

RoboCup 3D

Tímový projekt

Dokumentácia k riadeniu projektu



Tím: Androids (tím č. 5)
Vedúci tímu: Ing. Ivan Kapustík
Členovia tímu: Bc. Juraj Belanji
Bc. Miroslav Hruška
Bc. Roman Kováč
Bc. Andrej Minárik
Bc. Veronika Wolfová
Študijný odbor: Softvérové inžinierstvo
Akademický rok: 2010/2011



Obsah

Obsah	2
Úvod	3
Plán projektu	3
Roly a úlohy členov tímu	5
Riziká tímového projektu	9
Štábna kultúra	10
Všeobecné pravidlá a informácie pre projekty	10
Špecifické pravidlá pre vývoj kódu pre agenta JIM	11
Špecifické pravidlá pre vývoj kódu pre testovací framework	12
Ostatné	13
Pravidlá testovania a prehliadok kódu	13
Stručný opis nástrojov pre správu projektu	14
dotProject – plánovanie úloh	14
Google groups – komunikácia tímu	14
Instant messaging – rýchla komunikácia	14
SVN – systém na správu verzií	14
Ponuky – časť A	1
Tím	1
Ponuka č. 1 – Tréner mentálnych schopností	3
Ponuka č. 2 – Dizajn s použitím obohatenej reality	5
Ponuka č. 3 – Model používateľa pre jeho identifikáciu	7
Zoradenie tém podľa priority	9
Záznamy zo stretnutí – časť B	1
Metodiky a návody na použitie – časť C	1
Návod na nastavenie SVN	2
Návod na zverejnenie zápisníc	6
Metodika pre manažment úloh v distribuovanom projekte prostredníctvom nástroja dotProject	7
Metodika pre manažment verzií zdrojového kódu	15
Preberacie protokoly – časť D	1



Úvod

Tento dokument obsahuje informácie týkajúce sa manažmentu a riadenia tímového projektu RoboCup 3D tímu číslo 5 – Androids. Dokument je vytvorený v akademickom roku 2010/2011 pre účely predmetu Tímový projekt. Všetky informácie boli zbierané priebežne a zodpovedajú reálnemu stavu a priebehu riadiacej práce na projekte.

Plán projektu

Táto časť obsahuje základný plán projektu pre zimný semester a hrubý odhad plánov pre letný.

Týždeň	Plán	Štart	Koniec	Zodpovedný
1. týždeň	Výber témy tímového projektu	20.09.2010	27.09.2010	všetci
	Vypracovanie ponúk	20.09.2010	27.09.2010	Juro
2. týždeň	Dohodnutie o stretnutiach tímu	27.09.2010	29.09.2010	Veronika
	Naštudovanie témy RoboCup 3D	29.09.2010	06.10.2010	Veronika
3. týždeň	Analýza existujúcich RoboCup tímov	06.10.2010	13.10.2010	Veronika
4. týždeň	Vytvorenie web stránky tímu	13.10.2010	18.10.2010	Andrej
	Výber softvéru na plánovanie a manažment tímu	13.10.2010	20.10.2010	Maťo
	Vytvorenie základného dokumentu pre projektovú dokumentáciu	13.10.2010	20.10.2010	Juraj
	Vytvorenie základného plánu	13.10.2010	20.10.2010	Juraj
5. týždeň	Výber tímu, na ktorý budeme nadväzovať	20.10.2010	24.10.2010	všetci
	Špecifikácia požiadaviek a vytvorenie návrhu pre implementáciu prototypu	20.10.2010	03.11.2010	Veronika
6. týždeň	Spracovanie špecifikácie požiadaviek do dokumentu	27.10.2010	01.11.2010	Juraj
7. týždeň	Revízia dokumentu na prvé odovzдание	01.11.2010	03.11.2010	Veronika
	Odovzдание dokumentácie analýzy problému, špecifikácie požiadaviek a návrhu riešenia	03.11.2010	03.11.2010	Veronika
8. týždeň	Výber častí návrhu, ktoré sa budú implementovať do prototypu	03.11.2010	14.11.2010	Veronika



	Implementácia testovacieho frameworku	03.11.2010	14.11.2010	Miro
	Práca na základných pohyboch	03.11.2010	24.11.2010	Veronika
9. týždeň	Transformácia XML súborov	15.11.2010	24.11.2010	Andrej
	Práca na editore pohybov	15.11.2010	28.11.2010	Juraj
10. týždeň	Rozpracovanie prototypu	22.11.2010	28.11.2010	Veronika
11. týždeň	Príprava prototypu na odovzdanie	29.11.2010	05.12.2010	Veronika
	Revízia dokumentácie, pridanie implementačnej dokumentácie	01.12.2010	08.12.2010	Juraj
12. týždeň	Príprava prototypu na prezentáciu, konečné úpravy	06.12.2010	12.12.2010	všetci
	Odovzdanie prototypu spolu s dokumentáciou	14.12.2010	14.12.2010	Veronika
	Používateľská prezentácia prototypu	15.12.2010	21.12.2010	všetci

Letný semester:

Týždeň	Plán	Zodpovedný
1-4. týždeň	Úprava a rozšírenie prototypu	všetci
3-7. týždeň	Práca na produkčnom systéme pre vyššie správanie	Juraj
4-8. týždeň	Práca na testovaní pohybov a vyššej logike agenta	Roman
9. týždeň	Odovzdanie produktu a dokumentácie k produktu	Veronika
8-11. týždeň	Rozpracovanie dokumentácie	Juraj
	Oprava chýb, testovanie, overenie	Roman
10-12. týždeň	Prevádzka produktu	Veronika
12. týždeň	Odovzdanie finálnej verzie produktu a dokumentácie	všetci



Roly a úlohy členov tímu

V tejto časti dokumentácie je rozpracované, ktorý člen tímu vytvoril jednotlivé časti dokumentácie, a taktiež sú uvedené aj jednotlivé roly členov tímu. Jednotlivé konkrétne úlohy členov tímu sú rozpísané v zápisoch zo stretnutí.

Je nutné upozorniť, že pôvodne tím pozostával zo šiestich členov, avšak v piatom týždni semestra sa jeden člen tímu (Matej Moravanský) s nami rozlúčil, keďže sa rozhodol, že nebude pokračovať v štúdiu. Následne sa k nám ako externý člen tímu pridal Daniel Erban, ktorý nám čiastočne bude pomáhať.

Bc. Veronika Wolfová – vedúca tímu, manažérka podporných prostriedkov (od 5. týždňa)

- v dokumentácii k produktu vypracovala nasledovné časti:
 - 2.1.6 – analýza tímu Nexus 3D
 - 2.1.12 – analýza tímu Neurotics
 - 3.1 – špecifikácia pohybov agenta
 - 4.1 – hrubý návrh implementácie pohybov
 - 5.2.2 – základné pohyby (všetky pohyby okrem kopov)
- v dokumentácii k riadeniu vypracovala nasledovné časti:
 - Ponuka č. 2 – Dizajn s použitím obohatenej reality
 - Zápis č. 2 zo stretnutia 06.10.2010
 - Zápis č. 6 zo stretnutia 03.11.2010
 - Metodika pre manažment úloh prostredníctvom nástroja dotProject
 - Preberací protokol k dňu 03.11.2010
 - Preberací protokol k dňu 14.12.2010
- iné dôležité úlohy:
 - kontrola gramatickej a štylistickej správnosti dokumentácie a overenie dokumentácie
 - rozdelenie úloh a správa dotProject

Bc. Juraj Belanji – zástupca vedúcej tímu, manažér plánovania

- v dokumentácii k produktu vypracoval nasledovné časti:
 - Kapitola 1 – úvod
 - 2.1.1 – analýza tímu Kouretes
 - 2.1.7 – analýza tímu Agenty 007
 - 2.1.13 – zhodnotenie tímov
 - 2.2 – analýza servera
 - 2.4.1 – multiagentové systémy
 - 3.3 – špecifikácia rozšírenia editora pohybov
 - 4.3 – návrh úprav v editore pohybov



- 5.3.2 – symetria pohybov
- 5.3.3 – previazanie kľbov
- Prechodové časti dokumentácie
- v dokumentácii k riadeniu vypracoval nasledovné časti:
 - Ponuka č. 2 – Dizajn s použitím obohatenej reality
 - Zápis č. 1 zo stretnutia 29.09.2010
 - Zápis č. 5 zo stretnutia 27.10.2010
 - Zápis č. 11 zo stretnutia 08.12.2010
 - Plán projektu
 - Roly a úlohy členov tímu
 - Stručný opis nástrojov pre správu projektu
- iné dôležité úlohy:
 - zostavenie dokumentácie k produktu a k riadeniu
 - koordinátor dokumentácie
 - vytvorenie plánu projektu

Bc. Miroslav Hruška – manažér kvality

- v dokumentácii k produktu vypracoval nasledovné časti:
 - 2.1.2 – analýza tímu Nao Team Humboldt
 - 2.1.12 – analýza tímu Critical Error
 - 2.2 – zlepšenia vo verzii servera 0.6.4
 - 2.3 – analýza agenta JIM
 - 3.5 – špecifikácia testovacieho frameworku
 - 4.5 – návrh testovacieho frameworku
 - 5.1 – implementácia testovacieho frameworku
 - 5.1.1 – ukážka používateľom zadaného skriptu
- v dokumentácii k riadeniu vypracoval nasledovné časti:
 - Ponuka č. 1 – Tréner mentálnych schopností
 - Štábna kultúra
 - Zápis č. 7 zo stretnutia 10.11.2010
 - Zápis č. 10 zo stretnutia 01.12.2010
- iné dôležité úlohy:
 - prepracovanie agentov na server 0.6.4
 - analýza štábnej kultúry a vypracovanie metodiky pre písanie kódu

Bc. Roman Kováč – manažér vývoja

- v dokumentácii k produktu vypracoval nasledovné časti:
 - 2.1.3 – analýza tímu FC Portugal



