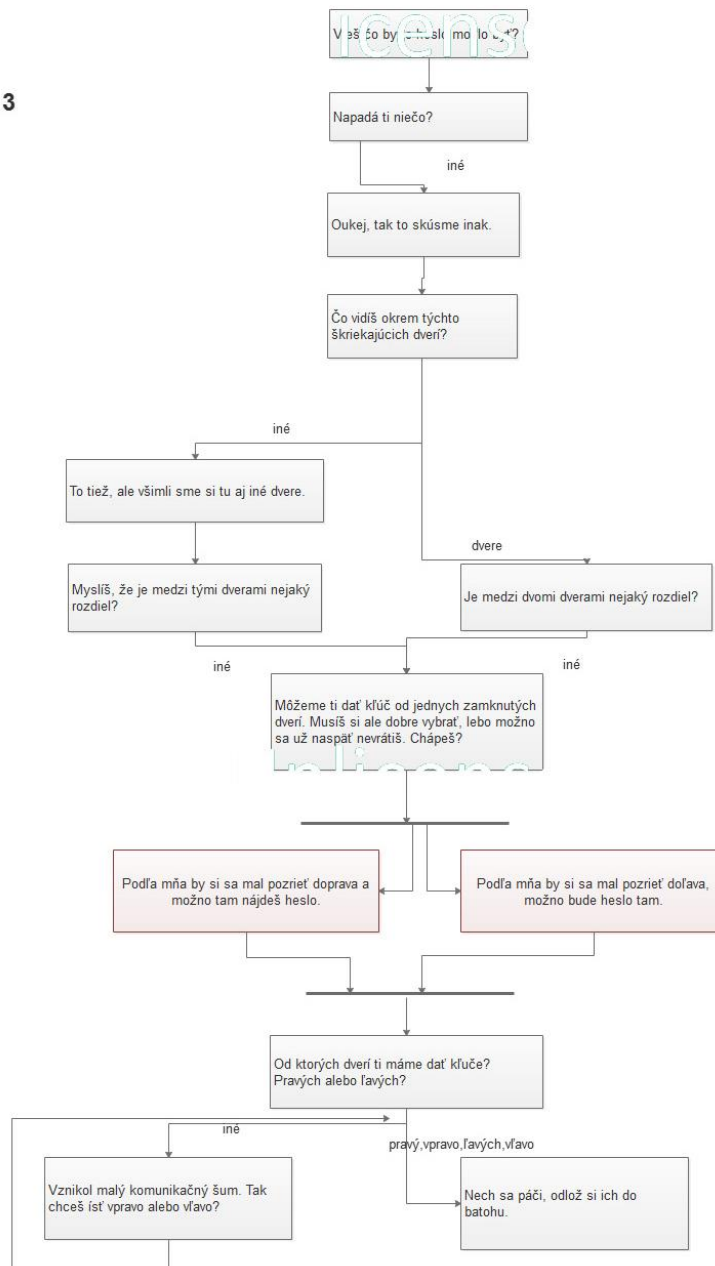
Nech sa páči, odlož si ich do batohu.

Nech sa páči, odlož si ich do batohu.

Vznikol malý komunikačný šum. Tak chceš ísť vpravo alebo vľavo?

Podľa mňa by si sa mal pozrieť doprava a možno tam nájdeš heslo.  
 Podľa mňa by si sa mal pozrieť doľava, možno bude heslo tam.

Scéna 3



## E-6 Scéna 4

Všimli sme si, že si si zahral hru. Koľko mincí sa ti podarilo vyhrať?

Skús sa pozrieť do obchodu, možno si môžeš niečo kúpiť, čo ťa privedie k heslu.

Aké predmety máš na výber?

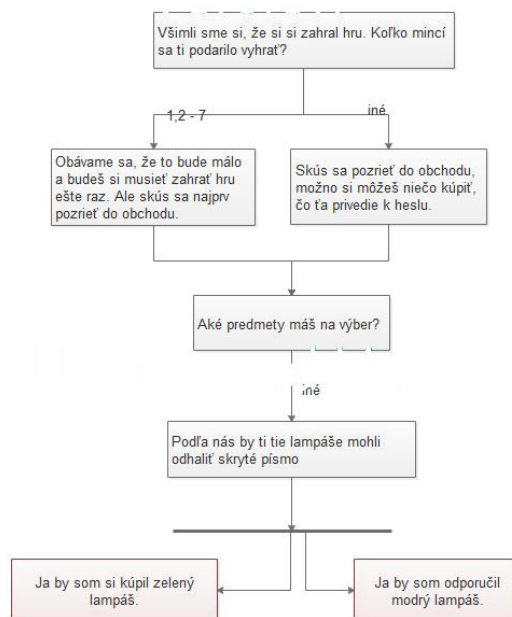
Obávame sa, že to bude málo a budeš si musieť zahrať hru ešte raz. Ale skús sa najprv pozrieť do obchodu.

Podľa nás by ti tie lampáše mohli odhaliť skryté písmo.

Ja by som si kúpil zelený lampáš.

Ja by som odporučil modrý lampáš.

### Scéna 4



## E-7 Scéna 4

Tu asi bude treba použiť niečo z batohu

Čo máš v batohu

Nám sa celkom páči tá disketa. Myslíme si, že by mohla zapnúť automat

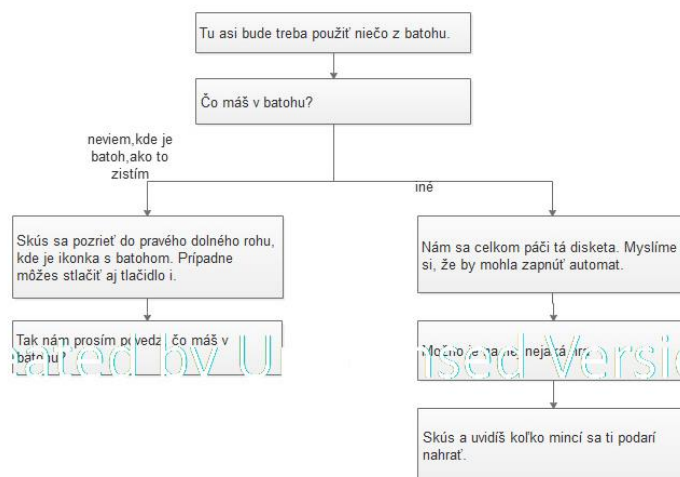
Skús sa pozrieť do pravého dolného rohu, kde je ikonka s batohom. Prípadne môžeš stlačiť aj tlačidlo i.

Tak nám prosím povedz, čo máš v batohu?

Možno je na nej nejaká hra.

Skús a uvidíš koľko mincí sa ti podarí nahrať.

#### Scéna 4



### E-8 Scéna 5

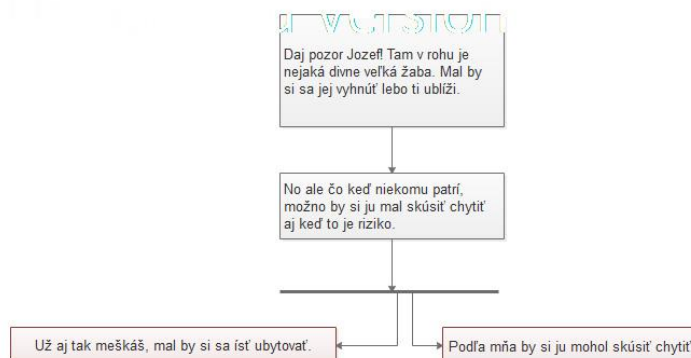
Daj pozor Jozef! Tam v rohu je nejaká divne veľká žaba. Mal by si sa jej vyhnúť lebo ti ublíži.

No ale čo keď niekomu patrí, možno by si ju mal skúsiť chytiť aj keď to je riziko.

Už aj tak meškáš, mal by si sa ísť ubytovať.

Podľa mňa by si ju mohol skúsiť chytiť.

#### Scéna 5



### E-9 Scéna 6

Tak Jozef, konečne si dorazil až sem, na ubytovňu.

Skôr ako si pôjdeš ľahnúť, čaká ťa dôležité rozhodnutie.

Musíš si vybrať, do akej fakulty sa chceš pridať.

V čom myslíš že si dobrý?

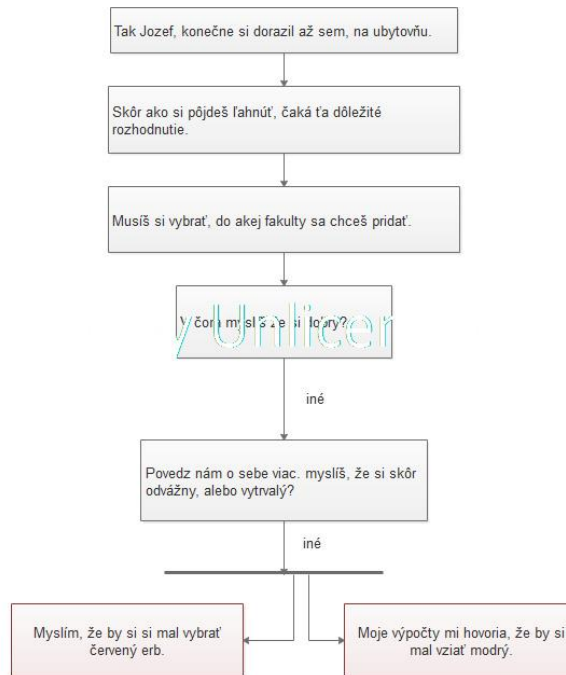
Povedz nám o sebe viac. myslíš, že si skôr odvážny, alebo vytrvalý?

Prosím?

Myslím, že by si si mal vybrať červený erb.

Moje výpočty mi hovoria, že by si mal vziať modrý.

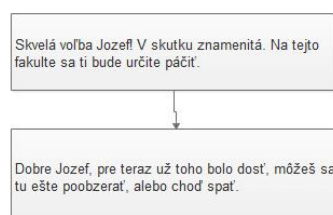
## Scéna 6



## E-10 Scéna 6

Skvelá voľba Jozef! V skutku znamenitá. Na tejto fakulte sa ti bude určite páčiť.  
Dobre Jozef, pre teraz už toho bolo dost', môžeš sa tu ešte poobzerať, alebo choď spať.

## Scéna 6



## F. Rozhodnutia

- Rozhodnutie ci v obchode kúpiť lampas alebo baterku. Každé ma inú ikonku a napokon zobrazí iné heslo.
  - Jozko je v nejakej chodbe kde sa nedá prejsť lebo je tam prepadnutá podlaha alebo niečo. Pred ňou sú nejaké tlačidlá alebo čo ktoré treba stlačiť v správnom poradí aby sa spriechodnila cesta – každý avatar poradí inú cestu
  - Poradenie schovať si nejaký predmet kamsi aby mu ho nezobral Vlado
  - Ulohy/test na hodine kde si treba vyberat odpovede.
  - Jozko získa nejaký kľúč a bude mať pred sebou dve truhlice tak avatari poradia, ktorú ma otvoriť.
  - (Ak by sme robili dačo s toho pôvodného scenára) Keby si mal dáky lektvar namiešať tak mu poradia medzi dvoma
  - Vedľa rusebne druha hodina alchymie – stolik s elixirmi. Pobrat itemy do inventara a hadzat do dzbanu avatari by radili čo brať aby sa namiesal spravny elixir. Recept by mal chybajúce stepy (to doplnia avatari). X stepov tvorby – poradiť len v jednom kroku outputu na konci doniesť profakovy aby sme nevedeli ktorá z tých 3 rad bola zla/dobra.
  - FINAL RADA – prestrihovanie drotikov na rusicke. Ale nevediet ako to dopadlo.
  - Keby sme mali aspoň dve minihry (alebo 2 levely flappy joe) tak mu môže každý radiť zahrať si niečo iné
  - V jednej scéne už máme truhlicu tak mu môže jeden z nich poradiť ju otvoriť (alebo nech mu obaja radia ju otvoriť ale rôznymi kľúčmi napríklad, a na základe toho, ktorým ju otvorí bude v nej niečo iné, kľúče môžeme podľa mňa kľudne zavesiť aj nad truhlicu nech ich hráč nemusí hľadať, alebo keby ich mal hľadať - tam by sa tiež dali vymyslieť ďalšie rozhodnutia napr. jeden mu povie pozri sa tam a druhý mu povie nech sa pozrie inam alebo nech sa ide porozprávať s hádankárkou - mohla by mať aj takéto veci, nielen diskety)
  - V scéne s tromi dverami, hneď po príchode hráča (na ubytovanie a tie dve čo vedú do rovnakej miestnosti) mu môže jeden poradiť nech ide rovno do ubytovne a druhý nech najskôr preskúma čo je za tými ďalšími dverami (alebo nech to neznie tak hlúpo lebo ľudia chcú v hrách vždy všade loziť proste ktorými dvermi ísť najprv) - pričom tie bočné by mohli byť zatiaľ zamknuté ale tie od ubytovne by sa mu prihovorili – takto by sme mohli ukázať hráčovi, že nie všetky rady sú úplne najlepšie
  - Po prvom vtipke bude vtipkárka urazená že sa Jozef nezasmial poriadne. Jeden avatar mu poradí nech jej dá bielu čokoládu, druhý mu povie nech jej dá mliečnu čokoládu. Ak jej dá mliečnu povie „ďakujem túto mám radšej ako bielu“, ak jej dá bielu povie „oo ako si vedel že toto je moja obľúbená“
  - Scéna 5 v rohu miestnosti je žaba, jeden avatar mu povie aby ju zobral, druhý že je to zbytočné nech ju tam nechá. V scéne 6 k nemu príde chlapec/ dievča a spýta sa ho či nevidel jeho žabu, ak ju zobral dá mu ju a on mu poďakuje, ak ju nezobral tak mu len povie že ju videl vedľa v miestnosti. (hráč zatiaľ nevie či je fajn že ju zobral alebo že ju nezobral)
- [
- Avatar1: Daj pozor Jozef! Tam v rohu je nejaká divne veľká žaba. Mal by si sa jej vyhnúť lebo ti ublíži.



Avatar2: No ale čo keď niekomu patrí, možno by si ju mal skúsiť chytiť aj keď to je riziko.

Avatar1: Už aj tak meškáme, mali by sme sa ísť ubytovať.

Avatar2. Ja myslím že by sme ju mohli chytiť.

]

- V scéne 6 posledné veľké rozhodnutie v tejto kapitole bude že jeden avatar mu povie že ak pôjde do pravých dverí pridá sa k združeniu múdrych a bude študovať v tej katedre, druhý mu povie nech ide do ľavých a bude študovať v katedre šikovních

Návrhy rozhodnutí pri využití súčasného stavu hry spolu s návrhmi dialógov (Stefan)

- truhlica: Keď dostanem kľúč, kliknem s ním na truhlicu a ozve sa od avatarov:
  - AV1: No, to je prekvapenie, nie je tu poklad, ale dve malé zatvorené nádoby
  - AV2: Na jednej je napísané „sušené muchy“ a na druhej „sušené dáždovky“ (alebo „živé muchy“)
  - AV1: Na čo by sa ti mohol zísť sušený hmyz?
    - Hráč: je jedno čo povie
  - AV2: Ja by som navrhoval zobrať si muchy
  - AV1: Ja doporučujem dáždovky
  - Takto sa napríklad toto môže použiť neskôr:
    - Žaba: pred tým, ako sa rozhoduje, či žabu zoberie:
    - AV1: tá žaba vyzerá dosť hladná
    - AV2: nemáš nič, čím by si ju nakrmil?
      - Hráč: mám sušené muchy/dáždovky (keywords: muchy/dáždovky, ak nie, tak AV povie “Veď si v truhlici získal muchy/dáždovky”)
- Disketa od Zuzy: po odpovedi na hádanku
  - Zuza: Mám tu 2 rôzne diskety, jedna je veľká a druhá malá, ale môžem ti dať len jednu, máš sa s kým poradiť?
  - Hráč: Áno, mám poradcov/avatarov (avatari začnú blikať).  
  
Po kliknutí na avatarov:
    - AV1: Aký máš problém?
    - Hráč: musím si vybrať jednu z dvoch \*diskiet\* od Zuzy (alebo niečo podobné)
    - AV2: Vieš o tých disketách niečo viac?
    - Hráč: Iba to, že jedna je veľká a druhá malá
    - AV1: malá disketa je lepšia, lebo nezaberie veľa miesta a určite sa tam zmestí veľa dát, ja by som zobral tú malú.
    - AV2: no, ale veľká disketa sa dá použiť do starého počítača, a táto škola je stará, ja radím vybrať si tú veľkú
- Posteľ: po príchode s použitím prístupovej kartičky
  - AV2: Tá prístupová karta je nesmierne dôležitá, dobre si ju stráž.
  - AV1: Bolo by dobre niekde si ju schovať keď budeš spať a nemať ju len tak v batohu.
  - AV2: Nevieš, či by ju niekto nechcel ukradnúť. Aké poznáš skryšy v posteli?
    - Hráč: pod vankúš, pod matrac, pod posteľ,...
  - AV1: Pod vankúšom je veľmi dobrá skryša, lebo ...
  - AV2: Pod matracom je tiež super miesto, lebo ...
- Erb: po rozhodnutí sa do ktorej fakulty bude patriť, pridanie erbu na odev (ak je to možné):

- AV1: Teraz, keď si sa rozhodol, kde budeš patriť, musíš si vybrať kde si chceš odznak svojej fakulty pripnúť.
- AV2: Podľa interných školských predpisov, máš na výber iba 2 časti svojho odevu, a to klobúk, alebo plášť.
- AV1: Čo si myslíš o takých predpisoch, je to dôležité?
  - Hráč:
    - áno, asi hej, možno → AV2: máš pravdu, lebo ostatní si hneď o tebe urobia podľa toho názor
    - nie, vôbec, → AV2: to nemáš pravdu, lebo ostatní si hneď o tebe urobia podľa toho názor
- AV2: Ja by som si ho dal na klobúk. Bude to všetkým viditeľné a ukážeš, že si hrdým členom fakulty.
- AV1: Ja by som si ho dal na plášť. Odznakom pri srdci všetkým ukážeš, že fakulta je pre teba veľmi dôležitá.
- Pomôcka na lúštenie hieroglyfov na dverách do učebni: pred tým, ako sa objaví Vlado
  - AV1: Čo je na tých šedých dverách napísané?
    - Hráč: neviem,...
  - AV2: Bolo by dobre to rozlúštiť.
  - AV1: Ja mám špeciálny dekodovací program SPT, myslím si, že by to rozlúštil
  - AV2: Ja zase mám originálny algoritmus KMZ na lúštenie hieroglyfov, navrhujem ten.
  - AV1: Ktorý chceš, aby sme vyskúšali?
  - Hráč: SPT/KMZ
  - AV2: Vyskúšali sme najprv jeden a potom aj druhý, ale žiaľ, nepodarilo sa nám nápis rozlúštiť

## G. Plagát TP Cup

Ako tím sme sa zúčastnili aj súťaže TP Cup na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave a v rámci študentskej konferencie IIT.Src sme prezentovali našu prácu. V tejto prílohe pripíjame plagát, ktorý bol vystavovaný v rámci tejto súťaže na konferencii.



## H. Schémy Xml súborov

### H-1 Model interakcií

```
<xs:schema targetNamespace="http://team20-15.studenti.fiit.stuba.sk"
  elementFormDefault="qualified"
  xmlns="http://team20-15.studenti.fiit.stuba.sk"
  xmlns:mstns="http://team20-15.studenti.fiit.stuba.sk"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:complexType name="gameEntityType">
    <xs:sequence>
      <xs:element name="name"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="gameEntitiesType">
    <xs:sequence>
      <xs:element name="gameEntity" type="gameEntityType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="gameEntityRefType">
    <xs:attribute name="ref" type="xs:string" use="required"/>
  </xs:complexType>

  <xs:complexType name="adviceType">
    <xs:attribute name="id" type="xs:integer" use="required"/>
  </xs:complexType>
  <xs:complexType name="advicesType">
    <xs:sequence>
      <xs:element name="advice" type="adviceType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="adviceRefType">
    <xs:attribute name="ref" type="xs:integer" use="required"/>
  </xs:complexType>

  <xs:complexType name="sceneType">
    <xs:sequence>
      <xs:element name="name"/>
      <xs:element name="gameEntity" type="gameEntityRefType"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="scenesType">
    <xs:sequence>
      <xs:element name="scene" type="sceneType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="sceneRefType">
    <xs:attribute name="ref" type="xs:string" use="required"/>
  </xs:complexType>

  <xs:complexType name="interactionTypeType">
    <xs:sequence>
      <xs:element name="name"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

        </xs:sequence>
        <xs:attribute name="id" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:complexType name="interactionTypesType">
        <xs:sequence>
            <xs:element name="interactionType" type="interactionTypeType"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="interactionTypeRefType">
        <xs:attribute name="ref" type="xs:string" use="required"/>
    </xs:complexType>

    <xs:complexType name="stateType">
        <xs:sequence>
            <xs:element name="name"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:complexType name="statesType">
        <xs:sequence>
            <xs:element name="state" type="stateType"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="stateRefType">
        <xs:attribute name="ref" type="xs:string" use="required"/>
    </xs:complexType>

    <xs:complexType name="interactionType">
        <xs:sequence>
            <xs:element name="source" type="gameEntityRefType"
minOccurs="1" maxOccurs="1"/>
            <xs:element name="destination" type="gameEntityRefType"
minOccurs="1" maxOccurs="1"/>
            <xs:element name="scene" type="sceneRefType" minOccurs="1"
maxOccurs="1"/>
            <xs:element name="interactionType"
type="interactionTypeRefType" minOccurs="1" maxOccurs="1"/>
            <xs:element name="requiredObjects" type="requiredObjectsType"/>
            <xs:element name="possibilities" type="possibilitiesType"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:complexType name="interactionsType">
        <xs:sequence>
            <xs:element name="interaction" type="interactionType"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="requiredObjectsType">
        <xs:sequence>
            <xs:element name="requiredObject" type="gameEntityRefType"
minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="possibilityType">
        <xs:sequence>
            <xs:element name="conditions" type="conditionsType"/>
            <xs:element name="actions" type="actionsType"/>
        </xs:sequence>
    </xs:complexType>

```

```

    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
    <xs:attribute name="advice_ref" type="xs:integer" use="optional"/>
  </xs:complexType>
  <xs:complexType name="possibilitiesType">
    <xs:sequence>
      <xs:element name="possibility" type="possibilityType"
minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="knowledgeType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="attr" type="xs:string"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="conditionType">
    <xs:sequence>
      <xs:element name="gameEntity" type="gameEntityRefType"
minOccurs="1" maxOccurs="1"/>
      <xs:choice minOccurs="1" maxOccurs="unbounded">
        <xs:element name="state" type="stateRefType" minOccurs="0"
maxOccurs="1"/>
        <xs:element name="knowledge" type="knowledgeType"
minOccurs="0" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="conditionsType">
    <xs:sequence>
      <xs:element name="condition" type="conditionType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="changeStateType">
    <xs:sequence>
      <xs:element name="gameEntity" type="gameEntityRefType"
maxOccurs="1"/>
      <xs:element name="state" type="stateRefType" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="addKnowledgeType">
    <xs:sequence>
      <xs:element name="gameEntity" type="gameEntityRefType"
minOccurs="1" maxOccurs="1"/>
      <xs:element name="knowledge" type="xs:string" minOccurs="1"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="leaveSceneType">
    <xs:sequence>
      <xs:element name="toScene" type="sceneRefType" minOccurs="1"
maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="changeSpriteLayerType">

```

```

    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="ref" type="xs:string" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="changeHotspotStateType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="ref" type="xs:string" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="sendMessageType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="ref" type="xs:string" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="playSoundType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="ref" type="xs:string" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="destroyGameObjectType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="ref" type="xs:string" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="pickGameObjectType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="ref" type="xs:string" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="unityActionType">
    <xs:choice minOccurs="1" maxOccurs="1">
      <xs:element name="changeSpriteLayer"
type="changeSpriteLayerType"/>
      <xs:element name="changeHotspotState"
type="changeHotspotStateType"/>
      <xs:element name="sendMessage" type="sendMessageType"/>
      <xs:element name="playSound" type="playSoundType"/>
      <xs:element name="destroyGameObject"
type="destroyGameObjectType"/>
      <xs:element name="pickGameObject" type="pickGameObjectType"/>
      <xs:element name="openInventory" type="openInventoryType"/>
      <xs:element name="openShop" type="openShopType"/>
    </xs:choice>

```

```

</xs:complexType>

<xs:complexType name="openInventoryType">
</xs:complexType>

<xs:complexType name="openShopType">
</xs:complexType>

<xs:complexType name="startAvatarBlinkType">
</xs:complexType>

<xs:complexType name="stopAvatarBlinkType">
</xs:complexType>

<xs:complexType name="itemType">
  <xs:attribute name="ref" type="xs:string" use="required"/>
</xs:complexType>
<xs:complexType name="addItemType">
  <xs:sequence>
    <xs:element name="item" type="itemType" minOccurs="1"
maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="removeItemType">
  <xs:sequence>
    <xs:element name="item" type="itemType" minOccurs="1"
maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="doConversationType">
  <xs:simpleContent>
    <xs:extension base="xs:integer">
      <xs:attribute name="advice_ref" type="xs:integer"
use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="actionsType">
  <xs:choice minOccurs="1" maxOccurs="unbounded">
    <xs:element name="changeState" type="changeStateType"
minOccurs="0"/>
    <xs:element name="addKnowledge" type="addKnowledgeType"
minOccurs="0"/>
    <xs:element name="executeDialog" minOccurs="0"/>
    <xs:element name="doConversation" type="doConversationType"
minOccurs="0"/>
    <xs:element name="sayJoke" minOccurs="0"/>
    <xs:element name="addItem" type="addItemType" minOccurs="0"/>
    <xs:element name="removeItem" type="removeItemType"
minOccurs="0"/>
    <xs:element name="leaveScene" type="leaveSceneType"
minOccurs="0"/>
    <xs:element name="unityAction" type="unityActionType"
minOccurs="0"/>
    <xs:element name="startAvatarBlink" type="startAvatarBlinkType"
minOccurs="0"/>
    <xs:element name="stopAvatarBlink" type="stopAvatarBlinkType"
minOccurs="0"/>
  </xs:choice>

```



```

</xs:complexType>

<xs:element name="definitions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="gameEntities" type="gameEntitiesType"/>
      <xs:element name="scenes" type="scenesType"/>
      <xs:element name="interactionTypes"
type="interactionTypesType"/>
      <xs:element name="states" type="statesType"/>
      <xs:element name="advices" type="advicesType"/>
      <xs:element name="interactions" type="interactionsType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:key name="sceneIdKey">
    <xs:selector xpath="mstns:scenes/mstns:scene"></xs:selector>
    <xs:field xpath="@id"></xs:field>
  </xs:key>
  <xs:key name="gameEntityIdKey">
    <xs:selector
xpath="mstns:gameEntities/mstns:gameEntity"></xs:selector>
    <xs:field xpath="@id"></xs:field>
  </xs:key>
  <xs:key name="interactionTypeIdKey">
    <xs:selector
xpath="mstns:interactionTypes/mstns:interactionType"></xs:selector>
    <xs:field xpath="@id"></xs:field>
  </xs:key>
  <xs:key name="stateIdKey">
    <xs:selector xpath="mstns:states/mstns:state"></xs:selector>
    <xs:field xpath="@id"></xs:field>
  </xs:key>
  <xs:key name="interactionIdKey">
    <xs:selector
xpath="mstns:interactions/mstns:interaction"></xs:selector>
    <xs:field xpath="@id"></xs:field>
  </xs:key>
  <xs:key name="interactionPossibilityIdKey">
    <xs:selector
xpath="mstns:interactions/mstns:interaction/mstns:possibilities/mstns:possi
bility"></xs:selector>
    <xs:field xpath="@id"></xs:field>
  </xs:key>
  <xs:key name="adviceIdKey">
    <xs:selector xpath="mstns:advices/mstns:advice"></xs:selector>
    <xs:field xpath="@id"></xs:field>
  </xs:key>
  <xs:keyref name="sceneGameEntityRefId"
refer="mstns:gameEntityIdKey">
    <xs:selector
xpath="mstns:scenes/mstns:scene/mstns:gameEntity"/>
    <xs:field xpath="@ref"/>
  </xs:keyref>
  <xs:keyref name="interactionSourceRefId"
refer="mstns:gameEntityIdKey">
    <xs:selector
xpath="mstns:interactions/mstns:interaction/mstns:source"/>
    <xs:field xpath="@ref"/>
  </xs:keyref>
  <xs:keyref name="interactionDestRefId"
refer="mstns:gameEntityIdKey">