- 2.1.9 – analýza tímu RoboKit
- 3.2 – špecifikácia vyššej logiky agenta
- 4.2 – návrh vyššej logiky pohybov agenta
- 5.1.2 – logovanie servera
- 5.1.3 – parsovanie správ zo servera
- v dokumentácii k riadeniu vypracoval nasledovné časti:
 - Ponuka č. 1 – Tréner mentálnych schopností
 - Zápis č. 3 zo stretnutia 13.10.2010
 - Zápis č. 8 zo stretnutia 16.11.2010
 - SVN – systém na správu verzií
 - Návod na nastavenie SVN
 - Metodika pre manažment verzií zdrojového kódu
- iné dôležité úlohy:
 - správa SVN
 - koordinátor implementácie

Bc. Andrej Minárik – manažér rizík

- v dokumentácii k produktu vypracoval nasledovné časti:
 - 1.1 – zadanie projektu
 - 2.1.4 – analýza tímu UT Austin Villa
 - 2.1.10 – analýza tímu Hviezdna jedenástka
 - 2.4.2 – vylepšenie stability agenta
 - 3.4 – špecifikácia použitia jazyka XABSL
 - 4.4 – návrh vyššieho správania v XABSL
 - 5.2.1 – pomôcka pri vytváraní pohybov
 - 5.2.2 – základné pohyby (kopy)
 - 5.3.1 – úprava XML exportu pohybov
- v dokumentácii k riadeniu vypracoval nasledovné časti:
 - Ponuka č. 3 – Model používateľa pre jeho identifikáciu
 - Zápis č. 4 zo stretnutia 20.10.2010
 - Zápis č. 9 zo stretnutia 24.11.2010
 - Riziká tímového projektu
- iné dôležité úlohy:
 - vytvorenie a udržiavanie web stránky tímu

Bc. Matej Moravanský – manažér podporných prostriedkov (do 4. týždňa)

- v dokumentácii k produktu vypracoval nasledovné časti:
 - 2.1.5 – analýza tímu SEU-3D



- 2.1.11 – analýza tímu RoboKopy
- v dokumentácii k riadeniu vypracoval nasledovné časti:
 - Ponuka č. 3 – Model používateľa pre jeho identifikáciu
- koncom 4. týždňa semestra opustil tím



Riziká tímového projektu

Krátko po začiatku projektu boli identifikované riziká, ktoré sa ukazovali ako potenciálne ohrozujúce úspech projektu. Pri každom riziku je uvedená (subjektívne) ohodnotenie na základe pravdepodobnosti nastania daného rizika a vážnosti jeho dopadu na projekt v prípade nastania.

Riziko straty jedného z členov tímu skutočne nastalo, a to v piatom týždni, kedy z časových dôvodov Matej Moravanský ukončil štúdium, a teda aj pôsobenie v tímovom projekte. Aktívnym prístupom k situácii sme zabezpečili prenos nezastrešených zodpovedností (konkrétne prehľad a voľbu podporných prostriedkov a spolupúčasť na spravovaní web stránky) na zvyšných členov tímu.

Oblasť rizík	Identifikované riziko	Dôležitosť rizika				Ošetrenie rizika	Pozn.	
		Pravdepodobnosť nastania (1-10)	X	Miera dopadu (1-10)	=			Ohodnotenie rizika (1-100)
Personálne nedostatky	Nedostatok času a pracovná vyťaženosť, potenciálne vedúce k odchodu člena tímu	6	X	8	=	48	Jasná komunikácia, v prípade nastania delegovanie nepridelených úloh na ostatných členov tímu, prebratie zodpovednosti za chýbajúceho člena tímu.	Situácia skutočne nastala v piatom týždni.
Nedodržanie termínov	NedisCIPLINOVANOSŤ členov tímu v dodržiavaní stanovených termínov	7	X	6	=	42	Stanovenie dostatočne skorých termínov, kontrola splnenia úloh vedúcou tímu.	
Nedodržanie požiadaviek	Vyvinutie nesprávnych funkcionality hráča	5	X	8	=	40	Dôkladná analýza existujúcich riešení, dobré a presné špecifikovanie požiadaviek na hráča. Definovanie spôsobu overenia, vytvorenie prototypu.	
Nedostatky v externých komponentoch	Skryté nedostatky hráča zvoleného na nadviazanie	4	X	7	=	28	Dôkladná analýza dokumentácie hráča, analýza zdrojového kódu, posúdenie kvality hráča viacerými členmi tímu, v prípade objavenia nedostatkov ich prioritne správne implementovanie.	
Problémy s hardvérom	Nedostatočný výkon potrebný na úspešnú simuláciu	5	X	3	=	15	Využitie výkonnejšieho hardvéru (aspoň dvojitý procesor), spúšťanie servera a jednotlivých hráčov na samostatných strojoch, ich zosieťovanie.	



Štábna kultúra

V nasledovnej časti je definícia štábnej kultúry tímu Androids pre projekt RoboCup 3D. Štábna kultúra zhŕňa základné všeobecné konvencie tvorby a vývoja softvéru a definuje formálnu stránku vývoja v prostredí Eclipse pre aplikácie, ktoré tím vyvíja.

Všeobecné pravidlá a informácie pre projekty

Nasledujúca kapitola zhrnie všeobecné pravidlá pre vývoj a testovanie softvéru. Uvedené pravidlá sú štandardnými konvenciami a riadi sa nimi aj náš tím.

Pomenovanie

Všetky názvy použité pri vývoji ktorejkoľvek časti kódu alebo pomenovania súborov musia byť v *anglickom* jazyku.

Element	Pravidlá názvu elementu
Balíky	<ul style="list-style-type: none"> • Výlučne malé písmená • Slovná forma <i>podstatného</i> mena • Názov musí zapadať do logickej štruktúry predchádzajúcich balíkov
Súbory, triedy	<ul style="list-style-type: none"> • Zložené z <i>podstatných</i> mien • Každé podslovo začína veľkým písmenom, ostatné písmená sú malé • Medzi podslovami nie je explicitná medzera (je určená veľkým písmenom ďalšieho podslova)
Metódy	<ul style="list-style-type: none"> • Prvé podslovo je <i>sloveso</i>, začína malým písmenom • Zvyšné podslová veľkým písmenom, bez explicitnej medzery
Premenné	<ul style="list-style-type: none"> • Ak premenná slúži na indexáciu, postačuje jednopísmenové slovo • Prvé podslovo začína malým písmenom, zvyšné podslová veľkým
Konštanty	<ul style="list-style-type: none"> • Všetky písmená veľké • Medzera je určená znakom " _ "



Testovanie

Element	Pravidlá názvu elementu
Súbory, triedy	<ul style="list-style-type: none"> Názov vychádza z testovanej triedy/súboru, obsahuje navyše na konci slovo <i>Test</i>

Špecifické pravidlá pre vývoj kódu pre agenta JIM

Anotácie

V projekte JIM sa používajú okrem základných anotácií aj nasledovné anotácie, ktoré je nutné dodržiavať pre udržanie meta-informácií o kóde.

Anotácia	Použitie
@Bug	Označenie elementov kódu, ktoré nepracujú tak, ako majú.
@Problem	Označenie elementov kódu, v ktorých sa našiel <i>problém</i> pri prehliadke kódu.
@Refactor	Označenie elementov kódu, ktoré je nutné podrobiť refaktoringu.
@ReviewOk	Označenie elementov kódu, ktoré: <ul style="list-style-type: none"> sú dôkladne otestované Unit testami prešli prehliadkou kódu boli otestované voči serveru
@UnderConstruction	Označenie elementov kódu, ktoré nebudú zasiahnuté prehliadkou kódu.

Komentáre

Pre dodržanie rovnakej kultúry v projekte JIM, do ktorého prispievajú viacerí autori, je potrebné používať nasledovné komentáre. Nastavenie komentárov sa viaže na prostredie Eclipse, ktoré náš tím na vývoj kódu používa.

Element	Obsah
Types	<pre>/** * * \${file_name} * *@Title Jim *@author \$\$Author: \${user} \$\$</pre>



	<code>*/</code>
Methods	<code>/** * TODO: Replace with purpose of method. Start with verb. * * \${tags} */</code>

Špecifické pravidlá pre vývoj kódu pre testovací framework

Vývoj aplikácie *Testovací framework* podlieha vo všeobecnosti rovnakým princípom, aké už boli v dokumente naznačené. Rozdiel je len vo formálnych pravidlách vývoja kódu a šablón pre používanie komentárov.

Vývoj kódu a komentáre

Vývoj kódu a tvorba komentárov sa riadi nasledovnými pravidlami:

- maximálna dĺžka riadku 80 znakov (štandardná konvencia)
- okomentovaná *každá* metóda (okrem metód typu *getter* a *setter*), trieda podľa šablóny
 - komentár môže byť minimálny, je však *záväzný* kód aspoň minimálne okomentovať
 - v prípade triviálnych metód, ktoré dostatočne vyjadrujú funkcionality skrze názvy parametrov, *nie sú* komentáre parametrov ani výstupnej hodnoty nutné
 - dôležité pri komentovaní je dbať na to, aby existencia komentáru bola prospešná

Nasledujúca tabuľka zhrňa elementy a ich nastavenie pre používanie komentárov:

Element	Obsah
Types	<code>/** * TODO: Replace with a brief purpose of class / interface. * * @author \${user} * * \${tags} */</code>
Methods	<code>/** * TODO: Replace with a brief purpose of method. * Start with verb. * * @author \${user} * * \${tags} */</code>



Ostatné

Tím sa zaoberá aj malými úpravami (aktualizáciou) aplikácie *RobotMotionEditor*. Nakoľko zmeny, ktoré sú v aplikácii vytvorené sú v porovnaní s veľkosťou aplikácie malé, nie je nutné dodatočne štabnu kultúru pre daný projekt špecifikovať.