```

```

        <xs:selector
xpath="mstns:interactions/mstns:interaction/mstns:destination"/>
        <xs:field xpath="@ref"/>
    </xs:keyref>
    <xs:keyref name="interactionSceneRefId" refer="mstns:sceneIdKey">
        <xs:selector
xpath="mstns:interactions/mstns:interaction/mstns:scene"/>
        <xs:field xpath="@ref"/>
    </xs:keyref>
    <xs:keyref name="requiredObjectRefId"
refer="mstns:gameEntityIdKey">
        <xs:selector
xpath="mstns:interactions/mstns:interaction/mstns:requiredObjects/mstns:req
uiredObject"/>
        <xs:field xpath="@ref"/>
    </xs:keyref>
    <xs:keyref name="interactionTypeRefId"
refer="mstns:interactionTypeIdKey">
        <xs:selector
xpath="mstns:interactions/mstns:interaction/mstns:interactionType"/>
        <xs:field xpath="@ref"/>
    </xs:keyref>
    <xs:keyref name="conditionObjectRefId"
refer="mstns:gameEntityIdKey">
        <xs:selector
xpath="mstns:interactions/mstns:interaction/mstns:possibilities/mstns:possi
bility/mstns:conditions/mstns:condition/mstns:gameEntity"/>
        <xs:field xpath="@ref"/>
    </xs:keyref>
    <xs:keyref name="conditionStateRefId" refer="mstns:stateIdKey">
        <xs:selector
xpath="mstns:interactions/mstns:interaction/mstns:possibilities/mstns:possi
bility/mstns:conditions/mstns:condition/mstns:state"/>
        <xs:field xpath="@ref"/>
    </xs:keyref>
    <xs:keyref name="changeStateObjectRefId"
refer="mstns:gameEntityIdKey">
        <xs:selector
xpath="mstns:interactions/mstns:interaction/mstns:possibilities/mstns:possi
bility/mstns:actions/mstns:changeState/mstns:gameEntity"/>
        <xs:field xpath="@ref"/>
    </xs:keyref>
    <xs:keyref name="changeStateStateRefId" refer="mstns:stateIdKey">
        <xs:selector
xpath="mstns:interactions/mstns:interaction/mstns:possibilities/mstns:possi
bility/mstns:actions/mstns:changeState/mstns:state"/>
        <xs:field xpath="@ref"/>
    </xs:keyref>
    <xs:keyref name="addKnowledgeObjectRefId"
refer="mstns:gameEntityIdKey">
        <xs:selector
xpath="mstns:interactions/mstns:interaction/mstns:possibilities/mstns:possi
bility/mstns:actions/mstns:addKnowledge/mstns:gameEntity"/>
        <xs:field xpath="@ref"/>
    </xs:keyref>
    <xs:keyref name="leaveSceneRefId" refer="mstns:sceneIdKey">
        <xs:selector
xpath="mstns:interactions/mstns:interaction/mstns:possibilities/mstns:possi
bility/mstns:actions/mstns:leaveScene/mstns:toScene"/>
        <xs:field xpath="@ref"/>
    </xs:keyref>

```

```
    <xs:keyref name="conversationAdviceRefId"
refer="mstns:adviceIdKey">
    <xs:selector
xpath="mstns:interactions/mstns:interaction/mstns:possibilities/mstns:possi
bility/mstns:actions/mstns:doConversation"></xs:selector>
    <xs:field xpath="@advice_ref"></xs:field>
    </xs:keyref>
    <xs:keyref name="possibilityAdviceRefId" refer="mstns:adviceIdKey">
    <xs:selector
xpath="mstns:interactions/mstns:interaction/mstns:possibilities/mstns:possi
bility"/>
    <xs:field xpath="@advice_ref"></xs:field>
    </xs:keyref>
  </xs:element>
</xs:schema>
```

## H-2 Model dialógov

```
<xs:schema targetNamespace="http://team20-15.studenti.fiit.stuba.sk/dialog"
elementFormDefault="qualified"
  xmlns="http://team20-15.studenti.fiit.stuba.sk/dialog"
  xmlns:mstns="http://team20-15.studenti.fiit.stuba.sk/dialog"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:complexType name="gameEntityType">
    <xs:sequence>
      <xs:element name="name"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="gameEntitiesType">
    <xs:sequence>
      <xs:element name="gameEntity" type="gameEntityType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="gameEntityRefType">
    <xs:attribute name="ref" type="xs:string" use="required"/>
  </xs:complexType>

  <xs:complexType name="interactionTypeType">
    <xs:sequence>
      <xs:element name="name"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="interactionTypesType">
    <xs:sequence>
      <xs:element name="interactionType" type="interactionTypeType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="interactionTypeRefType">
    <xs:attribute name="ref" type="xs:string" use="required"/>
  </xs:complexType>

  <xs:complexType name="dialogType">
    <xs:sequence>
      <xs:element name="interactionPossibility"
type="interactionPossibilityRefType"/>
      <xs:element name="lines" type="linesType"/>
      <xs:element name="interaction" type="interactionType"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
    <xs:attribute name="type" type="xs:string" use="optional"/>
  </xs:complexType>
  <xs:complexType name="dialogsType">
    <xs:sequence>
      <xs:element name="dialog" type="dialogType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="interactionType">
    <xs:sequence>
      <xs:element name="source" type="gameEntityRefType"
minOccurs="1" maxOccurs="1"/>

```

```

        <xs:element name="destination" type="gameEntityRefType"
minOccurs="1" maxOccurs="1"/>
        <xs:element name="interactionType"
type="interactionTypeRefType" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="lineType">
    <xs:sequence>
        <xs:element name="gameEntity" type="gameEntityRefType"/>
        <xs:element name="text" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="linesType">
    <xs:sequence>
        <xs:element name="line" type="lineType" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="interactionPossibilityRefType">
    <xs:attribute name="ref" type="xs:string" use="required"/>
</xs:complexType>

<xs:element name="definitions">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="gameEntities" type="gameEntitiesType"/>
            <xs:element name="interactionTypes"
type="interactionTypesType"/>
            <xs:element name="dialogs" type="dialogsType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:key name="gameEntityIdKey">
        <xs:selector
xpath="mstns:gameEntities/mstns:gameEntity"></xs:selector>
        <xs:field xpath="@id"></xs:field>
    </xs:key>
    <xs:key name="interactionTypeIdKey">
        <xs:selector
xpath="mstns:interactionTypes/mstns:interactionType"></xs:selector>
        <xs:field xpath="@id"></xs:field>
    </xs:key>
    <xs:keyref name="dialogLineGameEntityRefId"
refer="mstns:gameEntityIdKey">
        <xs:selector
xpath="mstns:dialogs/mstns:dialog/mstns:lines/mstns:line/mstns:gameEntity"/
>
        <xs:field xpath="@ref"/>
    </xs:keyref>
    <xs:keyref name="interactionTypeRefId"
refer="mstns:interactionTypeIdKey">
        <xs:selector
xpath="mstns:dialogs/mstns:dialog/mstns:interaction/mstns:interactionType"/
>
        <xs:field xpath="@ref"/>
    </xs:keyref>
    <xs:keyref name="interactionSourceRefId"
refer="mstns:gameEntityIdKey">
        <xs:selector
xpath="mstns:dialogs/mstns:dialog/mstns:interaction/mstns:source"/>

```

```
        <xs:field xpath="@ref" />
    </xs:keyref>
    <xs:keyref name="interactionDestRefId"
refer="mstns:gameEntityIdKey">
        <xs:selector
xpath="mstns:dialogs/mstns:dialog/mstns:interaction/mstns:destination"/>
        <xs:field xpath="@ref" />
    </xs:keyref>
</xs:element>
</xs:schema>
```

## H-3 Model konverzácií

```
<xs:schema targetNamespace="http://team20-
15.studenti.fiit.stuba.sk/avatarConversation"
elementFormDefault="qualified"
xmlns="http://team20-
15.studenti.fiit.stuba.sk/avatarConversation"
xmlns:mstns="http://team20-
15.studenti.fiit.stuba.sk/avatarConversation"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:complexType name="gameEntityType">
    <xs:sequence>
      <xs:element name="name"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="gameEntitiesType">
    <xs:sequence>
      <xs:element name="gameEntity" type="gameEntityType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="gameEntityRefType">
    <xs:attribute name="ref" type="xs:string" use="required"/>
  </xs:complexType>

  <xs:complexType name="interactionTypeType">
    <xs:sequence>
      <xs:element name="name"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="interactionTypesType">
    <xs:sequence>
      <xs:element name="interactionType" type="interactionTypeType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="avatarSentenceType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="id" type="xs:string" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="avatarSentenceEntityType">
    <xs:sequence>
      <xs:element name="sentence" type="avatarSentenceType"
minOccurs="1" maxOccurs="1"/>
      <xs:element name="waitForPlayerAnswer" type="isFinalType"
minOccurs="0" maxOccurs="1"/>
      <xs:element name="gameEntity" type="gameEntityRefType"
minOccurs="0" maxOccurs="1"/>
      <xs:element name="interaction" type="interactionType"
minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="useTextInput" type="isFinalType"
minOccurs="0" maxOccurs="1"/>
      <xs:element name="isFinal" type="isFinalType" minOccurs="0"
maxOccurs="1"/>
      <xs:element name="advice" type="adviceType" minOccurs="0">
```

```

maxOccurs="1"/>

    </xs:sequence>
</xs:complexType>

<xs:complexType name="avatarSentencesType">
    <xs:sequence>
        <xs:element name="sentenceEntity"
type="avatarSentenceEntityType" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="avatarAdviceType">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="id" type="xs:string" use="required"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="avatarAdvicesType">
    <xs:sequence>
        <xs:element name="advice" type="avatarAdviceType" minOccurs="1"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="keyWordsType">
    <xs:simpleContent>
        <xs:extension base="xs:string">
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="adviceType">
    <xs:simpleContent>
        <xs:extension base="xs:string">
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="isFinalType">
    <xs:simpleContent>
        <xs:extension base="xs:string">
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="responseTargetType">
    <xs:attribute name="ref" type="xs:string" use="required"/>
</xs:complexType>
<xs:complexType name="possibleResponseType">
    <xs:sequence>
        <xs:element name="keyWords" type="keyWordsType" minOccurs="1"/>
        <xs:element name="responseTarget" type="responseTargetType"
minOccurs="1"/>
        <xs:element name="advice" type="adviceType" minOccurs="0"/>
        <xs:element name="isFinal" type="isFinalType" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="possibleResponsesType">
    <xs:sequence>
        <xs:element name="possibleResponse" type="possibleResponseType"
minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

```



```

<xs:complexType name="avatarSentenceRefType">
  <xs:attribute name="ref" type="xs:string" use="required"/>
</xs:complexType>
<xs:complexType name="playerResponseType">
  <xs:sequence>
    <xs:element name="avatarSentence"
type="avatarSentenceRefType"/>
    <xs:element name="possibleResponses"
type="possibleResponsesType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="playerResponsesType">
  <xs:sequence>
    <xs:element name="playerResponse" type="playerResponseType"
minOccurs="1" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="interactionTypeRefType">
  <xs:attribute name="ref" type="xs:string" use="required"/>
</xs:complexType>
<xs:complexType name="defaultResponseType">
  <xs:sequence>
    <xs:element name="responseTarget" type="responseTargetType"/>
    <xs:element name="isFinal" type="isFinalType" minOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="conversationType">
  <xs:sequence>
    <xs:element name="avatarSentences" type="avatarSentencesType"/>
    <xs:element name="avatarAdvices" type="avatarAdvicesType"/>
    <xs:element name="defaultResponse" type="defaultResponseType"/>
    <xs:element name="playerResponses" type="playerResponsesType"/>
    <xs:element name="interaction" type="interactionType"
minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="conversationsType">
  <xs:sequence>
    <xs:element name="conversation" type="conversationType"
minOccurs="1" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="interactionType">
  <xs:sequence>
    <xs:element name="source" type="gameEntityRefType"
minOccurs="1" maxOccurs="1"/>
    <xs:element name="destination" type="gameEntityRefType"
minOccurs="1" maxOccurs="1"/>
    <xs:element name="interactionType"
type="interactionTypeRefType" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="interactionPossibilityRefType">
  <xs:attribute name="ref" type="xs:string" use="required"/>
</xs:complexType>

```

```
<xs:element name="definitions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="gameEntities" type="gameEntitiesType"/>
      <xs:element name="interactionTypes"
type="interactionTypesType"/>
      <xs:element name="conversations" type="conversationsType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

# I. Zápisnice zo stretnutí tímového projektu

## I-1 Zápisnica z 1. stretnutia tímového projektu

### Tím č. 20 - Patronus

|                              |                   |   |
|------------------------------|-------------------|---|
| <b>Stretnutie č.:</b>        | 1                 | <b>Prítomní:</b><br>Bc. Daniela Hajdu<br>Bc. Lukáš Marták<br>Bc. Aleš Mäsiar<br>Bc. Lukáš Miškovský<br>Bc. Zora Moravčíková<br>Bc. Filip Šandor |
| <b>Vedúci tímu:</b>          | Ing. Eduard Kuric |   |
| <b>Dátum stretnutia:</b>     | 22.09. 2015       |   |
| <b>Čas stretnutia:</b>       | 17:00             |   |
| <b>Miesto stretnutia:</b>    | STU FIIT 3.28     |   |
| <b>Téma stretnutia:</b>      | text              |   |
| <b>Zápisnicu vypracoval:</b> | Bc. Filip Šandor  | <b>Ďalší zúčastnení:</b><br>doc. Mgr. Štefan Beňuš, PhD.  |

### Náplň stretnutia

Na stretnutí sa zúčastnil Štefan Beňuš, ktorý nám bližšie objasnil požiadavky, ktoré sú na projekt kladené. Priblížil nám hlavný účel hry (vykonávanie experimentov za účelom zberu dát pre výskum, ktorý sa zaoberá vplyvom spôsobu reči a jej tónu na dôveru človeka), a tiež to ako zabezpečiť aby hra tento účel splnila (zakomponovanie dvoch postáv, avatarov, ktorí budú hráčovi radiť, pričom by sa mali odlišovať iba spôsobom reči – jeden z nich sa hráčovej intonácií a tónu prispôsobuje, druhý nie). Následne členovia tímu položili otázky a prebehla diskusia. Vybrané body a návrhy z tejto diskusie:

- Návrh aby si hráč najprv vyskúšal hru cvične, a až po pár pokusoch by hral už naozaj, čiže ako súčasť experimentu. (Zora Moravčíková)
- Návrh aby hra bola generická a jediné čo by sa v rámci levelov menilo bolo prostredie a sada úloh, ktoré tam má hráč vykonať. (Eduard Kuric)
- Navrhnutie výmeny „hádzania kockou“, ktoré malo slúžiť na určenie počtu krokov hráča, za sadu mini hier. Účel oboch týchto mechanizmov je to, aby si hráč nikdy nebol úplne istý, či sa mu darí/nedarí vďaka poslušnutej rade avatara, náhode alebo jeho vlastnej šikovnosti. (Filip Šandor)
- Hovorila sa o možnosti multiplayer-a, no zhodnotili sme, že nemá zmysel sa tým v týchto skorých štádiách zaoberať. (Lukáš Marták)
- Podstatné je získať od hráča dlhé, rozvité vety, ktoré sú zároveň aj zrozumiteľné.
- Bolo by vhodné aby hráčovi nebolo umožnené v rámci hry klamať avatarom.
- Je nutné doriešiť prečo by mal hráč vlastne veriť informáciám a schopnostiam avatarov natoľko aby si ich rady aktívne vypočul.

- Z viacerých dôvodov (zaujímavosť, samotné testovanie) sme sa rozhodli, že bude lepšie hru rozdeliť do levelov.
- Zaujímavá by tiež mohla byť možnosť výmeny rolí hráča a avatara, ktorý mu radí. Poskytlo by to možnosť na získanie rozsiahlejšej vzorky hlasu hráča a jeho spôsobu reči.
- Dohodli sme sa na oficiálnom čase ďalších stretnutí tímového projektu, a to v pondelok o 8:00.

Z diskusie ďalej vyplynulo, že je nutné klásť dôraz na to aby:

- sa hráč rozhodoval, ktorého avatara poslúchne, len na základe ich rozdielnej intonácie a spôsobu reči,
- nebolo hráčovi jasné, či rady, ktoré mu avatari poskytujú sú dobré alebo zlé,

hra bola dostatočne zaujímavá a pútavá na to aby odlákala pozornosť hráča od faktu, že sa jedná o experiment.

### **Úlohy do ďalších stretnutí**

- Vymyslieť názov tímu
- Vytvoriť logo tímu
- Vytvoriť plagát
- Vytvoriť stránku
- Vymyslieť dej hry

## I-2 Zápisnica z 2. stretnutia tímového projektu

### Tím č. 20 - Patronus

|                              |                        |   |
|------------------------------|------------------------|---|
| <b>Stretnutie č.:</b>        | 2                      | <b>Prítomní:</b><br>Bc. Daniela Hajdu<br>Bc. Lukáš Marták<br>Bc. Aleš Mäsiar<br>Bc. Lukáš Miškovský<br>Bc. Zora Moravčíková<br>Bc. Filip Šandor |
| <b>Vedúci tímu:</b>          | Ing. Eduard Kuric      |   |
| <b>Dátum stretnutia:</b>     | 29.09. 2015            |   |
| <b>Čas stretnutia:</b>       | 17:00                  |   |
| <b>Miesto stretnutia:</b>    | STU FIIT 3.28          |   |
| <b>Téma stretnutia:</b>      | Organizácia tímu       |   |
| <b>Zápisnicu vypracoval:</b> | Bc. Lukáš<br>Miškovský |   |

### Vyhodnotenie zadaných úloh

Úspešne sme vymysleli názov aj logo tímu a vytvorili plagát. Na stránke tímového projektu sa ešte pracuje, pričom aktuálna verzia bola prezentovaná. Bolo by vhodné vymyslieť na stránku ešte nejaké výstižné logo. Na dejí a detailoch hry aj naďalej aktívne pracujeme.

### Náplň stretnutia

Preberali sme mnoho problémov súvisiacich s rôznymi aspektmi projektu, napríklad organizácia tímu:

- Ktoré z možných nástrojov použiť pri komunikácií, organizácií či kontrole: TFS, Flow, GitHub, atď.
- Rozhodli sme sa pre TFS, ktoré je možné nájsť na školskom servery, takže nie je nutné riešiť server vlastný.
- Na zdieľanie dokumentov a iných súborov sa bude používať OneDrive.
- Na komunikáciu v rámci tímu budeme používať Slack.
- Rozprávali sme sa o skupinovom maily – premenovanie pôvodnej skupiny by bolo zbytočne komplikované meniť, založíme teda novú skupinu, kam sa presmerujú všetky maily zo starej skupiny.
- Komunikáciu k prideleným úlohám budeme riešiť priamo v TFS, formou pripomienok. Archivovať sa ďalej budú iba kľúčové rozhodnutia a riešenia.

Jazyky, herný engine, atď.:

- Hra by mala byť platformovo nezávislá, zvažovali sme Unity.

- Uvažovali sme nad možnosťami jazykov, pričom aj keď máme všetci bohaté skúseností s jazykom Java, zhodli sme sa, že to asi nebude ideálne riešenie. Priklonili sme sa k C#.
- Možno by bolo dobré spraviť to ako webovú hru.
- Pre Javu by sa dala využiť SDL knižnica na vývoj hier, no primárne je pre C++.
- Mali by sme si nainštalovať plugin do Visual studia, ktorý slúži na sledovanie programovacieho procesu používateľa. Tieto informácie sú pre jeden z výskumov na fakulte.

Detaily a ohraničenia hry:

- Hráč sa ocitne na neznámom mieste, bude postupne prechádzať viacerými rozdielnymi oblasťami, avatari mu radia.
- Úspešnosť hráča nemôže závisieť od ich schopností. To či sa hráčovi darí musí závisieť len od toho, ktorú radu si vyberie.
- Je nutné dať hru dopredu otestovať vzorke hráčov aby sme zistili na základe čoho sa hráči rozhodujú, ktorú z rád si vybrať.
- Je otázne či bude hráčovi umožnené dostať radu od jedného alebo od oboch avatarov v rámci jedného rozhodnutia. Ak prichádzajú do úvahy obe možnosti bude to treba pridať do nastavení hry.
- Treba doriešiť ako rozlíšiť avatarov bez toho aby to ovplyvnilo rozhodovanie hráča. Možnosťou je namapovanie na myš (ľavé a pravé tlačidlo) no avatari by sa nemali zobrazovať na rôznych častiach obrazovky.
- Výber avatara (vypočutia jeho rady) oslovením – tlačidlom sa spustí mikrofón, hráč osloví avatara menom/frázou a ten mu poskytne radu. Bolo by potrebné buď vymyslieť neutrálne mená pre avatarov alebo nechať hráča samotného nech si ich pomenuje.

Rozdelenie tímových rolí:

- Vedúci tímu - Filip Šandor
- Dokumentarista - Aleš Mäsiar
- Manažér rizík - Lukáš Miškovský
- Manažér kvality a komunikácie, Zástupca vedúceho - Lukáš Marták
- Plánovač a manažér monitorovania procesu - Zora Moravčíková
- Manažér komunikácie a monitorovania - Daniela Hajdu

Budúci týždeň sa na stretnutí zúčastní aj zamestnanec SAV, ktorý nám poskytne bližšie údaje o tom v akom formáte nám budú poskytovať dáta.

### Úlohy do ďalších stretnutí

Aj keď sú úlohy rozdelené a konkrétny členovia tímu za ne nesú zodpovednosť, ostatní môžu tieto úlohy pripomenkovať a prispievať k ich riešeniu tiež.

Rozdelenie úloh:

- Daniela Hajdu – vytvoriť šablónu pre zápisnice a prepísať zápisnice zo stretnutí č. 1 a 2
- Lukáš Marták – vyhľadať a rozposlať vhodné materiály a tutoriály
- Aleš Mäsiar – založiť Slack a preskúmať možnosti komunikácie, ktoré poskytuje
- vypracovať štruktúru a CSS web stránky
- Lukáš Miškovský – rozbehnúť TFS a pridať úlohy
- Zora Moravčíková – grafika Web stránky
- Filip Šandor – založiť dokument slúžiaci na gameplay brainstorming (nápady na dej hry)

Spoločné úlohy:

- Naštudovať si naše roly
- Vymyslieť názvy šprintov
- Zistiť či commit message ide automaticky

## I-3 Zápisnica z 3. stretnutia tímového projektu

### Tím č. 20 - Patronus

|  |                         |   |
|--|-------------------------|---|
| <b>Stretnutie č.:</b>  | <b>3</b>                | <b>Prítomní:</b><br>Bc. Daniela Hajdu<br>Bc. Lukáš Marták<br>Bc. Aleš Mäsiar<br>Bc. Lukáš Miškovský<br>Bc. Zora Moravčíková<br>Bc. Filip Šandor |
| <b>Vedúci tímu:</b>  | Ing. Eduard Kuric       |   |
| <b>Dátum stretnutia:</b>   | 08.10. 2015             |   |
| <b>Čas stretnutia:</b>   | 9:00                    |   |
| <b>Miesto stretnutia:</b>  | STU FIIT 3.28           |   |
| <b>Téma stretnutia:</b><br>Zhodnotiť úlohy z predošlého stretnutia<br>Dohodnúť spoluprácu so SAV<br>Základný koncept hry |                         |   |
| <b>Zápisnicu vypracoval:</b>   | Bc. Zora<br>Moravčíková | <b>Ďalší zúčastnení:</b><br>doc. Mgr. Štefan Beňuš, PhD.<br>Ing. Marián Trnka   |

### Vyhodnotenie zadaných úloh

- Dokončenie stránky:
  - o výzor je urobený
  - o treba dorobiť galériu a dokumenty
  - o pozrieť ešte tie štýly dizajnu
- Vytvorenie spoločného TFS
  - o funguje, už len nahodiť pridelené tasky
  - o priradiť ich do iterácií
- Vymyslieť názvy šprintov
  - o budú to zaklínadlá
  - o urobiť zoznam nejakých 10
- Pripraviť vzory pre zápisnice
  - o hotovo
- Vytvoriť komunikačný Slack
  - o hotovo a funguje



## Náplň stretnutia

### Hlasový modul SAV

Už sa vyvíja modul

Modely sú hotové, treba si ujasniť komunikáciu

Pre rozpoznávač bude dobré aby doména bola limitovaná čo najviac, mini dialóg, ale nemusíme sa limitovať slovníkom, rádovo môže obsahovať tisícky slov

Rozpoznávač bude bežať na SAV serveri

Ak bude server na SAV, je možné obslúžiť 5 hier na raz? – treba na to myslieť, ale nemá to na to vplyv

Hráč hovorí a SAV si sleduje parametre reči a bude ich vracat' a my si to budeme LOGovať  
SAV rozporná zvuk, nameria parametre a tie pošle do hry, aby sme ich mohli uchovať

- výstup z hry, parametre reči akými on hovorí, akými SAV hovorila a ako zareagoval.

Začatie počúvania: gombík – teraz chcem hovoriť a odošle sa to na server

2 možnosti prispôsobenia reči:

Change by change prispôsobovanie – k requestu pridáme tie údaje späť o reči a len povieme, či chceme alebo nechceme prispôbiť, alebo taký čo pôjde úplne inak

Alebo na začiatku – na začiatku nejaké otázky, podobné aké sa budú používať v hre, nemusíme ísť na slová, nemusí to byť hra, ale len niečo prečítaj napr. návod

Dialógy po slovensky

Možno spraviť jazykovú mutáciu

### Komunikácia s hlasovým modulom

Bude to webová hra, preto by bolo dobrá REST služba

Minimálne dve RESTové služby od SAV

1. Pošleme zvuk
  - a. Dostaneme odpoveď s textom a parametrami reči, (intonácia – základný tón, rýchlosť reči – slabiky za sekundu, energia / hlasitosť)
2. Pošleme text
  - a. Posielame text s parametrami reči

Budeme si posielat' redundantné dáta, ale SAV nemusí riešiť nejaké session.

Jediný problém je v tom, že ak sa rozpoznáva počas hovoru, je to fajn, ale ak sa to bude postupne – odozva bude pomalšia : Bez kompresie 10s = 300kB

Detekcia pauzy - Oreže, pošle a hráč zatiaľ hovorí

Server musí byť pripravený keď pustíme hru, samotná inicializácia modulu trvá dlhší čas

Úvodná fáza rozpoznávača trvá dlho, samotné spracovanie môže byť aj 2s. – zabaviť ho na 2s, použiť rozmýšľanie – výskum, keď robot robil hmmm (hezitácie) boli lepšie výsledky ako keď nerobil hmm

Treba myslieť na konfiguráciu komunikácie pri hre, nastavenie portu a IP servera

### Vyvíjaná jednoduchá skúšobná hra

Vyvíjajú jednoduchú hru, kde sa pýtajú na karty (Komunikácia „čo mám urobiť?“)

V Argentíne nemajú taký dobrý rozpoznávač ako my na SAV.

Avatari: Tú sú dvaja, sú si len podobní, a hráč si pýta, ktorého chce počuť

Malá hra už je – na základe nej si môžeme postaviť komunikáciu