Pravidlá testovania a prehliadok kódu

Testovanie

Tím vyvíja softvér inkrementálnym a iteratívnym spôsobom. Z tohto hľadiska je nutné si uvedomiť, že jednotlivé funkcie systému budú na seba prirodzene nadväzovať. V záujme vyhnutia sa neskorším implementačným ťažkostiam, je potrebné dodržiavať nasledovné zásady pri tvorbe kódu:

- *každá* podstatná komplexná funkcia systému, ktorá bude využívaná viacerými funkciami na vyššej úrovni logiky, *musí* byť pokrytá testami
- je vhodnejšie testy písať pred samotným kódom (inšpirovať sa prístupom Test-Driven Development), lebo bolo preukázané, že písanie testu po kóde neodhalí toľko chýb, koľko by odhalilo v opačnom prípade (človek má tendenciu napísať test, ktorý vyhovuje kódu)
- radšej test napísať, ako nenapísať
- je to priama, dôveryhodná spätná väzba od systému
- čas vynaložený na písanie testov by sa mal vrátiť neskôr v podobe skrátenia času stráveného debugovaním

Prehliadky kódu

Nakoľko tím je malý, pracujúci na nezávislých častiach systému, nie sú prehliadky kódu zatiaľ nutné. S rastúcim množstvom implementácie netriviálnej funkcionality, ktorá sa udeje najmä v letnom semestri bude tieto prehliadky potrebné vykonávať. Frekvencia prehliadok bude závisieť od množstva implementovanej funkcionality a výskytu problémov.



Stručný opis nástrojov pre správu projektu

V tejto časti dokumentu sa nachádza stručný opis nástrojov, ktoré používame pri práci na tímovom projekte.

dotProject – plánovanie úloh

Na správu a plánovanie úloh projektu používame dotProject. Sem zaznamenávame všetky úlohy, ktoré identifikujeme na stretnutiach tímu. V tomto nástroji tak môžeme sledovať celkový priebeh projektu a môžeme sledovať dodržiavanie termínov plnenia úloh jednotlivých členov tímu. Keďže všetci členovia tímu majú administrátorské práva, každý člen môže pridávať a meniť úlohy. Zhodli sme sa však, že úlohy bude do dotProject pridávať iba vedúca, Bc. Veronika Wolfová, a ostatní členovia budú iba zaznamenávať svoj pokrok v splňaní jednotlivých, im pridelených úloh.

Google groups – komunikácia tímu

Na všeobecnú komunikáciu tímu používame Google groups. Skupina nášho tímu sa nachádza na stránke <http://groups.google.com/group/team05fiit?hl=sk>. Tento spôsob komunikácie využívame najčastejšie, aby každý člen tímu bol oboznámený so všetkými aspektmi práce na tímovom projekte. Cez dokumentový server skupiny taktiež zdieľame dokumenty, takže každý člen tímu je schopný prezrieť si nejakú časť dokumentácie a prípadne ju upraviť alebo vyjadriť pripomienky.

Zatiaľ nám tento spôsob zdieľania dokumentácie vyhovuje, avšak v budúcnosti plánujeme previesť zdieľanie dokumentov na SVN.

Instant messaging – rýchla komunikácia

Na rýchlu komunikáciu používame prevažne ICQ. Tento spôsob komunikácie používame hlavne v prípade, že sa nejaký člen tímu potrebuje rýchlo poradiť s iným členom tímu ohľadom nejakého akútneho problému. Často tento typ komunikácie používame tiež pri dohovore stretnutí.

SVN – systém na správu verzií

Ako systém na manažment verzií zdrojového kódu sme si zvolili nástroj Subversion, ktorý obsahuje pre nás nevyhnutnú funkcionálnosť:

- Organizovanie projektu
- Stiahnutie zdrojových kódov
- Vytváranie vetiev
- Sledovanie zmien v kóde
- Potvrdzovanie zmien v kóde



- Riešenie konfliktov
- Spájanie vetiev

System Subversion je podporovaný aj samotným vývojovým prostredím Eclipse, pričom je nutné stiahnuť jeden z pluginov pre jeho podporu. Zvolený bol plugin Subclipse. Pri práci s ním je potrebné dodržiavať nasledovné pravidlá:

1. Nepotvrdzovať zmeny, ktoré znemožňujú kompiláciu projektu.
2. Zamknúť súbor, ktorý chceme editovať, v prípade, že sme sa nedohodli s iným členom tímu na spoločnej práci s daným súborom.
3. Nezamykať súbory, ktoré nejdeme meniť.
4. Podrobne opisovať vykonané zmeny.
5. Potvrdzovať čo možno najmenšie ucelené zmeny a čo možno najčastejšie.
6. V prípade, že si nevieme rady s riešením konfliktu, kontaktovať vedúceho vývoja.
7. Pred začatím práce a aj po potvrdení zmien si stiahnuť najnovšiu verziu súborov.

Štruktúra súborov bola rozdelená na 5 častí:

- dokumentácia
- zdrojové súbory agenta – robota
- editor pohybov (a prípadne aj správania)
- zdrojové súbory testovacieho frameworku
- iné zdrojové súbory



Ponuky – časť A

V tejto časti dokumentu sa nachádzajú ponuky k témam v rámci výberu témy tímového projektu.

Tím

Bc. Juraj Belanji

Počas štúdia získal skúsenosti s programovaním v jazykoch C a Java. Má manažérske a organizátorské skúsenosti, keďže v minulosti organizoval niekoľko študentských stretnutí. V oblasti IT je najmä skúsený s nástrojmi, akými sú Rational Software Architect firmy IBM a niektoré nástroje pre modelovanie vlastností, keďže v tejto oblasti vytvoril aj svoju bakalársku prácu. Je dobrým matematikom a analytikom, pričom v minulosti pracoval na návrhu dizajnu niekoľkých interaktívnych user-friendly aplikácií. Vo voľnom čase rád rieši logické úlohy.

Bc. Miroslav Hruška

Zaujíma sa o objektové programovanie, 2D grafiku a umelú inteligenciu (najmä neurónové siete). Objektové programovanie realizuje najmä v prostredí Netbeans a s programovacím jazykom Java. V danom prostredí vytvoril aj niekoľko jednoduchých mobilných aplikácií založených na J2ME. V prípade, že by bol projekt webová aplikácia, bolo by z jeho strany možné pracovať na tvorbe Java appletov. Rovnako aj v prípade, že by si projekt vyžadoval tvorbu mobilnej aplikácie.

Bc. Roman Kováč

V zamestnaní pracuje na projektoch založených na platforme .NET framework s dôrazom na najnovšie technológie ako Silverlight alebo WPF. Tieto znalosti využil aj vo svojej bakalárskej práci, v ktorej navrhol on-line hru pre viacerých hráčov. Počas štúdia nadobudol skúsenosti s programovacím jazykom Java, v ktorom by sa rád naďalej zdokonaľoval. Jeho silné stránky sú spoľahlivosť a ochota pomôcť. Je skúsený v oblasti návrhu hier a spôsobov kontroly programov.

Bc. Matej Moravanský

Počas štúdia sa naučil programovať v jazykoch C, Java a spoznal techniky ako procedurálneho, tak aj objektového programovania. Bakalársku prácu vypracoval na tému grafická vizualizácia činnosti vyrovnávacej pamäte počítača. Táto práca bola realizovaná v e-learningovom prostredí Moodle a obsahovala viacero interaktívnych častí vytvorených s použitím programu Adobe Flash. Popri škole sa venoval práci na pozícií IT technika vo firme zaoberajúcej sa predajom a servisom vozidiel, kde bol pri implementácii nového informačného systému a zároveň nahradení starého. Aktuálne pracuje v reklamnej agentúre na pozícií IT manažéra a koordinuje implementáciu informačného systému pre približne 40 ľudí, ako aj chod celej firmy po technickej stránke. Taktiež získal skúsenosti s komunikáciou s ľuďmi, keďže im zabezpečuje každodennú technickú podporu.



Bc. Veronika Wolfová

Napriek tomu, že znalosťami v oblasti IT a programátorskými schopnosťami nepatrí k najlepším, oplýva vlastnosťami potrebnými pre riadenie, organizáciu a komunikáciu v tíme. Jej silnými stránkami sú zodpovednosť, trpezlivosť, spoľahlivosť a obetavosť. Jej najväčšou prednosťou je silný cit pre gramatiku a prirodzené jazyky ako také. V neposlednom rade kladie dôraz nie len na obsahovú a funkčnú stránku vecí, ale aj na ich praktickosť a estetický aspekt. Nechýba jej ani predstavivosť pri návrhu riešení a dizajnu. Vo voľnom čase sa často venuje riešeniu logických úloh, hraniu hier rozvíjajúcich jednotlivé stránky psychiky človeka, získavaniu informácií z rozmanitých oblastí vedy a života a experimentovaniu s aplikáciami na tvorbu a úpravu digitálneho obrazu.

Bc. Andrej Minárik

Absolvoval na FIIT bakalárske štúdium v študijnom programe Počítačové systémy a siete. Zaoberal sa programovaním v jazykoch C, Java a vo voľnom čase sa venuje oblasti webu. Zaujíma sa o interakciu s používateľom a user experience. Popri škole pôsobí v mládežníckej organizácii, ktorej cieľom je osobnostný rozvoj mladých ľudí. Do jeho záberu patrí práca so statickou 2D grafikou, čo podporuje aj jeho záujem o grafiku v 3D. Motivuje ho pracovať na projekte, ktorý má potenciál nájsť si využitie v praxi a dá mu možnosť učiť sa nové veci.



Ponuka č. 1 – Tréner mentálnych schopností

Motivácia – Fyzický tréning? Prečo nie aj mentálny tréning?

Táto téma nás zaujala hneď z viacerých dôvodov. Prvým dôvodom je samotné zameranie témy na **trénovanie mentálnych schopností**, ktorými disponuje každý z nás a sú kľúčové pri **strategických rozhodnutiach** či už počas života, ale aj všeobecne pri riešení problémov. Je preto samozrejmé, že danú problematiku pokrýva niekoľko viac alebo menej úspešných riešení.

Ďalším dôvodom pre výber tejto témy je výzva v návrhu a implementácii vlastného riešenia, ktoré by odstránilo nedostatky existujúcich riešení a prišlo s novými myšlienkami. Vhodným návrhom takéhoto riešenia by bolo možné zozbierať množstvo údajov, ktoré by sa dali štatisticky vyhodnocovať. Príkladom môže byť sledovanie **progressu hráča pri určitej hre**, následne zavedenie novej hry a sledovanie jeho progressu v nej. Taktiež by sa mohla sledovať aj reakcia hráča na zmenu v tej istej hre. Ak miera progressu ostane konštantná pri novej alebo zmenenej hre, je možné potvrdiť výsledky štúdií, ktoré sa zhodujú v tom, že tréning mysle hrou prináša zlepšenia iba v danej hre, a teda tieto nadobudnuté zručnosti nie sú prenositeľné inam.

Bolo by určite nesmierne zaujímavé vytvoriť hry, ktoré sú na abstraktnej úrovni takmer identické, ale ich konkrétna realizácia vytvorí hry rozdielne a v danom prípade taktiež sledovať progres. V neposlednom rade ide o motiváciu vytvoriť používateľsky prívetivé grafické rozhranie, využitie našich skúseností s tvorbou takých rozhraní a zdokonalenie sa v ich tvorbe.

Konceptuálne riešenie

Na internete je možné nájsť niekoľko webových portálov zaoberajúcich sa rozvojom mentálnych schopností či logického myslenia. Napríklad portál *Math Playground* umožňuje používateľom hrať mnohé hry ako sudoku, hexagon, hanojské veže, Rubikovu kocku a iné, avšak v danom prípade chýba monitorovanie konkrétneho hráča a jeho napredovania v danej hre. Aplikácia by mala byť koncipovaná tak, aby používateľ práve tieto informácie mal dostupné a prehľadné aj pre ostatné hry. Bolo by zaujímavé vyskúšať, či možnosť porovnávať svoje výsledky s výsledkami iných by pôsobila na hráčov motivujúco alebo demotivujúco.

Existuje široká skupina ľudí, ktorí by mohli hry hrať, preto je vhodné hru vytvoriť ako webový portál. Vzhľadom na bohaté skúsenosti s vývojom programov v platforme Java by tvorba hier pozostávala z tvorby appletov. Dané riešenie je univerzálne a s možnosťami, ktoré Java ponúka, je za prijateľných okolností možné vytvoriť graficky príťažlivé hry v rozumnom čase. Webový portál musí ponúkať možnosť jednoznačne identifikovať používateľa, aby bolo možné uskutočňovať hodnotné štatistické prieskumy.

Portál by bolo vhodné rozdeliť minimálne do dvoch častí: rozvoj myslenia a rozvoj pamäti. Taktiež by portál mal aspoň stručne zhromažďovať výsledky doterajších štúdií a niekde v statickej časti portálu by



boli k dispozícii. Okrem bežného on-line hrania hier by mohol portál ponúkať niečo špecifické k danému dňu, aby tam používateľ zakaždým našiel niečo nové. Napríklad číselné postupnosti by mohli byť vždy v danú polnoc generované pomocou rafinovane navrhnutého algoritmu v niekoľkých verziách obtiažnosti a používateľ by mal za úlohu po zobrazení si “problému dňa” daný problém vyriešiť.

Do úvahy pripadá aj možnosť prepojenia systému s mobilnými Java aplikáciami spustiteľnými na moderných mobilných telefónoch, aby mal používateľ kedykoľvek prístup k hrám, prípadne mohol rozohranú hru na portáli dohrať na svojom mobilnom telefóne a naopak.

Náročnosť hier by bola závislá od aktuálnych mentálnych schopností používateľa tak, aby hráčov prílišná náročnosť neodradila, ale aj aby ich nízka náročnosť nezačala nudiť. Z tohto dôvodu bude nutné analyzovať možné typy hráčov a otestovať systém na vhodne zvolenej vzorke hráčov, ktorá by dokázala odhaliť možné nedostatky systému.

Ďalšia možnosť rozšírenia systému by bola v oblasti zamestnávania, kde by sa pomocou určitého súhrnu hier dalo otestovať, či je používateľ mentálne „schopný“ a vhodný pre danú prácu.



Ponuka č. 2 – Dizajn s použitím obohatenej reality

Motivácia – Môže sa sen stať realitou?

Predstavme si manželský pár, ktorý ide nakúpiť nábytok pre svoj nový dom. Vchádzajú do obchodného domu s nábytkom a vidia tam stovky skríň, stoličiek, stolov, poličiek a iných vecí. Potrebujú sa rozhodnúť a vybrať si z toho množstva iba niekoľko. Ako sa rozhodnú? Musia si predsa predstaviť ten nábytok v ich dome. Ľudská predstavivosť je silný nástroj, ale na vyzdobenie domu potrebujeme zájsť o krok ďalej. Téma dizajnu s použitím obohatenej reality nás zaujala najmä z toho dôvodu, že by sme mohli vytvoriť aplikáciu, ktorá by pomáhala ľuďom pri realizovaní ich snov.

S obohatenou realitou by používatelia mohli priamo v obchodnom dome s nábytkom vytvárať kombinácie nábytku a priamo ich vkladať do priestoru ich domu. Takýto interaktívny spôsob nákupu by sa mohol uplatniť nie len v prípade nábytku, ale aj dizajnu šatstva, záhrad, reklám a iných výrobkov a služieb.

Ľudská predstavivosť je ďalším dôvodom, prečo nás táto téma zaujala. Prevažne možnosť transformácie ideí na skutočnosť použitím obohatenej reality. Veľakrát si človek kúpi nejaký kus odevu z dôvodu, že si v hlave vytvorí predstavu o tom, ako v ňom bude vyzerieť, a keď príde domov zistí, že to nie je také, ako si predstavoval a nie je to v kombinácii s nejakým iným kusom odevu prijateľné. Pomocou obohatenej reality a priameho vkladania objektov do obrazu by si používateľ mohol priamo na mieste, kde nakupuje tovar, cez mobil alebo počítač v predajni prezrieť, ako to v skutočnosti bude vyzerieť.

Bolo by nesmierne zaujímavé vytvoriť aplikáciu, ktorá bude široko zaužívaná nie len ako praktická, ale aj ako zábavná.

Konceptuálne riešenie

Na vytvorenie aplikácie bude potrebné najskôr urobiť analýzu problematiky. Potrebujeme najmä určiť, v ktorých oblastiach by sa mohla používať obohatená realita, a najmä do akej miery by sa dala uplatniť. Následne potrebujeme vybrať, na ktorú oblasť sa budeme sústrediť.

Po výbere oblasti realizácie musíme prispôbiť aplikáciu tak, aby mohla dvojrozmerné fotky pretransformovať do 3D priestoru (ak ide napríklad o úpravu nábytku v dome). Potrebujeme umožniť používateľovi transformáciu vlastných fotiek domu na jeho 3D model. V tomto prípade používateľ taktiež musí určiť, kam bude vkladať objekty.

Aplikáciu je potrebné navrhnuť tak, aby bola prispôsobiteľná prostrediu a nezávislá na operačnom systéme. Potrebujeme umožniť používateľovi priniesť si do podniku vlastné 3D prvky, ktoré bude vkladať do priestoru. Do úvahy pripadá aj možnosť prepojenia systému s mobilnými Java aplikáciami spustiteľnými na moderných mobilných telefónoch, aby mal používateľ kedykoľvek možnosť cez



kameru si prezrieť priestor, pridať doň identifikátory miesta a následne aj samotné objekty obohatenej reality.

Základným problémom pri vkladaní objektov je pridávanie objektov v rozmere. Aplikáciu navrhujeme tak, aby bolo možné meniť veľkosť vkladáných objektov. Niekedy aj veľkosť mení na konečnom dojme.

Pri realtime vkladaní objektov, teda pri vkladaní objektov do kamerou snímaných častí, potrebujeme vložený objekt natvrdo usadiť do snímanej plochy tak, aby sa pri pohybe kamerou správal reálne, čiže aby bol na identifikovanom mieste a pri zmene uhla snímania sa správal reálne.

Po navrhnutí a implementácii aplikácie bude potrebné otestovať možnosti vkladania objektov. Bude potrebné zistiť, do akej miery systém udržiava aj komplexnejšie objekty, odstrániť chyby a vylepšiť „správanie“ objektov.



Ponuka č. 3 – Model používateľa pre jeho identifikáciu

Motivácia – Každý jedinec je unikátny, prečo to nevyužiť?

Jedným z trendov dneška sú personalizované služby - od giganta Google po e-learningový nástroj Moodle sa mnohé prvky systémov prispôbujú konkrétnemu používateľovi. Podmienkou je samozrejme najskôr rozpoznať, o koho ide. Nebolo by výhodné automaticky identifikovať používateľa krátko po tom, ako začne používať systém? A čo keby sa takáto identifikácia využila na zvýšenie bezpečnosti systému? Čo keby v systéme prebiehalo dvojité overovanie, najskôr zadaním hesla a následne napísaním ľubovoľnej vety kvôli rozpoznaníu intervalu medzi jednotlivými stlačeniami kláves a rýchlosti uvoľnenia jednotlivých kláves?

Klikanie a pohyby kurzora myši, rýchlosť či rytmus písania na klávesnici sú parametre, ktoré si ľudia postupne pri používaní počítača vyvinú. Takisto robenie chýb pri písaní a ich počet, množina používaných softvérových prostriedkov, priemerné nastavenie hlasitosti v systéme daného používateľa či nastavenie jasu obrazovky, to všetko môže charakterizovať celkové správanie používateľa v systéme. Na tomto a ďalších sa dá postaviť rozlišovanie používateľa v systéme.

Myslíme si, že určovanie identity používateľa na základe jeho správania môže v niektorých prípadoch dosiahnuť vyššiu úroveň bezpečnosti ako klasická autorizácia heslom a priniesť nové výhody, pretože tak ako len málokto dokáže ovládnuť vlastný tep, len málokto dokáže dokonale maskovať svoje používateľské návyky a biometrické charakteristiky. Obzvlášť spoľahlivá môže byť takáto identifikácia používateľa v prípade, že tento o jej nasadení nevie, a teda nebude zámerne ovplyvňovať svoje správanie tak, aby sťažil svoju identifikáciu.