### Hra a radenie:

Že radil by iba jeden, a buď by ho počúval, alebo nie.

Potrebovali by sme veľa ľudí

Nemal by vedieť či vyhral alebo prehral na základe rady avatara.

Levely budú iné len v interface, ale rovnaká logika.

Dvaja avatari – dvojičky, musia mať podobný hlas, ale zároveň iný ?

Števo sa prikláňa k jednému, len treba veľa ľudí a veľa dát so začiatočným dotazníkom

Budú dvaja avatari, ktorý by hovorili stále a striedali by sa v tom, kto ide prvý

2 avatari: hovoria naraz, v princípe sa to dá urobiť tak, že ak by bol jeden, tak by tie dve možnosti museli byť zrejme a jeden avatar by radil jednu možnosť, keby boli dvaja tak

Scenár by mal byť stavaný tak, aby tie dve možnosti boli zrejme.

Mini hry nezavrhuje

Veľa materiálu potrebujeme, ale nie úplne veľa, aby nevznikli chyby. Malý mini dialóg, že ho možno začne aj avatar, bude sa pýtať a nedáva mu úplnú voľnosť. Takže opis miestnosti, kde očakávame niečo, ale donútime ho to povedať. Možno nejaké nápisy, piktogramy a na základe toho čo si všimol, si avatar prispôsobí odpoveď. Každý človek môže vidieť niečo iné a my vieme čo by mohol vidieť a na základe toho mu vygenerovať odpoveď.

Jazyk bude slovenský, ale ľahko preložiteľné do angličtiny

Dialógový manažér si bude viesť rozhovory a bude vedieť, kde s nachádza a bude mať bázu, musí to byť dostatočne limitované, aby sa veľa nemýlil. Konkrétne vedomosti ku konkrétnemu kroku.

Môže byť úlohou hráča, určiť, ktorý avatar je lepší – hráč sa bude na tieto hlasy sústrediť

### **Časové ohraňenie**

Po prvom semestri – koncom Januára, by už mal byť prototyp, s testovacou komunikáciou

Do 15.10. mať komunikáciu – tie Restové služby, to by sme mohli vedieť do dvoch týždňov

Zatiaľ by sme mohli vymyslieť jeden level / miestnosť

Minimálne mesiac na spracovanie textov a prispôbení. Do týždňa nejaký základ, aby sme mali s čím pracovať.

Mať jednu miestnosť a pripraviť čo najviac dialógov, aby sme mohli testovať komunikáciu

2 týždne na slušné premyslenie príbehu, jednej miestnosti, mini hier

### **Potrebné parametre na výskum**

Počet rozhodnutí od jedného človeka v rámci jednej hry:

- Rádovo 50 rozhodnutí (Go Fish hrali za sebou 3 x 15 rozhodnutí)
- Je dôležité aby sada otázok a odpovedí bola rovnaká? Počet by bol dobrý keby bol rovnaký, ale otázky a odpovede môžu byť rôzne
- Váhy odpovedí ? (Rozdiel medzi rozhodovaním dôverou a potom aj s tou váhou), budeme mať 20 rozhodnutí pritom 10 bude vážnejších, v podstate stačia dve váhy, ako faktor analýzy. Argentínčan mal taký prístup, aby na konci prišlo to rozhodnutie či zomriem alebo nie a tomu dám prednosť, ako data point. V rámci mini hier by malo byť kasíno a získal body a dostal by aj peniažky kde určíme minimum a maximum eur odmenu, a urobiť to tak, aby nevidel hneď výsledok – nemôže špekulovať, a nemôžem jedného obviňovať
- Dávkach 30-45 minút aj s dotazníkom
- Neprenášať dôsledky rozhodnutí medzi levelmi

•

### **ZHRNUTIE:**

Postaviť hru na princípe dvoch avatarov, tak že sa dá používať len 1

Stanley parable - <http://www.stanleyparable.com/>

Keď vytvoríme dialógový model treba ho čo najskôr poskytnúť SAV, a pridať podľa testov aké slová, prípadne 10000 najčastejších slovenských slov

Posunúť ďalej príbeh s miestnosťou

## Úlohy do ďalších stretnutí

- Filip
  - Analýza hry Stanley parable
  - analýza problému, prečo je to zaujímavé
- Lukáš Ma.
  - Navrhnutie architektúry klientskej časti
  - pripraviť UML modely
  - Identifikovať moduly, ktoré budeme potrebovať – diagram komponentov
    - Dialógový manažér s modelom scény
    - Prezentačná vrstva oddelená od logiky
    - Oddelenie medzi levelmi
      - Neprenášajme údaje z predchádzajúcich levelov
    - Manažér logov
- Zora
  - vyskúšať unity web player
  - zistiť obmedzenia pre webovú hru
- Daniela
  - Založiť dokument s ohraničeniami a možnosťami
  - Spísať veci, pri ktorých sme sa ešte nerozhodli a nedohodli
- Lukáš a Aleš
  - intenzívne sa venovať inšpirácií a príbehu hry
  - dôvody prečo to tí ľudia hrajú
  - zamyslieť sa aj nad iným štýlom hry
    - stavačky, skákačky, akčné ?
      - zhodnotiť závislosť od hráčskych skúseností
- Všetci
  - vytvoriť nejakú miestnosť
  - oboznámiť sa s Unity 3D
  - Stretnutie utorok skôr o 15tej

GRAFIKA: Treba nám zjednotiť štýl v celej hre a tak ho potom používať  
Zapnúť verziovanie v dokumentoch  
Spraviť plán šprintov – dvojtýždňové šprinty  
Zbieranie dát bude len v rámci experimentu na nejaký vonkajší server

## I-4 Zápisnica z 4. stretnutia tímového projektu

### Tím č. 20 - Patronus

|                              |                                  |                  |                      |
|------------------------------|----------------------------------|------------------|----------------------|
| <b>Stretnutie č.:</b>        | 4                                | <b>Prítomní:</b> | Bc. Daniela Hajdu    |
| <b>Vedúci tímu:</b>          | Ing. Eduard Kuric                |                  | Bc. Lukáš Marták     |
| <b>Dátum stretnutia:</b>     | 13.10. 2015                      |                  | Bc. Aleš Mäsiar      |
| <b>Čas stretnutia:</b>       | 15:30                            |                  | Bc. Lukáš Miškovský  |
| <b>Miesto stretnutia:</b>    | STU FIIT FabLab                  |                  | Bc. Zora Moravčíková |
| <b>Téma stretnutia:</b>      | Architektúra<br>systému a príbeh |                  | Bc. Filip Šandor     |
| <b>Zápisnicu vypracoval:</b> | Bc. Daniela Hajdu                |                  |                      |

### Vyhodnotenie zadaných úloh

Analýza hry Stanley Parable – niekoľko členov tímu si hru zahrlo, no primárne robil analýzu Filip Šandor. Zaujímavá bola hra najmä kvôli vtípnemu rozprávačovi, tiež obsahovala 19 rôznych koncov. Rozprávač dáva Stanlymu rady, no rozpráva v minulom čase, ako keby rozprával príbeh (Stanley urobil toto...), je na hráčovi či ho bude počúvať. V hre sa tiež vyskytujú náhle zmeny lokalít (vesmír) či grafiky (rozprávač povie, že tu nie sú dokončené textúry a zrazu to vyzerá ako Minecraft). Možno by bolo zaujímavé aj našich avatarov riešiť týmto spôsobom, že by akoby rozprávali príbeh čo sa odohral v minulosti.

Architektúra systému – Lukáš Marták vytvoril diagram komponentov. Mali by sme síce začínať use case-ami a nie architektúrou, no tie sa nedajú robiť bez scenára, ktorý ešte nemáme. Predstavenie jednotlivých častí architektúry:

- unity engine – prezentačná vrstva, obsahuje objekty a ich vlastnosti, slúži na vykresľovanie a tvorí interface pre C#.
- game engine – obsahuje skripty a log manager čiže ukladá dáta, taktiež obsahuje bázu znalostí (informácie o tom kde sa hráč nachádza)
- dialog manager – prepojený s produkčným systémom, tu sme sa bavili o možnosti istého napovedajúceho systému. Čiže keď hráč prejde ponad istý predmet myšou vypíše sa mu názov daného predmetu, na základe toho by sa hráč mohol spýtať na tento predmet jeho konkrétnym menom, tým by sme získali key words, podľa ktorých môžeme jednoduchšie identifikovať čo hráč povedal. Treba toto rozpoznávanie vytvoriť tak, aby ani neúplná informácia (rozoznávanie reči, ktoré zabezpečuje SAV nie je 100%) nespôsobila problém. Toto sa dá čiastočne riešiť aj defaultnými odpoveďami. Čiže ak hráč vo vete použije len jedno key word, bude iná defaultná odpoveď ako keď ich použije dve.

- chýba nám know-how, potrebujeme model
- ak budeme hru sprístupňovať veľkému počtu používateľov bude potrebné ju odpojiť od SAV. Vstup by bol písaním či výber niekoľkých možností viet (dialógových možností)

Zorka spravila malú webovú hru v Unity – môže to byť programované v C# a Unity si je samo schopné vytvoriť z neho java script pri exporte. Hra bude fungovať na všetkých platformách.

Názvy šprintov vybrala Zorka – budú to názvy kúziel z Harry Potter série.

Ohraničenia hry (ohľadom avatarov a kladených požiadaviek) spísala Daniela.

### **Náplň stretnutia**

Zhrnuli sme čo sme vymysleli k príbehu – dva varianty:

1. Variant s mimozemšťanmi.
2. Externista, elektrikár, ktorý prišiel niečo opraviť.

Debata k druhému variantu:

- máme zabudnúť, že avatari sú dvaja, je to len kozmetická záležitosť
- je otázne či netreba trochu viac rozviesť príbeh - bude človeka vkuse baviť len utekať z budovy?
- možno by bolo fajn zmeniť postavu, a s ňou aj cieľ. Napríklad príde poslíček s pizzou, pokiaľ by boli po ceste nejaké body, ktoré by bolo treba zbierať mohli by byť reprezentované rôzne pre rôzne postavy. Tieto rôzne postavy by sa mohli nakoniec stretnúť pri vyvrcholení príbehu.
- ešte sme si nevybrali budovu, v ktorej sa celý príbeh bude odohrávať, mohol by to byť napr. výskumný ústav alebo kancelárie
- postavy by mohli byť dostatočne generické na to aby si ich hráč mohol personalizovať a prispôbiť
- bude treba vymyslieť postavy (vybrať zamestnania), ktoré hráčovi poskytneme na výber
- hráč si bude môcť vybrať vlastnosti postavy a od toho sa budú odvíjať úlohy (poslíček s pizzou hľadá zákazníka, ktorému má doručiť pizzu) a poprípade sa bude prispôsobovať scéna
- hráč sa nemôže vracieť do miestností, ktorými už prešiel a skúšať ďalšie možnosti
- treba vymyslieť akým spôsobom sa budú striedať postavy, prečo, kto presne to bude, či sa stretnú počas hry alebo nie (mohli by jeden druhého zahliadnuť v miestnosti s kamerovým systémom)
- môžeme sťahovať objekty a textúry ale treba si dávať pozor aby hra vyzerala konzistentne takže treba nájsť nejaký vhodný zdroj objektov, ideálne od jedného autora
- minihry ešte nemáme doriešené ale mohli by predstavovať prekážku, cez ktorú sa hráč musí dostať aby vstúpil do ďalšej miestnosti (pre postavu elektrikára sa môžeme

inšpirovať hrou Hook), alebo by mohli slúžiť na to aby sa hráč dozvedel niečo viac o deji, poprípade získal nejaký špeciálny predmet alebo schopnosť

- je nutné vymyslieť prečo sú všetci v tej budove a nejaké odhalenie na záver
- nápad, že by hra zároveň mohla psychologicky otestovať hráča samotného na základe toho ako sa rozhodol plniť/neplniť úlohy jednotlivých postáv
- možno by sa do minihier dal zakomponovať health bar alebo achievements

## Úlohy do ďalších stretnutí

Spoločné úlohy:

- vymyslieť príbeh a založiť dokument, do ktorého sa bude brainstormovať
- napísať scenár aspoň pre jeden level či postavu, vytvoriť dialógy pre oboch avatarov (konverzácia musí dávať zmysel aj keď prehovorí iba jeden z avatarov), scenár je špecifický pre jednotlivé miestnosti, obsahuje aj opis scén. Aj keď je úloha pre všetkých Filip ju koriguje.

Pridelené úlohy:

- Daniela Hajdu a Lukáš Marták – analýza produkčných systémov a možností pre dialógy, zistiť či je k dispozícii open source

Nepridelené úlohy:

- Získať bližšie informácie a napísať draft prihlášky do TP Cup

## I-5 Zápisnica z 5. stretnutia tímového projektu

### Tím č. 20 - Patronus

|                              |                    |   |
|------------------------------|--------------------|---|
| <b>Stretnutie č.:</b>        | <b>5</b>           | <b>Prítomní:</b><br>Bc. Daniela Hajdu<br>Bc. Lukáš Marták<br>Bc. Aleš Mäsiar<br>Bc. Lukáš Miškovský<br>Bc. Zora Moravčíková<br>Bc. Filip Šandor |
| <b>Vedúci tímu:</b>          | Ing. Eduard Kuric  |   |
| <b>Dátum stretnutia:</b>     | 20. a 22. 10. 2015 |   |
| <b>Čas stretnutia:</b>       | 9:00               |   |
| <b>Miesto stretnutia:</b>    | STU FIIT 3.28      |   |
| <b>Téma stretnutia:</b>      |                    |   |
| <b>Zápisnicu vypracoval:</b> | Bc. Aleš Mäsiar    | <b>Ďalší zúčastnení:</b>  |

### Vyhodnotenie zadaných úloh

#### Náplň stretnutia

Diskusia o novom návrhu:

- Rozhodnutie o spracovaní cudzieho (existujúceho) príbehu
- Treba myslieť na copyright – kopírovať iba voľnejšie (nekopírovať mená a presný príbeh)
- Ostávajú hlavolamy a útek z miestnosti
- Prelínanie reálneho príbehu s novým – ako bude riešené (napr. prechody medzi scénami)
- Filip – urobiť rozhodnutie iba tak, aby zmenilo neskôr iba menší detail (napríklad farbu oblečenia)
- Edo – pôvodná myšlienka: príbeh rozsekaný na menšie segmenty – ak by hráč išiel príliš dlho podľa pôvodného príbehu, tak by sme ho vrátili späť
- Časové okno – ľudia sú v nejakej miestnosti, ale hlavná postava ich nevidí (môže posunúť notebook a potom, keď bude hrať za inú osobu, ktorá napríklad vlastní ten notebook, tak by ho nevedel nájsť)
- Určiť si, čo je cieľom hráča – napríklad nájsť nejaký mýtický predmet
- Príšery môžu ohroziť iné postavy, ale nie hráča
- Ak by sa hráčovi podarilo zachrániť postavu, tak si s ňou môže napríklad pokecať, ale ak by ju nezachránil, tak by ju tam nemal
- Príbeh môže mať viacero ciest
- Treba vymyslieť alternatívne mená pre postavy
- Zamyslieť sa nad kamarátmi, ktorých by mohla hlavná postava mať, čo by mohli robiť (rozprávať vtipy, ručať – troll, kamarátka, ktorá dáva hádanky a za uhádnutie dostane hráč odmenu, ...)



- Vychytávky – záškodnícka mapa, alebo tajná miestnosť, kde môže pozerat' youtube videá, čarovné zrkadlo (kde uvidí nejakú blbosť)
- Dano Drevo
- Minihry:
  - dostaneš nejakú mincu za achievement a môžeš ju využiť na zahratie si minihry v automate
  - v Metlobale pozbierajú mince do automatu
  - vymyslieť spôsoby, ako pozbierať mince
  - rozhodnutie, načo budú slúžiť a či budú potrebné k prejdeniu príbehu
  - Napríklad Flappy Harry
  - Rozdeliť veci na useless a useful
  - musím byť v miestnosti, aby som si mohol zahrať minihru
  - V minihre získam mince, ktoré potom využijem na kúpenie dočasných vychytávok (zrýchlenie, vyššie skoky, zmenšenie)
  - Hádanky odomykajú nové minihry a zároveň zvyšujú množstvo zarobených peňazí v minihre
- Zahrnúť zaklínadlá – mohol by ich povedať, alebo zobrazíť výber pravým tlačidlom myši
- Hráč by prešiel najprv zopár hodinami, kde by sa naučil zaklínadlá
- Netreba vymýšľať veci, ktoré vyžadujú príliš veľké úsilie
- Analýza, čo je cieľom Harryho Pottera
- Jednoznačne identifikovať prerekvizity potrebné k nájdeniu Kameňa mudrcov
- Rozhodnutia avatarov budú v miestnostiach (ako vyriešil hlavolam v miestnosti), nie samotný výber dejovej línie (prípadne určiť rozhodnutiam prioritu)
- Zaujímavý element v hre je čas – Hermione nemá čas dať radu, lebo sa musí učiť a hráč musí získať nejaký item, ktorý by zabezpečil, aby mu tú radu dala
- Otravní ľudia – niekto bude hráča stále otravovať (že má pokazený záchod) a on mu bude môcť pomôcť až vtedy, keď niečo získa
- V každej scéne (leveli) ho môže otravovať niekto iný
- Čo budú hlavné fičúry
  - Jasný cieľ (získať dumbledorove boxerky)
  - Míľníky, v ktorých sa bude hráč rozhodovať
  - Minihry
  - Vtipy
  - Hádanky
- Zamyslieť sa nad rôznymi typmi itemov, ktoré môžeme zbierať (čokolády na vtipy, mince na čiapku, ...) alebo mať iba mince a rôzne ceny za rôzne veci
- Ako bude hráč riešiť hádanky (cez komunikačný modul, možnosti a/b/c) – rozhodnúť sa, ale musí to byť konzistentné a dostatočne generické
- Rozhodnúť sa, či chceme 2D alebo 3D
- Postavy by mali byť zaujímavé a vtipne zobrazené – nájsť si štýl
- Niekto by mal kontrolovať konzistenciu postáv a animácie (identitu)
- Diskusia o prvej scéne
- Avatari sa môžu meniť – problém s hlasmi (viac postáv by malo rovnaký hlas)
- Niekto anonymný posielal radu po sovách

- Určiť si dôležité fragmenty príbehu, ktoré je možné opísať jednou vetou
- Určiť si kľúčové postavy
- 

### **Úlohy do ďalších stretnutí**

- Vymyslieť alternatívne mená pre postavy, objekty
- Vymyslieť minihry a spôsoby, ako pozbierať mince
- Jednoznačne identifikovať prerekvizity potrebné k nájdeniu Kameňa mudrcov
- Pohľadať Harry Potter vtipy
- Zmapovať 2D hry
- Scenár (core)

## I-6 Zápisnica z 6. stretnutia tímového projektu

### Tím č. 20 - Patronus

|                              |                   |                          |                      |
|------------------------------|-------------------|--------------------------|----------------------|
| <b>Stretnutie č.:</b>        | <b>6</b>          | <b>Prítomní:</b>         | Bc. Daniela Hajdu    |
| <b>Vedúci tímu:</b>          | Ing. Eduard Kuric |                          | Bc. Lukáš Marták     |
| <b>Dátum stretnutia:</b>     | 26.10. 2015       |                          | Bc. Aleš Mäsiar      |
| <b>Čas stretnutia:</b>       | 8:30              |                          | Bc. Lukáš Miškovský  |
| <b>Miesto stretnutia:</b>    | STU FIIT 3.28     |                          | Bc. Zora Moravčíková |
| <b>Téma stretnutia:</b>      |                   |                          | Bc. Filip Šandor     |
| <b>Zápisnicu vypracoval:</b> | Bc. Lukáš Marták  | <b>Ďalší zúčastnení:</b> |                      |

### Vyhodnotenie zadaných úloh

Diskusia ohľadom aktuálneho stavu scenára, príbehu a návrhu prvotných scén:

Stav návrhu príbehu/scenára:

- Ihriskové intro ani vykresľovanie Vladovho príbehu ako sme sa bavili naposledy tam teda nebude.
- Budú tam 2 svety: svet čarodejníkov a nečarodejníkov. V bezprostrednom okolí čarodejníkov nefunguje elektronika vďaka ich magickej aure či čo. Medzi svetmi vládne sociálne napätie kde čarodejníci sú sociálni vyvrheli.
- V intre bude načrtnuté že ako nečarodejníci šikanujú čarodejníkov a čarodejníci sa chytajú za hlavu lebo im nejde wifi/počítač ani nič. Potom im dôjde nejaká pozvánka do školy kde im už bude lepšie lebo budú viac medzi svojimi.
- Začína sa to príchodom na strednú školu, teda mladí čarodejníci, čo práve prídu do školy (Jožo a spol., Vlodo) majú okolo 15 rokov. Hugo dovezie decká do školy.
- Prvá scéna - nováčikovia sa idú ubytovať a ako idú po chodbe, preletia sovy (Avatari) a Jožo spadne, lebo sa ich zlakne, asi mu vletia do hlavy alebo čo a vtedy začne počuť ich hlasy. Zatiaľ čo sa mu Avatari predstavujú, jeho spolužiaci rovesníci spolu s Hugom ako kulisa odchádzajú sa ubytovať a on zaostane. Keď dokecajú skupina s Hugom už je preč a Jožo sa ide ubytovať sám.
- Druhá scéna - Cestou stretne vtipkárku a hádankára. Kým ho hra pustí ďalej, musí si vypočuť 1 vtip a uhádnuť 1 hádanku, za ktorú dostane disketu na ktorej bude minihra FlappyHarry alias JožoNaMetle.
- Tretia scéna - Ďalej príde na koniec chodby kde sú hovoriace dvere, ktoré ho nepustia na intrák/ubytovňu/do klubovne kým nepovie heslo. Avatari radia Jožovi každý do iných dverí (jeden doľava, druhý doprava) avšak čo hráč nevie, za oboma bude tá istá miestnosť (keď vojde do jednej, dvere do druhej zmiznú).

- Štvrtá scéna - V miestnosti ktorá pripomína sklad upratovačky je automat kde sa dajú hrať minihry. Avatari poradia hráčovi aby dal disketu do automatu a vyhral v minihre peniaze na svietnik ktorým si posvieti na steny v miestnosti a nájde heslo. Jeden avatar poradí napr. Modré svetlo a druhý fialové svetlo, každý svietnik zobrazí iné slovo, ale v oboch prípadoch to bude správne heslo. Keď si hráč zahrá minihru a kúpi svetielko, Avatari mu poradia aby si s ním zasvietil niekde pod stolík na stenu kde keď príde a zasvieti uvidí heslo. Vtedy hra nenápadne naznačí že hráč už môže odísť a ísť sa ubytovať (napr, otvoria sa dvere).
- Piata scéna - Jožo si cez tretiu scénu (koniec chodby a hovoriace dvere) prejde do tejto scény čo je vnútro ubytovne / klubovne. Tu stretne Vlada ktorý naňho vrhne svoj arogantný opovrhlivý pohľad a odtanuje moonwalkom za nenápadného náznaku basovej melódie z Billy Jean od Michalea Jacksona niekam preč :D (toto možno ešte vypustíme lebo s tým nie sme tímovo stotožnení).
- Na škole sa pracuje na projekte, kde profesori vyvíjajú zariadenie (rušička) ktoré má umožniť čarodejníkom používať techniku. Jozef sa do školy prišiel v prvom rade učiť, no neskôr sa s kamarátmi o tom dozvedia na prvej hodine rušenia - základy rušenia - tam sa učia rušiť tú zlú auru čo im znemožňuje používať elektroniku. Napr. Sa pokúšajú rozsvietiť nejakú mini elektrickú žiarovku alebo tak.
- Zlo sa ukáže v druhej kapitole. Vladio sa vyskytne v škole ako nový študent ale ich rovesník. Jožo ho stretne a zistí že je hipster a že je divný, že by mu nemal veriť, lebo sa sním nechce kamarátiť.

Pripomienka ku skiciam a grafickým návrhom scén:

- Scény by bolo dobré kresliť do formy komixu, ku každej scéne 1 obrázok čo ilustruje čo sa tam nachádza a čo sa tam deje, načrtnuté objekty ktoré tam budú + krátky popis deja, dialógy.

## Náplň stretnutia

Agile - Scrum:

- Vytvorenie 2. šprintu - Alohomora
- Vytvorenie backlogu do Alohomory v TFS:
- User story:
  - Viem sa pohybovať v nejakej scéne.
- User story:
  - Viem komunikovať s Avatarom. Vie mi poradiť keď sa opýtam.
- User story:
  - Vedieť interagovať s abstraktnou osobou - funkcionality dialógu.
- User story:
  - Interakcia s hádankárkou - dostanem od nej hádanku a ja ju viem zodpovedať (uhádnuť) a indíciu môžem dostať niekde v prostredí.
  - Ak uhádnem, dostanem od nej disketu do inventára.
  - Ak neuhádnem, nevádi, môžem hádať donekonečna, proces hádania neskončí kým neuhádnem. Môžem ju stretnúť aj v iných scénach a odpovedať ak som medzičasom na niečo prišiel.
- User story:

- Interakcia s vtipkárkou - 1. krát povie vtip. Druhý krát ešte možno tiež, čím menej nových vtipov jej bude zostávať, tým viac si bude pýtať čokolády (energiu) "teraz už ti nepoviem lebo som unavená, ale dala by som si čokoládu", prípadne začne rásť pravdepodobnosť že ma odbije že nech jej dám pokoj už. Čokoládu a iné itemy by mali mať generické použitie, aby nemalo zmysel mať 1 item len na 1 vec v celej hre.
- Vytvorenie backlogu ďalších šprintov:
  - K rozbehaniu Dialog Managera
  - K spojeniu so SAV
  - K písaniu a nahrávaniu dialógov etc..

## Úlohy do ďalších stretnutí

Daniela:

- Nakresliť 1 scénu (3h)
- Nakresliť postavu Jozefa (2h)

Filip:

- Model postavy z nákresov (3h)
- Nakresliť scénu (4h)

Zorka:

- Vytvoriť projekt a scénu v Unity (5h)

Aleš:

- Ovládanie a animovanie pohybu postavy po scéne - aby sa okrem prehrávania animácie pohybu aj samotný objekt po scéne hýbal ako collider (6h)

Lukáš Mi

- Animácia pohybu postavy (5h)

Lukáš Ma

- Vymyslieť abstraktnú reprezentáciu scény na úrovni kódu (iniciálne súradnice postavy na scénach, vytvoriť objekty scény, postava, nejako reprezentovať mapu scén, nejaký graf a ako sa dá medzi nimi pohybovať alebo čo) (6h)

## I-7 Zápisnica z 7. stretnutia tímového projektu

### Tím č. 20 - Patronus

|                              |                                      |   |
|------------------------------|--------------------------------------|---|
| <b>Stretnutie č.:</b>        | 7                                    | <b>Prítomní:</b><br>Bc. Daniela Hajdu<br>Bc. Lukáš Marták<br>Bc. Aleš Mäsiar<br>Bc. Lukáš Miškovský<br>Bc. Zora Moravčíková<br>Bc. Filip Šandor |
| <b>Vedúci tímu:</b>          | Ing. Eduard Kuric                    |   |
| <b>Dátum stretnutia:</b>     | 03.11. 2015                          |   |
| <b>Čas stretnutia:</b>       | 17:00                                |   |
| <b>Miesto stretnutia:</b>    | STU FIIT 3.28                        |   |
| <b>Téma stretnutia:</b>      | Grafika a animácia hry, User Stories |   |
| <b>Zápisnicu vypracoval:</b> | Bc. Daniela Hajdu                    | <b>Ďalší zúčastnení:</b><br>doc. Mgr. Štefan Beňuš, PhD.  |

#### Vyhodnotenie zadaných úloh

Daniela

- scéna aj postava Jozefa boli nakreslené

Filip

- na prekreslení scény do digitálnej podoby sa ešte pracuje, ale vytvorených je niekoľko alternatív

- model Jozefa vytvorený bol, no treba na ňom popracovať

Zorka

- projekt bol vytvorený, no čo sa source control týka tak sa vyskytujú problémy s Unity, keď niekto preberá prácu po druhom členovi tímu nastávajú problémy s mapovaním

Aleš

- pohyb postavy po scéne funguje

Lukáš Miškovský

- pohyb postavy bol naanimovaný, ešte treba doladiť (prispôsobiť neskôr novému modelu)

Lukáš Marták

- na abstraktnej reprezentácii scény sa pracuje aj naďalej

#### Náplň stretnutia

Stretnutie začalo telefonátom cez Skype - bližšie sme Štefanovi objasnili príbeh a nasledovala debata.

- je otázne či chceme aby hráč s určitosťou vedel, že rady avatarov môžu byť horšie a lepšie, lepšie by bolo keby si tým nemohol byť istý (nedovolíme mu preskúmať druhú možnosť)
- pokiaľ sú rady horšie a lepšie sú dva prístupy, a to: rady avatarov sú aj dobré aj zlé, no sú distribuované rovnako (avatari majú rovnaký počet dobrých a zlých rád) alebo aj po počúvnutí rady si hráč nemôže byť istý, či by výsledok nebol lepší keby poslúchol druhého avatara (rady nie sú binárne)
- dialóg avatara s hráčom je iniciovaný aj avatarom ale môže byť aj hráčom samotným