Tento spôsob identifikácie používateľa je z pohľadu frekvencie nasadzovania ďaleko za bežnejšími spôsobmi typu login-heslo. Identifikácia používateľa na základe biometrických vlastností má aj ďalšiu veľkú výhodu oproti klasickému heslu. Heslo možno zabudnúť, ale biometrické parametre nie. Našou motiváciou je práve možnosť rozvinúť zriedka využívané vytváranie modelu používateľa v systéme na základe biometrie a správania sa v systéme a nasadiť ho potenciálne aj do reálneho prostredia e-learningu na fakulte, prípadne zaviesť takýto softvér medzi všeobecne používané systémy. V prípade navrhnutia nástroja s dostatočnou spoľahlivosťou by sme tak vedeli ponúknuť alternatívu k drahším spôsobom identifikácie používateľov pomocou prídavného hardvéru.

Konceptuálne riešenie

Celému riešeniu bude predchádzať dobrá analýza problematiky sledovania jednotlivých biometrických charakteristík, ako aj nájdenie vhodného spôsobu ich sledovania. Ktoré charakteristiky sú najlepšie na vytvorenie čo najspoľahlivejšieho modelu používateľa pre jeho identifikáciu? Ako ich v systéme sledovať? Aké programové prostriedky pri tom využiť? Ako fungujú existujúce riešenia? Prihliadať pri tom budeme na oblasť predpokladaného nasadenia (webové prostredie? lokálny systém?), ako aj skúsenosti z oblasti biometriky.



Medzi charakteristiky, ktoré budeme sledovať, patria tempo a rytmus stláčania kláves na klávesnici, priemerná rýchlosť pohybu myši v čase, keď sa pohybuje, spôsob klikania, frekvencia používania jednotlivých programov daného používateľa, časové intervaly počas dňa, v ktorých je počítač používaný (v kombinácii s ostatnými sledovanými parametrami by tento údaj mal potvrdiť používateľa).

Sledovanie charakteristík správania používateľa bude od nášho riešenia vyžadovať nepretržité monitorovanie viacerých rozličných aspektov - sledovanie vstupných zariadení (myš, klávesnica, meranie časových intervalov medzi jednotlivými interakciami), prehľad spustených a spúšťaných procesov v systéme, aktuálnu dennú dobu, aktuálne nastavenie hlasitosti v systéme a pod. Preto bude riešenie smerovať k rezidentnému softvéru, ktorý bude schopný tieto činnosti vykonávať v reálnom čase.

Ako výsledky identifikácie pripravíme dáta na ďalšie použitie v ľubovoľnom systéme. Pre jednoduché prehliadnutie nazbieraných dát o identifikácii užívateľa pripravíme grafy, ktoré budú obsahovať namerané hodnoty parametrov v prehľadnej forme.

Aby bolo nami navrhnutý systém možné ďalej nasadiť v praxi, bude veľmi dôležité percentuálne určiť mieru správnej identifikácie používateľa na základe nami sledovaných parametrov. Jedine meraniami a vyhodnotením budeme môcť zmerať výsledok nami navrhnutého modelu na identifikáciu užívateľa. Medzi dôležité parametre, ktoré vypovedajú o celkovej spoľahlivosti systému, patria FAR a FRR. FAR je pravdepodobnosť, že systém prijme identifikovaného používateľa napriek tomu, že nie je určený správne. FRR je pravdepodobnosť, že systém odmietne, neidentifikuje používateľa napriek tomu, že parametre namerané súhlasia s jeho biometriou, čiže nesprávne odmietne používateľa, aj keď to bol naozaj on. Medzi ďalšie patrí napríklad FTC, čo je pravdepodobnosť, že systém nezaznamená biometrický vstup používateľa, napríklad stlačenie klávesy.

V našom riešení počítame s dostatočnou mierou modularity a interoperability, aby bolo možné vytváraný systém nasadiť v spolupráci s existujúcimi e-learningovými systémami.

Súčasťou riešenia tímového projektu je získanie vhodnej vzorky testovacích dát a otestovanie riešenia. Okrem možnosti tieto dáta umelo vytvoriť vidíme potenciál v oslovení širšej skupiny ľudí z nášho okolia (študenti FIIT, rodina, známi na rôznej úrovni používania PC) a získaní testovacích dát z reálneho prostredia. Testovanie a doladovanie nášho systému tak povedie k vyššej presnosti vytváraných modelov a identifikácie.



Zoradenie tém podľa priority

1. Tréner mentálnych schopností
2. Dizajn s použitím obohatenej reality
3. Model používateľa pre jeho identifikáciu

--- pre prípad, že nedostaneme preferované témy, zoradenie zvyšných tém podľa priority ---

4. Virtuálna FIIT
5. Portál pre časopis
6. Správa študentských projektov na fakulte
7. Vyhľadávanie a sprístupnenie citácií
8. Crowdsourcing verejných dát
9. Simulated Car Racing Competition 2011
10. Platforma pre realizovanie transakcií prostredníctvom mobilných zariadení
11. 3D grafická podpora vyhľadávania znalostí v dokumentoch
12. Prispôsobiteľný Widget
13. Tvorba rozvrhov
14. Objektové úložisko dát
15. Interaktívna vizualizácia grafových štruktúr v 3D priestore
16. Evolučný simulátor UI založený na heuristických pravidlách
17. Adaptívny proxy server
18. RoboCup tretí rozmer
19. Imagine Cup 2011: Game Design



Aktuálny rozvrh všetkých členov tímu pre zimný semester

		7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20	20-21
Pondelok	Juraj														
	Miroslav														
	Roman														
	Matej														
	Veronika														
Utorok	Andrej														
	Juraj														
	Miroslav														
	Roman														
	Matej														
Streda	Veronika														
	Andrej														
	Juraj														
	Miroslav														
	Roman														
Štvrtok	Matej														
	Veronika														
	Andrej														
	Juraj														
	Miroslav														
Piatok	Roman														
	Matej														
	Veronika														
	Andrej														
	Juraj														

Tím 05 – Androids – Dokumentácia k produktu

Vysvetlivky:

prednáška	iné aktivity
cvičenie	osobná preferencia
práca	stretnutia tímu





Záznamy zo stretnutí – časť B

V tejto časti dokumentu sa nachádzajú zápisnice z oficiálnych stretnutí tímu.



Zápis 1. stretnutia tímu Androids

Autor zápisu:	Bc. Juraj Belanji
Dátum:	29.09.2010
Čas:	18:00 – 20:00
Prítomní:	vedúci: Ing. Ivan Kapustík členovia tímu: Bc. Juraj Belanji Bc. Miroslav Hruška Bc. Roman Kováč Bc. Andrej Minárik Bc. Matej Moravanský Bc. Veronika Wolfová
Miestnosť:	softvérové štúdio
Téma:	organizácia tímu a úvod do problematiky RoboCup 3D

Priebeh:

- Stretnutie otvoril vedúci Ing. Ivan Kapustík. Hovoril o zápisnici a riadení tímu.
- Zápisnica má obsahovať konkrétne veci, teda kto dostal akú úlohu a čo sa konkrétne hovorilo.
- Zápisnica má byť obsiahlejšia a má obsahovať, čo sa nám páčilo alebo nepáčilo na návrhoch, aby neprišlo k opakovaniu diskusií.
- Bc. Veronika Wolfová položila otázku, ako monitorovať komunikáciu.
- Ing. Ivan Kapustík odpovedal, že je potrebné do dokumentácie zaradiť, aký spôsob komunikácie slúžil na aký účel.
- O rolách v tíme sa ešte bude hovoriť. Potrebné je urobiť si plán. Potrebné je pozerieť na požiadavky na stránke. Potrebné je dodržiavať termíny odovzdania.
- Diskusia ďalej pokračovala a hovorilo sa o metóde SCRUM. Prevažne sa diskutovalo, ktorá metóda je vhodnejšia pre riešenie problému našej témy. Členovia tímu usúdili, že sa pre náš problém hodí inkrementálny a iteratívny vývoj.
- Za vedúcu tímu bola zvolená Bc. Veronika Wolfová.
- Na webovej stránke tímu je potrebné, aby bolo:
 - úvodná stránka tímu (téma a meno vedúceho)
 - časť o tíme (o členoch)
 - časť s plánom
 - časť s dokumentáciou na stiahnutie:
 - zápisnice
 - zdrojové kódy
 - dokumentácia k programovej časti
 - ostatné dokumenty (prílohy a podobné)
 - časť s linkami (odkazy na iné tímy – domáce a zahraničné)
 - aktuality – môžu a nemusia byť
- RoboCup 3D robot (NAO robot) má 22 kĺbov.



- Ďalej sa diskutovalo o ligách v RoboCup-e, o obmedzeniach robota a samotných vlastnostiach robota.
- Ide o simulovaný robotický futbal. Robot má obmedzenia len na otáčanie kĺbov a rýchlosť ich otáčania, a to ako u človeka. Nemá žiadne silové obmedzenia. Informácie sa dostávajú zo servera (robot sa správa ako klient) a na základe tých informácií sa rozhoduje, čo má robiť s kĺbmi. Existuje editor na pohyby.
- Robot sa správa v súlade s fyzikálnymi zákonmi (ťažisko, rýchlosť pohybu,...).
- Keď robot nič nevidí, potrebuje sa otočiť a na základe senzorov sa ďalej pohybuje a reaguje. Kopnúť loptu je možné v každom smere, ktorý implementujeme.
- Na budúce stretnutie je zavolaný Ivan Hujsi – odborník z oblasti RoboCup 3D.
- Potrebné je dohodnúť sa, či tím bude vystupovať na IITSRC a TPCup.
- Bude sa pokračovať a nadväzovať na minuloročné tímy. Analyzovať sa budú viaceré tímy a nadväzovať sa bude iba na jeden. V prvom rade sa budú zlepšovať základné pohyby.
- Na budúcich stretnutiach:
 - čo sa riešilo do teraz
 - prečítajú sa úlohy
 - každý člen povie, čo ako urobil, či sa mu podarilo alebo nie
 - ďalej sa podľa plánu určia ďalšie úlohy

Nové úlohy:

ID	Riešiteľ	Opis	Dátum zadania	Termín
1.1	všetci	Preštudovať, ako fungujú roboty	29.09.2010	06.09.2010
1.2	všetci	Naštudovať zahraničné a domáce tímy a určiť, na ktorý tím budeme nadväzovať	29.09.2010	06.09.2010
1.3	všetci	Rozdeliť si roly v tíme	29.09.2010	06.09.2010



Zápis 2. stretnutia tímu Androids

Autor zápisu:	Bc. Veronika Wolfová
Dátum:	06.10.2010
Čas:	18:00 – 21:00
Prítomní:	vedúci: Ing. Ivan Kapustík členovia tímu: Bc. Juraj Belanji Bc. Miroslav Hruška Bc. Roman Kováč Bc. Andrej Minárik Bc. Matej Moravanský Bc. Veronika Wolfová ostatní: Bc. Ivan Hujsi
Miestnosť:	softvérové štúdio
Téma:	prednáška na tému RoboCup 3D, rozdelenie pozícií v tíme a tímov na analýzu

Priebeh:

- Stretnutie začalo prednáškou Bc. Ivana Hujsiho z tímu Critical Error.
- Tím Critical Error po svojich predchodcoch upravil tvorbu pohybov, výpočet pozície, základnú hraciu logiku a spravil refactoring kódu.
- Modul správania je oddelený od kódu, pozná 2 typy hráča a je vytvorený ako rozhodovací strom.
- Hráči tímu Critical Error majú obmedzené zorné pole a na výpočet pozície používajú 3 body, ktoré nie sú na priamke. Ak „nevidia“ 3 body, tak sa pozícia vypočítava pomocou gyroskopu a akcelerometra.
- Informácie o hráčovi pochádzajú z tela robota a sú v sférickej sústave.
- Pohyby sú uchovávané v XML, s ktorým vie pracovať editor pohybov.
- Pôvodne sa dokončenie pohybu kontrolovalo len podľa 1 kľbu, čo spôsobovalo problémy, takže teraz je to už riešené na základe času vypočítaného na vykonanie pohybu.
- Critical Error nadviazali na DreamTeam, RoboKopy nadviazali na Agenty 007.
- RoboKopy vstavali do Editoru správania podporné prostredie, ktoré im umožnilo robiť pomocné funkcie.
- Pohyb majú rovnaký, ale zorné pole majú neobmedzené (360°).
- Pohyby sú lepšie spravené u Critical Error a logika je lepšia u RoboKopov (riešená stavovým automatom).
- Ďalej sa hovorilo o hráčovi JIM (v Jave):
 - nový pohybový mechanizmus
 - tiež má oddelenú logiku a pohyb, ale je možné oboje meniť za behu
 - kontroluje sa čas na vykonanie pohybu a nie stav, v ktorom je kľb
 - rýchlosť pohybu (otáčania kľbu) je 7° za 1 cyklus
 - sú použité Kalmanove filtre na spresnenie odhadu polohy
 - gyroskop je menej presný
- Logy sa dajú prehrať na monitore (logy z roku 2009 nie).



- Logy z roku 2008 fungujú pre obe verzie servra (0.6.2 aj 0.6.3).
- Rozdiel medzi servery 0.6.2 a 0.6.3 je v tom, že prvý reprezentuje hráča ako 1 bod a druhý ako sústavu bodov (objekt), pričom táto komplexná informácia sa zatiaľ u nás moc nevyužíva.
- Z logu sa dajú získať pohyby, ale záznam prichádza každých 20 ms a v logu je vzorkovanie 200 ms.
- Vylepšiť by sme mohli rozhodovanie, pohyby, podporné prostriedky, prípadne nájsť využitie pre doteraz nevyužívané informácie.
- Chôdza môže byť rýchla alebo pomalá (presná), rôznymi smermi (napr. aj šikmo).
- Na náročnú simuláciu väčšieho zápasu by možno bolo možné využiť výkonný cluster, ktorý je na škole.
- Máme upravenú verziu servra, aby bolo možné beamovať hráča aj na súperovu polovicu ihriska.
- Pohyby je možné dynamicky ladit', existujú rôzne teórie.
- Pohyb je opísaný začiatkom, koncom, časom, uhlami, kĺbmi a rýchlosťou.
- Naše agenty zatiaľ nemajú implementované učenie.
- Server pri pohybe zavádza chyby.
- Učenie sa pomocou neurónovej siete je zbytočne zložitá.
- Materiály Ivana Hujsiho máme prístupné.
- Ďalej sme si rozdelili manažérske roly v tíme:
 1. Bc. Juraj Belanji – manažér plánovania, zástupca vedúcej
 2. Bc. Miroslav Hruška – manažér kvality
 3. Bc. Roman Kováč – manažér vývoja
 4. Bc. Andrej Minárik – manažér rizík
 5. Bc. Matej Moravanský – manažér podporných prostriedkov
 6. Bc. Veronika Wolfová – vedúca
- Rozdelili sme si tímy na podrobnú analýzu:
 1. Bc. Juraj Belanji – Agenty 007 + Kouretes
 2. Bc. Miroslav Hruška – Critical Error + Nao Team Humboldt
 3. Bc. Roman Kováč – RoboKit + FC Portugal
 4. Bc. Andrej Minárik – Hviezdna jedenástka + UT Austin Villa
 5. Bc. Matej Moravanský – RoboKopy + SEU-3D
 6. Bc. Veronika Wolfová – Neurotics + Nexus 3D
- Analýza má mať rozsah 2-3 strany A4 na tím, má sa odkazovať na študovanú dokumentáciu.
- Critical Error si má naštudovať každý člen tímu, keďže na nich nadviažeme.

Úlohy z minulých stretnutí:

ID	Riešiteľ	Opis	Zadané	Termín	Status
1.1	všetci	Preštudovať, ako fungujú roboty	29.09.2010	06.09.2010	Splnené
1.2	všetci	Naštudovať zahraničné a domáce tímy a určiť, na ktorý tím budeme nadväzovať	29.09.2010	06.09.2010	Splnené



1.3	všetci	Rozdeliť si roly v tíme	29.09.2010	06.09.2010	Splnené
-----	--------	-------------------------	------------	------------	---------

Nové úlohy:

ID	Riešiteľ	Opis	Zadané	Termín	Status
2.1	všetci	Vypracovať analýzu pridelených tímov	06.10.2010	13.10.2010	Zadané
2.2	Matej	Preskúmať podporné prostriedky pre organizáciu, komunikáciu a zdieľanie dokumentov v tíme	06.10.2010	13.10.2010	Zadané
2.3	Andrej	Zamyslieť sa nad rizikami tímového projektu	06.10.2010	13.10.2010	Zadané
2.4	Andrej	Vytvoriť web tímu	06.10.2010	18.10.2010	Zadané



Zápis 3. stretnutia tímu Androids

Autor zápisu:	Bc. Roman Kováč
Dátum:	13.10.2010
Čas:	18:00 – 21:00
Prítomní:	vedúci: Ing. Ivan Kapustík členovia tímu: Bc. Juraj Belanji Bc. Miroslav Hruška Bc. Roman Kováč Bc. Andrej Minárik Bc. Veronika Wolfová
Miestnosť:	softvérové štúdio
Téma:	analýza existujúcich riešení

Priebeh:

- Člen tímu Bc. Matej Moravanský sa z pracovných dôvodov nemohol zúčastniť stretnutia. Pedagogický vedúci tímu Ing. Ivan Kapustík požiadal o zaslanie mailu na jeho adresu v prípade, že sa niektorý z členov tímu nemôže zúčastniť stretnutia.
- Tím bol následne oboznámený s úlohami, ktoré určil garant predmetu a musia byť splnené do 5. týždňa semestra:
 - vytvorená HTML prezentácia tímu
 - zvolené nástroje na podporu manažmentu projektu, najmä manažment úloh
 - zvolené technológie, ktoré chce tím používať
- Boli spomenuté náležitosti, ktoré má webová stránka obsahovať (okrem základných)
 - je vhodné uviesť informácie o členoch tímu
 - v 5. týždni na nej musia byť 3 zápisnice
 - odkazy na naštudované materiály (ideálne zoradené podľa zaujímavosti, aktuálnosti a pod.)
- Štandardne má byť zápisnica uverejnená na stránke najneskôr na druhý deň od stretnutia.
- V 5. týždni semestra musí byť dostupný plán (minimálne základ).
- Stretnutie pokračovalo prezentovaním analýz tímov, ktoré boli pridelené jednotlivým členom tímu:
 - Bc. Juraj Belanji prezentoval analýzu tímov Agenty007 a Kouretes, a zároveň prezentoval aj analýzy tímov, ktoré boli pridelené Bc. Matejovi Moravanskému (RoboKopy a SEU-3D)
 - Bc. Veronika Wolfová prezentovala analýzu tímov Neurotics a Nexus 3D
 - Bc. Roman Kováč prezentoval analýzu tímov RoboKit a FC Portugal 3D
- Vytvorené analýzy tímov sa priložia k dokumentácií.
- Je vhodné vytvoriť si zakladač, kam sa budú postupne zakladať vytvorené dokumenty.
- Na stránke Ing. Ivana Kapustíka je dostupný odkaz na 2D tím 12. hráč, ktorý riešil stratégiu a bolo by vhodné z tohto tímu čerpať.
- V analýze je dôležitý aj jej záver, v ktorom tím uvedie, čo z toho využije a aké boli výhody a nevýhody jednotlivých riešení.