- uprednostnili by sme keby neboli úplne zlé rady vôbec, keďže hráč by sa potom rozhodoval na základe výsledku a nie hlasu avatara, no zároveň by si hráč mal myslieť, že niektoré rady sú horšie ako iné, mal by pochybovať (na navodení tohto pocitu hráčovi treba popracovať)
- pôvodne bolo naplánované, že sa hráč na záver dozvie, že avatari, ktorí sa zjavujú vo forme sov mu pomáhajú komunikovať s jeho stratenými rodičmi – je však otázne či je takýto koniec vhodný keďže hráč by si mal vybrať avatara, ktorému verí viac
- na záver, po tom čo účastník experimentu dohral hru bude mu predložený dotazník, kde by sa mal vyjadriť, ktorému z avatarov veril viac/ktorí si myslí, že mu radil lepšie
- možno by bolo dobré keby hráč dostal tento dotazník ešte pred vyvrcholením hry. Takto by neprekážalo keby sovy predstavovali komunikáciu s jeho rodičmi, pretože by sa to dozvedel až po zodpovedaní otázok v dotazníku, a tak by to neovplyvnilo jeho rozhodnutie
- do hry bude zakomponovaný ekonomický systém - hráč získava v minihrách mince, za ktoré si následne môže kúpiť vylepšenia pre svoju postavu, no ešte nie je isté či sa zozbierané mince budú premietat' aj do reálnych peňazí, ktorými bude hráč odmenený za účasť na experimente. V takomto prípade by to totiž mohlo mať efekt na to koľko mincií hráč minie (keby mal na základe nich dostať finančnú odmenu mohol by sa rozhodnúť, že ich nebude míňať ale šetriť), a tiež by tak mohol stráviť viac času hraním minihier a zbieraním mincií ako prechádzaním príbehu
- hráč možno bude finančne odmenený aj za to, že sa dostal do cieľa, a teda prešiel hru celú
- prvý prototyp by mal byť hotový začiatkom budúceho roka a už by mal, na základnej úrovni, komunikovať zo SAV

Po telefonáte so Štefanom:

- predviedol sa model Jozefa (hlavnej postavy) - je nutné ho ešte prerobiť
- riešila sa kamera - či bude statická alebo sa bude pohybovať spolu s hráčom - zhodli sme sa, že by asi bolo lepšie keby hráča nasledovala
- bavili sme sa o tom ako robíme SCRUM, treba na tom trocha popracovať - nerobili sme retrospektívu, potrebujeme napísať metodiky, a tiež nemáme ešte spravený backlog

- treba doriešiť ako sa budú reprezentovať informácie o jednotlivých scénach v rámci Unity, napr. graf kde sú objekty obsiahnuté v scéne a pod nimi v grafe ich features

Ďalej sme identifikovali User Stories. Počas tohto proces sme prebrali viacero vecí (väčšina výsledkov tejto debaty je zapísaná v TFS pod jednotlivými User Stories, ktorých sa problém týkal):

- použitie predmetov – zmena kurzoru na práve používaný objekt (napr. baterka), tento predmet bude teda namapovaný na kurzor takže pri tej baterke by sa svietilo na tie časti scény kam ukazuje hráš myšou
- obchod by nemal byť len statická scéna ale mal by obsahovať predmety, s ktorými je možné interagovať

### Úlohy do ďalších stretnutí

- Prekresliť scénu do digitálnej podoby - Filip
- Vytvoriť model postavy podľa náčrtu - Filip, Daniela
- Vytvoriť animáciu pohybu postavy - Lukáš Miškovský
- Vymyslieť abstrakciu reprezentáciu scény - Zorka, Lukáš Marták
- Vyladiť funkčnosť source control - Zorka, Lukáš Marták
- Zaškoliť tím do RealDraw - Filip
- Testovanie v rámci Unity - Aleš



## I-8 Zápisnica z 8. stretnutia tímového projektu

### Tím č. 20 - Patronus

|                              |                     |  |
|------------------------------|---------------------|--|
| <b>Stretnutie č.:</b>        | <b>8</b>            | <b>Prítomní:</b> Bc. Daniela Hajdu<br>Bc. Lukáš Marták<br>Bc. Aleš Mäsiar<br>Bc. Lukáš Miškovský<br>Bc. Zora Moravčíková<br>Bc. Filip Šandor |
| <b>Vedúci tímu:</b>          | Ing. Eduard Kuric   |  |
| <b>Dátum stretnutia:</b>     | 10.11. 2015         |  |
| <b>Čas stretnutia:</b>       | 17:00               |  |
| <b>Miesto stretnutia:</b>    | STU FIIT 3.21       |  |
| <b>Téma stretnutia:</b>      |                     |  |
| <b>Zápisnicu vypracoval:</b> | Bc. Lukáš Miškovský | <b>Ďalší zúčastnení:</b><br>prof. Ing. Mária Bieliková, PhD.   |

### Vyhodnotenie zadaných úloh

Testovanie v unity – treba vypracovať metodiku na vytváranie testov. Dá sa robiť aj continuous integration. (vzhľadom na povahu nášho projektu to však nie je veľmi prínosné). TFSko to možno nezvládne.

Treba dotvoriť 2D Jozefa

Animáciu takisto

Aktívne úlohy – vymyslieť abstraktnú reprezentáciu scény,

Vymyslieť abstraktnú reprezentáciu scény – samael nechal 2hod. otvorene na reálne otestovanie na kóde

### Náplň stretnutia

TFS nevie sortovať podľa stĺpcov

Efekty z realdrawu neaplikovať na celý objekt (vybrať si časť, napr. bundu). Dohodnúť sa na jednom filtri a používať ho všade.

Puppet2D – spísať čo majú spĺňať skice na dobré animovanie

Natočiť si videá na animovanie. Aby sme vedeli od čoho ísť a ako tvoriť pózy.

Scéna pre nás == scéna v unity == jeden sprite.

Stav scény meniť

Teleport do miestnosti s Hadankarkou, automatom atd.

Znovupoužitelnosť scén – mať viacero vecí ktoré sa randomne zobrazia. Zmazať/pridať predmety za behu. Pamätáme si len aktuálnu scénu.

Metodika zapísaná v scena-opis.

V scene by mal byť zoznam podmienok. Objekt ma zapísaný stav - v tom stave čo ten objekt dá aby sa naplnila tá a tá podmienka. Nepísať len odrážky v dokumente ale skúsiť aj reálny príklad. Scény, objekty, stavy načrtnúť.

Interakciou sa vymenia nejaké info. Naviazaná vec na túto interakciu a má definované že po x krokoch sa zmení to a to.

---

Zhrnutie gameplay-u by filip pre Profesorku.

Minimalizovať odlišnosti medzi ľuďmi sa nedá. Človek si z podanej informácie zapamätá vždy niečo iné.

Odkomunikovať čo v hre bude a nebude - dať ale dopredu vedieť čo je nad naše kapacity.

---

Treba zdefinovať jednotlivé časti interakcie a z čoho sa skladá. Sled, vedieť zistiť koniec interakcie aby sme to vedeli naviazať na zmenu v hre.

Objekty reprezentujúce text – výmena. (Jozef dostane input – klik na objekt – objekt mu posle „som to a to“, „som zobratelný“ a pod. a on to vie transformovať do akcie čo spraví v hre. Nema v sebe info Jozef, to mu davaju veci)

Musíme vedieť byť schopný vyskladať si hru zo scén a tie v sebe majú objekty so stavmi.

Príklad: Jozef a zamknuté dvere.

Do dokumentu odôvodniť prečo sme v prvom šprinte nerobili retrospektívu.

Spravenie metodiky na metodiku.

### **Úlohy do ďalších stretnutí**

Doplniť descriptions ku taskom – ku všetkým taskom

Poprehadzovať user stories do všeobecnej cesty z prvého šprintu

Spraviť metodiku na testovanie. (čo, ako, kde treba)

Testovanie v unity –

Integration testovanie (riešiť neskôr, ďalší semester?)

Task – reporty v TFS-ku. Kto koľko cez týžden robil, ako to exportnúť?

Metodika na umiestňovanie vo OneDrive.

### **Retrospektíva:**

Podarila sa prvá reprezentácia Jozefa.

Máme v podstate zdefinovaný štýl.

Máme vymyslený rámec hry.

Naučili sme sa robiť s realdrawom.

## I-9 Zápisnica z 9. stretnutia tímového projektu

### Tím č. 20 - Patronus

|                              |                      |                          |                      |
|------------------------------|----------------------|--------------------------|----------------------|
| <b>Stretnutie č.:</b>        | <b>9</b>             | <b>Prítomní:</b>         | Bc. Daniela Hajdu    |
| <b>Vedúci tímu:</b>          | Ing. Eduard Kuric    |                          | Bc. Lukáš Marták     |
| <b>Dátum stretnutia:</b>     | 20.11. 2015          |                          | Bc. Aleš Mäsiar      |
| <b>Čas stretnutia:</b>       | 9:00                 |                          | Bc. Lukáš Miškovský  |
| <b>Miesto stretnutia:</b>    | STU FIIT 3.28        |                          | Bc. Zora Moravčíková |
| <b>Téma stretnutia:</b>      |                      |                          | Bc. Filip Šandor     |
| <b>Zápisnicu vypracoval:</b> | Bc. Zora Moravčíková | <b>Ďalší zúčastnení:</b> |                      |

### Vyhodnotenie zadaných úloh

- Grafika Jozefa
  - o Bug RealDraw-u, keď rotuješ package, tak sa zväčší
  - o trošku moc plastický
  - o chýba mu ucho
  - o spredu
    - Musí sa hýbať aj plášť – len sprava doľava
  - o z boku
    - len zrkadlovo, možno nejaký malý detail urobiť
  - o Trošku farby na líčka, pehy
  - o task nie je dokončený, ale jeho prvá iterácia je hotová, aby sa mohol meniť
  - o cez víkend sa to dokreslí - postava spredu
- Dokumentácia
  - o inžinierske dielo
    - Do dokumentácie Návrh, treba dať aj skicu a prekreslenie do návrhu, niečo ako story board
    - Dať do dokumentu všetko čo sa robí – skice
    - Celé kreslenie dať do návrhu, nech to nie je oddelené
    - V implementácii už môžu byť sprity, ako budú v Unity, kde vidím aj animáciu a tak všetko
    - Môže byť, že skica v návrhu a prekreslenie do implementácie
  - o Riadenie
- Abstraktná reprezentácia scény
  - o inventory musí byť vlastný typ
  - o Treba poprehadzovať jazyk na C# z Java v class diagrame

### Náplň stretnutia

- Na každom tasku treba definovať výstup, aby sme vedeli, že je hotový

- Žiadne krčenie ramenami - predmet sa bude hodnotiť podľa našich manažérskych schopností, takže zlepšovať odhady
- Ak sa chceš spojiť s Edom, napíš mu mail – mail vidí vždy, ale keď neodpisuje tak nemá čas, keď odpíše tak je možný interakcie
  - o reply bez textu je, že odpíšem neskôr, teraz nemám čas
- Každý, kto niečo urobí dá to do dokumentácie, bez nejakého formátovania a proste bude sa to tam dávať stále a budeme to tam mať a budeme mať ušetrenú robotu, lebo tam všetko bude

### Úlohy do ďalších stretnutí

- Dokončiť Jozefa zozadu
  - o a už hotový definitívne
- Urobiť pohyb Jozefa
- Implementácia stavového priestoru tej scény
- Diagram interakcií / sekvenčný diagram pre tie interakcie, kto bude s kým volať, ako sa budú meniť stavy, kto bude rozhodovať o tom, že môže sa zmeniť stav
  - o používať to v jednom, jeden model, posielala sa to medzi diagramami
- Aleš spraví databázu alebo niečo čo bude prezentovať ten stavový priestor
- Pravidlový systém – niečo vy-googliť
  - o Rule based engine
- Reprezentácia
  - o Z class diagramu sa dá vygenerovať C# kód, aby sme zistili čo nám chýba a podobne
    - Niečo ako reverzné inžinierstvo
    - Tak isto s databázou, vygeneruje SQL
- k XML treba zadať schému, tiež možné v EA – validovať, vygenerovať
- Stavový diagram pre každý objekt
  - o Commandy ?
  - o Transformovať do gramatiky TZI
  - o Stavový diagram pre scénu, pre všetky objekty, celý stavový priestor
  - o Stavový priestor musí zachytiť všetky možnosti, do ktorého stavu sa môže dostať
  - o Petriho siete ? (petri net)
    - Možno použiteľnejšie
  - o Alebo pripraviť metodiku na vytváranie scény
    - Slovník pojmov, slovník stavov
  - o Typy interakcie (look\_at, use)
    - look at / speak to – rozdielne interakcie
  - o Dialógy sú separátne vec, iba keď sa mení stav tak to tam bude zadefinované
  - o Stavy, ktoré ja nezadefinujem (mám čokoládu, ale nič s ňou) – hmm nič, nedá sa nie
  - o Každý objekt by si mal evidovať svoj stav

## I-10 Zápisnica z 10. stretnutia tímového projektu

### Tím č. 20 - Patronus

|                              |                   |                          |                      |
|------------------------------|-------------------|--------------------------|----------------------|
| <b>Stretnutie č.:</b>        | <b>10</b>         | <b>Prítomní:</b>         | Bc. Daniela Hajdu    |
| <b>Vedúci tímu:</b>          | Ing. Eduard Kuric |                          | Bc. Lukáš Marták     |
| <b>Dátum stretnutia:</b>     | 27.11. 2015       |                          | Bc. Aleš Mäsiar      |
| <b>Čas stretnutia:</b>       | 9:00              |                          | Bc. Lukáš Miškovský  |
| <b>Miesto stretnutia:</b>    | STU FIIT 3.28     |                          | Bc. Zora Moravčíková |
| <b>Téma stretnutia:</b>      |                   |                          | Bc. Filip Šandor     |
| <b>Zápisnicu vypracoval:</b> | Bc. Filip Šandor  | <b>Ďalší zúčastnení:</b> |                      |

#### Vyhodnotenie zadaných úloh

Grafika môže byť, priebežne doladovaná vo voľnom čase.