- Počas diskusie padla otázka ohľadom fungovania servera, z ktorej vyplynula úloha pre tím nastudovať a spustiť server.
- Nasledovala diskusia o pohybe hráča, z ktorej vyplynulo, že je vhodné dostať hráča do harmónie.
- Pri kopnutí lopty je dôležité, ako sa hráč k lopte postaví. Hráč kopne silnejšie, ak má podpornú nohu vedľa lopty ako keď ju má za loptou. Najsilnejšie hráč kopne loptu, keď je lopta vo vzduchu (platí pre človeka).
- Aktuálny robot nemá kameru na hlave, a tak je zbytočné otáčať hlavou (robot sa musí otočiť).
- V zásade je lepšie mať viacúrovňovú architektúru hráča, avšak je potrebné rátať so zvýšenou komunikáciou medzi jednotlivými vrstvami.
- Je vhodné mať model sveta, ktorý hovorí o aktuálnych pozíciách objektov v hre. Tieto pozície by si mal robot pamätať, aby mal informácie aj o objektoch, ktoré už nevidí.
- RoboCup ligy:
 - RoboCup Soccer
 - liga pre malé roboty
 - liga pre stredné roboty
 - liga pre štvornohé roboty
 - liga pre dvojnohé roboty
 - štandardizované (v súčasnosti Nao robot)
 - nezávislé
 - simulačné ligy
 - 2D – existuje tu aj súťaž vo vytváraní monitorov (sledovanie zápasu, komentovanie, ...)
 - 3D
 - RoboCup Rescue – orientácia v bludisku, pohybovanie sa poschodoch, ...
 - RoboCup Junior
- Na FIIT momentálne prebiehajú 2D a 3D súťaže. V 3D súťaži sa zatiaľ súťažilo len v pohyboch (ich rýchlosť a presnosť).
- Očakáva sa, že minimálne v letnom semestri sa bude riešiť aj vyššia logika hráča.
- Je potrebné vyhradiť si čas na ladenie hráča v letnom semestri.
- V zimnom semestri je potrebné spraviť prototyp hráča.
- Treba skontrolovať, či nám dostupný server dovolí umiestniť hráča na súperovu polovicu ihriska (štandardný to nepovoľuje).
- Pri výbere technológií je potrebné spraviť súhrn a o každej napísať pár slov ako zdôvodnenie jej výberu.
- Na záver stretnutia boli členom tímu ukázané vytvorené zakladače s dokumentáciou z predchádzajúcich rokov.



Úlohy z minulých stretnutí:

ID	Riešiteľ	Opis	Zadané	Termín	Status
2.1	všetci	Vypracovať analýzu pridelených tímov	06.10.2010	13.10.2010	Splnené
2.2	Matej	Preskúmať podporné prostriedky pre organizáciu, komunikáciu a zdieľanie dokumentov v tíme	06.10.2010	13.10.2010	Nesplnené
2.3	Andrej	Zamyslieť sa nad rizikami tímového projektu	06.10.2010	13.10.2010	Rozpracované
2.4	Andrej	Vytvoriť web tímu	06.10.2010	18.10.2010	Rozpracované

Nové úlohy:

ID	Riešiteľ	Opis	Zadané	Termín	Status
3.1	Juraj	Výber podporných prostriedkov	13.10.2010	20.10.2010	Zadané
3.2	Juraj	Vytvorenie plánu (základ)	13.10.2010	20.10.2010	Zadané
3.3	Roman	Identifikovanie technológií	13.10.2010	20.10.2010	Zadané
3.4	všetci	Naštudovanie servera	13.10.2010	20.10.2010	Zadané
3.5	všetci	Spustenie si zvoleného hráča	13.10.2010	20.10.2010	Zadané



Zápis 4. stretnutia tímu Androids

Autor zápisu:	Bc. Andrej Minárik
Dátum:	20. 10. 2010
Čas:	18:00 – 20:40
Prítomní:	vedúci: Ing. Ivan Kapustík členovia tímu: Bc. Miroslav Hruška Bc. Roman Kováč Bc. Andrej Minárik Bc. Matej Moravanský (na cca 20 min) Bc. Veronika Wolfová ostatní: Daniel Erban
Miestnosť:	softvérové štúdio
Téma:	personálne zmeny v tíme, analýzy zvyšných tímov, možnosti nadviazania na Critical Error, nová verzia simulačného servera, špecifikácia prototypu

Priebeh:

- Webstránka bola skontrolovaná vedúcim tímu - k spokojnosti, je vytvorená podľa pripomienok.

Personálne zmeny v tíme

- Vyvstala otázka, čo robiť v prípade odchodu člena tímu. Do počtu 5 členov je to ešte únosné (výpadok 1-2 členov je teda ešte OK).
- Matej prišiel oznámiť ukončenie svojho štúdia -> v tíme teda zostáva 5 ľudí.
- Zároveň sa naskytla možnosť spolupracovať s Danielom Erbanom, ktorý pracuje na RoboCup-e 3D v rámci bakalárskeho projektu.
- Prítomní (Veronika, Roman, Andrej, Ing. Kapustík; Miro meškal kvôli cvičeniu z Bezpečnosti počítačových systémov) sa zhodli, že sa môže zúčastňovať stretnutí a prípadne spolupracovať na niektorých častiach projektu (tieto časti budeme v práci uvádzať ako prevzaté).
- Daniela pridáme aj do Google skupiny a dotProject podporného nástroja, aby mal prehľad o dianí v tíme a mohol spolupracovať.
- Túto situáciu bude dobré *spomenúť aj v dokumentácii k riadeniu tímu*.
 - Čo s chýbajúcou manažérskou funkciou?

Dokončenie prezentovania analýz ďalších tímov

- Miro odprezentoval analýzu tímov Critical Error a NAO Team Humboldt.
- Učenie sa pohybov z logov zápasov (pracoval na tom tím Critical Error, konkrétne Ivan Hujši) je nepresné, zatiaľ sa ho nepodarilo s výhodou využiť (logy sú vzorkované po 200 ms – každý desiaty cyklus servera) – možnosť využitia je iba ako návrh pohybu, ako základný pohyb na dopracovanie (teda už vieme, že tadiaľto asi cesta nevedie).



- Nápad – zobrať agenta nejakého tímu, spustiť na našom serveri a nastaviť mu lepšie vzorkovanie. Dalo by sa takto niečo odpozorovať? -> Asi tiež len niečo ako návrh, ktorý je potrebné dopracovať. Zrejme takto nedostaneme hotový pohyb.
- Od Humboldtov – možnosť využiť editor XABSL, majú aj zaujímavú architektúru.
- Andrej odprezentoval tímy Hviezdna jedenástka a UT Austin Villa.

Critical Error a ďalšie nápady

- V prípade nadväzovania Critical Error – zvážiť možnosť pozmeniť XML, aby sa neriadilo rýchlosťou, ale časom (nekontrolovať skončenie pohybu; navrhol Ing. Kapustík). Čakať na kľby môže byť nepríjemné (šum zo servera) – prerobiť podobným spôsobom, ako to má JIM (natočenia kľbov sú nepresné, ale v čase sa orientujeme).
- Otázka nadväzovania na hráča – Critical Error vs. JIM? Máme na rozhodnutie max týždeň.
- Vývojové prostredie? Môže byť rôzne, podstatné je, aby sme to vedeli zjednotiť v SVN.
- Kontrola úloh: plán, podporný prostriedok, spustenie simulačného servera.

Nová verzia simulačného servera

- Vyskúšať sever 0.6.4, ktorá Mirovi beží bez problémov! (budeme pravdepodobne robiť hráča pre najnovšiu verziu, teda treba overiť, ako to funguje).
- Staršia a novšia verzia by sa mali dať inštalovať vedľa seba (premenné v cmd súboroch).
- Orientačne by bolo fajn pozrieť changelog novej verzie.

Špecifikácia prototypu do budúceho týždňa

- Špecifikácia, čo chceme vylepšiť a zahrnúť do prototypu – napr. zameniť pohyby riadené rýchlosťou za časové, editor správania (?), pohyby – pohyb vpred, možnosť navrhnúť stabilizované pohyby (čo máme po novom k dispozícii – už nie len akcelerometer, ale aj gyroskop – zistiť), práca s editorom pohybov.
- Na chodidlá robota pôsobí sila – aj na základe toho sa dá testovať, či robot stojí, ale napr. keď robot stúpa, hodnoty lietajú.
- Na realizáciu prototypu budú asi 3-4 týždne.
- Pri práci na špecifikácii – každý by mal mať aj nejaký *krátky výstupný dokument*.
 - Čo tam? Ako to vieme urobiť? Ako to overiť (akceptačné testy)? (-> toto mať budúci týždeň na stretnutie)

Rozhodnutie o voľbe hráča

- Výber hráča nakoniec padol na Critical Error, technológia C++.
- Členovia tímu majú za úlohu pozrieť si ho, spustiť.
- Pozor na novembrovú stredu (17.11.) – bude treba *dohodnúť náhradné stretnutie*.

**Úlohy z minulých stretnutí:**

ID	Riešiteľ	Opis	Zadané	Termín	Status
2.3	Andrej	Zamyslieť sa nad rizikami tímového projektu	06.10.2010	13.10.2010	Rozpracované
2.4	Andrej	Vytvoriť web tímu	06.10.2010	18.10.2010	Splnené
3.1	Juraj	Výber podporných prostriedkov	13.10.2010	20.10.2010	Splnené
3.2	Juraj	Vytvorenie plánu (základ)	13.10.2010	20.10.2010	Splnené
3.4	všetci	Naštudovanie servera	13.10.2010	20.10.2010	Rozpracované
3.5	všetci	Spustenie si zvoleného hráča	13.10.2010	20.10.2010	Rozpracované

Nové úlohy:

ID	Riešiteľ	Opis	Zadané	Termín	Status
4.1	Andrej	Zaradiť „externého člena“ do Google skupiny, dotProjectu	20.10.2010	25.10.2010	Zadané
4.2	Andrej	Dať na web link na Humboldt tím, editor XABSL a robota, ktorý je k dispozícii (3 linky)	20.10.2010	25.10.2010	Zadané
4.3	Juraj	Doplniť do plánu presné termíny odovzdania, zvýrazniť ich	20.10.2010	25.10.2010	Zadané
4.4	Andrej	Nahrať upravený plán na webstránku tímu	20.10.2010	25.10.2010	Zadané
4.5	Miro	Skúsiť spustiť logy z minulých rokov na 0.6.4 a dať vedieť ostatným, či to funguje	20.10.2010	27.10.2010	Zadané
4.6	všetci (Veronika)	Špecifikácia prototypu na základe výberu hráča spolu s predstavami ako nápady zrealizovať	20.10.2010	27.10.2010	Zadané
4.7	Roman, Miro, Dano	Pozrieť sa na verziovací systém (Team system? SVN? Ako to majú Critical Error?)	20.10.2010	27.10.2010	Zadané