Predstavenie grafickej reprezentácie(Zorka): jednotlivé objekty zoradiť podľa časovej následnosti, okienka pomeniť tak aby boli všade veľké alebo všade malé začiatkové písmená. Opakované konverzácie s jednou postavou sa pridávajú ako speakToAvatar 1,2.... Pridať jednotné značenie (add spoken to jozef). Štart – end, prechody. Lepšia forma na reprezentáciu prerokvizít potrebných na interakciu.

Bol rozchodený source control projektu(comitovanie).

Implementácia stavu a scény(Aleš) je robená v xml. Výhody: definovaná schéma, generickosť. XML obsahuje typ interaction ktorý má definované source a destination v nejakej scene, určí sa typ interakcie, zoznam požadovaných objektov(znalostí) a zoznam možností. Takto si budeme vedieť dopytovať potrebné akcie pre dané vstupné parametre. XML sa bude robiť per scene

Game controler(Lukáš Marták) obsahuje zoznam scén, ukladá ich stavy.. pri spustení hry sa z xml načítajú všetky dáta o scéne a možných vetveniach interakcii. Následne po kliknutí na nejakú gameEntity sa spustí doInteraction. **Zistiť ako sa pridáva pole na gameEntity.**

Grafická reprezentácia hlavnej postavy(Daniela) bola splnená ale na poslednú chvíľu čo spôsobilo že animácia hlavnej postavy(Lukáš Miškovský) nebola spravená s novými skicami a zistilo sa že jednoduchá zmena skíc nieje možná. Treba teda prerobiť animáciu aj na nové skice.

Vylepšenie grafickej reprezentácie scény(Filip) bolo splnené, takisto bola vytvorená animácia pohybu vpred s novou skicou.

Cieľ šprintu bol splnení, spravili sme abstraktný návrh interakcie z objektom a vylepšili sme grafiku hlavnej postavy, scény a animácie.

## Náplň stretnutia

Treba viacej kontrolovať opisy a dbať na to aby boli obširne a definovali všetko čo sa má robiť. **Spätne doplniť opis predchádzajúce user story, vstupy, výstupy.**

Cez planing poker sme zhodnotili user story implementáciu interakcie s objektom na 5 story pointov a vybrali sme si ho pre riešenie na tomto šprinte.

Po stretnutí sa budeme stretávať a špecifikovať si poriadne náplne každej úlohy. Doplniť death line v rámci týždňa kedy má kto čo odovzdať.

Koncom týždňa musí byť implementovaná „klikateľná“ interakcie s objektom. Implementovať priamo do unity.

## Úlohy do ďalších stretnutí

Zorka – otestovať a spísať metodiku source control, vytvoriť hierarchiu v projekte

Upraviť diagram stavového priestoru kapitoly 1.

Spísať metodiky na notáciu, reprezentáciu scény.

DeathLine: sobotu ráno (29.11). – nedeľa

Aleš - spísať metodiky na vytváranie xml a napisovanie scény

Špecifikácia dialogovacieho nástroja

Implementácia manažéra interakcii

Lukáš Marták - zadefinovať metodiku písania zdrojového kódu

Implementácia herného modelu DL: sobota 8:00

Implementácia gameControler

Lukáš Miškovský – definovať metodiku na testovania, rozhodnúť sa čo sa bude testovať

Dorobiť animáciu z boku a zozadu

Interpreter

Filip Šandor - spraviť grafiku scén 1 a 2

Spísať postup na prípravu obrázkov a export animácie

Špecifikovať presun medzi scénami

Daniela Hajdu- grafika hlavnej postavy

Grafika hádankárky

Implementovať dialógové okno, s možnosťou určenia pozícií, transparentnosti, fontu, farby

Všetci - dopísať opis user story

## I-11 Zápisnica z 11. stretnutia tímového projektu

### Tím č. 20 - Patronus

|   |                   |  |
|---|-------------------|--|
| <b>Stretnutie č.:</b>   | <b>11</b>         | <b>Prítomní:</b><br>Bc. Lukáš Marták<br>Bc. Aleš Mäsiar<br>Bc. Lukáš Miškovský<br>Bc. Zora Moravčíková<br>Bc. Filip Šandor |
| <b>Vedúci tímu:</b>   | Ing. Eduard Kuric |  |
| <b>Dátum stretnutia:</b>                                      | 04.12. 2015       |  |
| <b>Čas stretnutia:</b>  | 07:30             |  |
| <b>Miesto stretnutia:</b>                                     | STU FIIT 3.28     |  |
| <b>Téma stretnutia: vyhodnotenie progresu v šprinte Lumos</b> |                   |  |
| <b>Zápisnicu vypracoval:</b>                                  | Bc. Lukáš Marták  | <b>Ďalší zúčastnení:</b>   |

#### Vyhodnotenie zadaných úloh

Filip:

- Bola navrhnutá a vytvorená grafika 2 scény
- Bol spísaný postup pre tvorbu grafiky postáv
- Rozpísaná špecifikácia presunu medzi scénami, kolízia tasku s taskom Herného kontroleru

Zorka:

- Bola spísaná metodika pre používanie source control (kedy čo a ako)
- Bola definovaná notácia pre modelovanie stavového priestoru scén
- Súborová štruktúra projektu v Unity bola definovaná

Aleš:

- Spísaný postup pre tvorbu a popis štruktúry XML súborov na definíciu interakcií v scénach
- Implementovaný manažér interakcií

Lukáš Ma:

- Spísané pravidlá pre tvorbu zdrojového kódu
- Implementovaný prototyp herného modelu
- Herný kontroler a rozsah jeho funkčnosti nebol implementovaný a musí byť dodefinovaný

Lukáš Mi:

- Implementovaný prototyp interpretera akcií

#### Náplň stretnutia

#### Grafika:

- Detaily grafiky druhej scény - treba symetrizovať uhly medzi koncami podlahy a okrajmi. Stĺpy nesmú pôsobiť flat designovo. Zostriť uhol podlahy a steny aby to vyzeralo realisticky, že tá budova nepadá.
- Grafika Jozefa: Klobúk by nemusel byť úplne rovno nasadený na hlave. Uši sú najlepšie. Od pása dole je to to hybrid koňa a človeka :D (vrátane rúk)
- Odhaľovanie bugov v RealDrawe.

#### Implementácia:

- Herný kontroler je závislý od špecifickosti Unity frameworku. Treba zistiť, čo z jeho návrhu je a čo nie je možné urobiť a definovať čo bude a čo nebude jeho súčasťou a naimplementovať ho.
- Volanie interakcií by malo byť v kóde univerzálne, a variácie by mali byť definované v XML. Teda na každý interakcieschopný objekt pôjde ten istý skript, ktorý bude volať metódy na vybranie a vykonanie interakcie.

Testovanie služieb SAV cez webový formulár. Diskusia k zaujatiu stanoviska tímu ohľadom použitia poskytnutých webových služieb.

Ďalšie stretnutie bude budúci štvrtok 10.12.2015 o 9:00

#### Úlohy do ďalších stretnutí

- Dokončiť interakciu
- Definovať procesy ktorými sa riadime v rámci šprintov
  - Ako vyberáme user stories do šprintov
  - Ako odhadujeme úlohy, ako robíme a vyhodnocujeme retrospektívu medzi šprintami
- Aleš - dialógovač (kto čo kedy povie)
- Lukáš zelený so Zorkou - dokončiť interakciu s objektom na základe interaction managera a action interpretera
- Ak to bude easy, spraviť dialógové okno - rozhranie na výpis textu



## I-12 Zápisnica z 12. stretnutia tímového projektu

### Tím č. 20 - Patronus

|                              |  |   |
|------------------------------|--|---|
| <b>Stretnutie č.:</b>        | <b>12</b>  | <b>Prítomní:</b><br>Bc. Daniela Hajdu<br>Bc. Lukáš Marták<br>Bc. Aleš Mäsiar<br>Bc. Lukáš Miškovský<br>Bc. Zora Moravčíková<br>Bc. Filip Šandor |
| <b>Vedúci tímu:</b>          | Ing. Eduard Kuric  |   |
| <b>Dátum stretnutia:</b>     | 10.12. 2015  |   |
| <b>Čas stretnutia:</b>       | 9:00   |   |
| <b>Miesto stretnutia:</b>    | STU FIIT 3.28  |   |
| <b>Téma stretnutia:</b>      | <b>zhodnotenie šprintu Lumos, report aktuálneho stavu Štefanovi, príprava na nový šprint</b> |   |
| <b>Zápisnicu vypracoval:</b> | Bc. Lukáš Miškovský  | <b>Ďalší zúčastnení:</b><br>doc. Mgr. Štefan Beňuš, PhD.<br>Ing. Marián Trnka<br>- videohovor   |

#### Vyhodnotenie zadaných úloh

Animácie – dopredu sa šmýka a preto treba zmeniť pomer rýchlosti animácie k reálnemu pohybu. Ruka trochu vyskakuje.

Interakcia – s dverami čo mení stav. Stav Jozefa sa zmení na speaking a vypíše sa obsah. Ešte nie je vyriešený výpis na obrazovku (aktuálne riešené cez debug výpis). Prechod medzi scénami ešte treba doplniť.

Pozadie – stĺpy stále nie sú oblé ináč ok. Perspektíva v prvej scéne možno stále nie je úplne vhodná.

Testovanie nie je dokončené, treba dorobiť.

Grafika – sú spravené skice avatarov. No hádankárka aj vtipkárka by mala byť niečím špecifická. Treba vymyslieť čo zaujímavé na nej treba.

#### Náplň stretnutia

Problém všeobecný – teraz sa praktikuje viac vodopádový model ako scrum. Treba na šprint viacero user stories.

V TPCupe sa hodnotí aj dokumentácia, detailne sa posudzujú rôznymi tabuľkami a medzi tímami býva minimálny rozdiel.

Mali by sme začať riešiť aj zvuky do hry. Zvuky dvier, pozadia, chôdze a podobne.

Musíme nájsť ako sme vyriešili avatarov a ich umiestnenie/meno/pozadie v predchádzajúcich zápisniciach, aj sa tým riadiť.

Pri dialógu treba vyriešiť nejaké blokovanie vstupu od hráča – nech mu to nedovolí nič iné ako čítať text. Nepotrebujeme posúvanie textu – len sa zobrazí.

Presun medzi scénami – dá sa posielat' globálny objekt z jednej scény do druhej, tento prístup.

V stavovom priestore máme hádankárku, vtipkárku aj avatarov v jednej scéne.

Potrebujeme generickú postavu na ktorej sa dá meniť oblečenie, vlasy a podobne. Aj s animáciami na úvodnú skupinu detí. Nezávislé NPC čo sa vie pohybovať zadané. Podľa Filipa by to šlo ľahko.

### Video hovor so Števom

Interakciu sme odprezentovali, výhrady neboli. Z návrhov avatarov je vhodné vybrať neutrálnu sovu čo nevzbudzuje emócie a nemala by vyzerat' moc reálne. Avatar bude prítomný ako ikonka, nebude chodiť s hráčom.

Komunikácia so SAV – ponúknuté skripty sú okej. Je pripravená slovenská syntéza čiže už to nebude brať len španielčinu – poslali linku. Pri opačnej transformácii atribút gramatika – na lepšiu identifikáciu majú Španieli možnosti čo asi môže hráč povedať v každej scéne, nám netreba. Výstupom rozoznania sú aj dva parametre – rýchlosť a tón reči, ktoré aj my budeme spätne posielat'. (je nutné dohodnúť si formu XML) Spätná transformácia pre slovenčinu by mohla byť v priebehu budúceho týždňa.

Ženský a mužský hlas – default je ženský, netreba používať tento parameter.

Časová odozva – už aj teraz je teoreticky celkom dlhá. Prenos dát nie je veľký problém, skôr čas samotnej syntézy. Optimalizovať sa to dá na strane SAV-ky, aktuálne sa znovu načítavajú všetky modely pri každom request, mala by sa robiť len syntéza. Do sekundy by mal byť rozoznaný text a poslaná späť reč.

Španielska kartová hra už zbiera dáta. V januári by mala byť slovenská verzia na vyskúšanie z našej strany.

Grant je financovaný armádou, vojak = komunikácia so strediskom. Nie je to však nutné mať v hre, možno neskôr vojenské prvky pridať. V hre nebezpečenstvo predstavujúce reálnu situáciu ale nie je potrebné to mať vizuálne s vojenskými prvkami. Postavy by mohli mať vojenské správanie a komunikáciu. Pridať kódy do komunikácie (vojenská inšpirácia).

Set miestností čo sa náhodne vyberajú sme zrušili, je tam dejová línia. Všeobecnosť do hry pridávajú minihry.

Nebude príliš komplikované podať príbeh hráčovi čo nikdy nič nehral? Nemal by, možno bude zo začiatku problém s všeobecnými princípmi hier no dá sa to dať priamo do hry (čo robí kliknutie kam a pod.)

- Do konca januára sa snád' podarí prototyp na našich dvoch scénach – už by sme chceli aj interakciu so SAVkou, minihry a nejaký ten objekt.

Syntetizátor – nevie intonovať pri otázkach.

Dialógy nahrávanie – dá sa v skúšobni. Konzistentnosť nahrávok treba riešiť. Ambient zvuky by sme mohli používať dostupné z netu po minimálnom masteringu.

Custom animácie – treba v AdventureCreatore zistiť ako ich spustiť.

Poupravovanie backlogu. Názvy taskov upraviť. Pri šprintoch dávať aj stories v štýle iterácia vylepšenia grafiky vybranej časti a pod.

Input manažér na avatara.

Planning poker: Až na príklady nižšie sme sa rýchlo dohodli na story pointoch. Zrevidovali sme niektoré userstories a v prípade potreby ich rozdelili aby mali vhodnejšiu granularitu.

Interaktívna komunikácia – nezhodli sme sa, ales upravil svoj názor po vysvetlení.

Správanie vtipkárky a hádankárky – prvé ľahšie, druhé ťažšie.

### **Úlohy do ďalších stretnutí**

Hádankarka – presunutie do ďalšieho šprintu.

Zobrazenie informácií v hre – umiestnenie v strede dole a vyriešiť vzhľad.

Implementovať návrh dialógovača.

Doplniť a porozdeľovať tasky vybraných userstories

### **Retrospektíva**

Robíme osobný manažment? Rozvrhnúť si tasky na týždeň, zapísať.

Na TP sme mali málo času kvôli koncu semestra a ostatným predmetom. No snažili sme sa vždy aspoň niečo spraviť.

Dodefinovanie taskov nám veľmi nešlo. Treba si určiť po stretnutí hodinku-dve keď sa ešte celý tím dohodne na nových úlohách a vyplnia sa aj ich metadáta.

Názvy taskov nám niekedy nesúvisia s názvami userstories. Do budúca vytvoriť viac userstories a tasky pridávať k zodpovedajúcim.

Riziká pri konkrétnych userstories treba mať evidované.