Zápis 5. stretnutia tímu Androids

Autor zápisu:	Bc. Juraj Belanji
Dátum:	27.10.2010
Čas:	18:00 – 20:45
Prítomní:	vedúci: Ing. Ivan Kapustík členovia tímu: Bc. Juraj Belanji Bc. Miroslav Hruška (30 min meškal) Bc. Roman Kováč Bc. Andrej Minárik Bc. Veronika Wolfová ostatní: Daniel Erban (40 min meškal)
Miestnosť:	softvérové štúdio
Téma:	špecifikácia požiadaviek na prototyp, diskusia o hrubom návrhu, dohovor ohľadom prvého kontrolného odovzdania dokumentácie

Priebeh:

- Ing. Kapustík prečítal mail od prof. Bielikovej o pokynoch k odovzdaniu v prvej etape; každý tím má v čase stretnutia odovzdať dokumentáciu, a to k produktu (inžinierske dielo) a k riadeniu. Budúci týždeň sa odovzdáva.
- Do dokumentácie treba pridať ešte špecifikáciu a návrh prototypu.
- Ing. Kapustík hovoril o softvérových technológiách a priniesol vytlačený papier zo stránky (SwTech.com). Vystala otázka, či je testovanie softvérová technológia. Uzavrelo sa, že technológia by mala mať formálny opis (z toho hľadiska napríklad JUnit je softvérová technológia na testovanie, ale samotné testovanie nie je).
- Potrebné je upraviť literatúru tak, aby sa odkazy na stránky uvádzali pod čiarou (ako footnote) a nie v literatúre. Ak je to však odkaz priamo na nejaký PDF dokument (nie manuál, tutoriál a pod.), tak môže byť v literatúre.
- Andrej rekapituloval, o čom sa na minulom stretnutí hovorilo.
- Ing. Kapustík hovoril o 0.6.4 verzii servera, že sa dohodol s diplomantmi, aby upravili zdrojové kódy tak, aby sa dalo pokračovať na novej verzii (0.6.4) servera.
- Andrej má vytvoriť dokument o rizikách tímového projektu pre dokumentáciu k riadeniu.
- Definitívne sa potvrdilo a uzavrelo rozhodnutie na základe e-mail komunikácie, že budeme pokračovať na vývoji robota JIM (vytvorený v Java).
- Väčšine členov tímu sa spustenie servera podarilo, aj keď prebiehalo s chybami (externý monitor zhadzoval server). Tento problém sa vyriešil zmenením premennej `$enableInternalMonitor` v súbore `simspark.rb` z `false` na `true`, vďaka čomu sa monitor spúšťa interne.
- Na web stránku boli úspešne pridané nové linky a plán.
- Miro hovoril o changelogoch k novému 0.6.4 serveru.



Špecifikácia požiadaviek

- Každý člen tímu povedal niečo o špecifikácii požiadaviek.
- Špecifikácia sa následne rozvinie do návrhu. Je vhodné navrhnúť toho viac, ale nemusí sa potom dorobiť všetko. Potrebné je len v časti implementácie napísať, ktoré časti sa budú implementovať a ktoré nie (môže to byť ako práca do budúcnosti). Veronika spomenula šesť okruhov špecifikácie, ktoré sme identifikovali:
 - Zlepšenie pohybov
 - Vylepšenie editora pohybov
 - Migrácia na nový server – eventuálna automatizácia migrácie (vznikla otázka, či tento proces nie je príliš náročný na plnú automatizáciu – najkritickejšia vec v tomto ohľade je komunikácia so serverom; potrebné je, aby robot (agent), vytvorený na staršej verzii servera, vedel komunikovať aj s novšou)
 - Futbalová logika a stratégia
 - XABSL – jazyk pre vyššiu logiku správania
 - Testovací framework pre pohyby hráča (prípadne aj logiku)
- Ing. Kapustík hovoril o tom, čo to je prototyp v našom prípade. V našom prípade je prototyp nejaký návrh modelu. Sú problémy, ktoré nevieme dostatočne modelovať (povedzme pre stratégiu) – je vhodné urobiť prototyp z hľadiska vytvorenia dobrého modelu komunikácie medzi modulmi (model problému = prototyp). Toto znamená, že výsledkom zimného semestra nemusí byť prototyp hráča robota, ale môže to byť aj len model.
- Juraj hovoril o prototypu editora pre vyššiu logiku, eventuálne spojenie myšlienky práce s vyššou logikou pomocou produkčného systému a jazyka XABSL.
- Veronika ďalej pokračovala v časti špecifikácie o zlepšovaní pohybov. Vstávanie je teraz vytvorené pre obe východzie polohy (z brucha aj z chrbta). Dalo by sa vylepšiť a prirobiť vstávanie, ktoré by sa nespoliehalo na možnosť ťahať nohy po zemi (bolo by použiteľné aj pre fyzické roboty).
- Potrebné je do dokumentácie doplniť, že sa multiagentové systémy s n agentmi správajú ako multiagentové systémy s $n+1$ agentmi – n robotov a navyše prostredie – teda pri učení robota potrebujeme zapojiť aj prostredie a využívať výhody, ktoré nám ono poskytuje.
- Ing. Kapustík oznámil, že v miestnosti bc300 je každý pondelok o 14:00 prednáška prof. Kvasničku o umelej inteligencii – má tam prísť aj hosťujúca prednášajúca z Košíc, ktorá bude hovoriť o robote NAO.
- Veronika ďalej hovorila o chôdzi robota. Špecifikovala viaceré typy chôdze podľa smeru – dopredu, do strany, šikmo, a podľa rýchlosti a presnosti – pomalú (presnú) a rýchlu chôdzu (prípadne beh). K pohybu otáčania vznikol nápad vyskúšať implementovať vojenské otáčanie!!!
- Potrebujeme do špecifikácie pridať aj niečo o stabilizácii robota – framework pre tvorbu stabilných pohybov – padol návrh v editore určiť ťažisko a podľa neho určovať stabilitu (využiť teóriu ZMP a prípadne do editora pridať možnosť otáčania modelu robota okolo fixného bodu).
- Bolo by vhodné prevziať kopanie od tímu Critical Error a prispôsobiť ho pre robota JIM. Potrebujeme sa rozhodnúť, aké kopy chceme, aby robot vedel uskutočňovať, a to vzhľadom na silu kopu, presnosť kopu, či si môžeme dovoliť, aby robot spadol, a pod.



- U brankára by bolo najvodnejšie vytvoriť oba používané spôsoby bránenia – pádom na bok aj rozkročením sa a dosadnutím.
- Juraj hovoril o úprave editora pohybov, pridaní joint (spojených) kĺbov do editora, nejakej úprave XML súboru a zrkadlovom či symetrickom pohybe kĺbov. Ďalej hovoril o produkčnom systéme, ktorý by sa dal implementovať do editora pohybov (bol by to potom editor pohybov a správania robota).
- Andrej hovoril o metrikách pre overovanie stability a rýchlosti chôdze – uzavrel, že by bolo vhodné si to porovnávať s tímami na RoboCup stránke FIIT.
- Andrej ďalej spomenul využitie gyroskopu (GyroRatePerceptor) a akcelerometra – potrebujeme zistiť, ako to presne pracuje a prípadne k tomu vytvoriť nejaký dokument (ako to využiť).
- Miro oznámil, že agent vypracovaný pre server 0.6.3 sa dá bez problémov a úpravy spustiť aj na 0.6.4 serveri.
- Andrej pokračoval v špecifikácii a hovoril o vyššej logike (XABSL), kde sa správanie rozpracováva cez stavový automat (tak sa dá zadefinovať vyššie správanie, povedzme „zabrán prihrávke“).
- Ing. Kapustík prevzal slovo a hovoril o tom, čo by bolo vhodné urobiť v rámci prototypu: bolo by vhodné otestovať, či agent môže zistiť, ako je súper otočený, či agent vie odhadnúť, či sa mu podarí dobehnúť k lopte a zabrániť prihrávke... To by sme mali zahrnúť a overiť v prototypu. Na základe konkrétnych hodnôt (čo sa rozhodol agent urobiť) je potrebné urobiť nejaké ohodnotenie. JIM má implementovaný Kalmanov filter. Táto skutočnosť by sa tiež dala využiť na určovanie rýchlosti a smeru pohybu (po nejakú malú rýchlosť sa budeme tváriť, že hráč stojí, pre väčšiu, že sa pohybuje). Tieto skutočnosti sú späté s predpovedacím modulom, ktorý by mohol pomôcť pri riadení správania robota (pozor na mieru prispôsobovania – ak sa robot stále bude vyhábať nárazu, zistíme, že je to neprípustné, čiže potrebujeme určiť mieru vyhábania).
- Roman sa vrátil k stratégii a vyšším schopnostiam. Agent musí vedieť plánovať činnosť, musí vedieť vyhodnotiť situáciu a musí sa vedieť rozhodnúť, či bude útočiť alebo brániť. Taktiež potrebuje najprv zistiť u koho je lopta. Robot potrebuje poznať pozíciu svojich spoluhráčov a súperov, aby mohol na základe toho konať (pridať pamätanie minulosti a odhadovanie budúcich pozícií). Je nutné, aby dôveryhodnosť informácií z minulosti klesala v priebehu času, aby bolo možné určiť, nakoľko je informácia o polohe hráčov, ktorých nevidí (problém, že je kamera v tele), správna.
- Vystala otázka, či je povolená komunikácia medzi agentmi (posielanie správ). Hráči si medzi sebou nesmú posielat' informácie, teda priama komunikácia nie je možná – riešenie je to, že server už teraz poskytuje rozsiahlejšie informácie o tom, čo sa okolo agenta deje.
- Potrebujeme navrhnuť overenie (metodiky overovania) konkrétnych vecí – zamyslieť sa nad tým, čo potrebujeme zistiť, aký pomer úspešných pokusov stačí (v špecifikácii potrebujeme uviesť nie len ako to chceme mať, ale aj ako to overíme).
- Miro ešte hovoril o serveri 0.6.4, že je migrácia zo servera 0.6.3 vybavená bez úprav.
- Miro spomenul niečo ohľadom testovacieho frameworku. Ak by sa vytvoril testovací framework, ktorý by rýchlo pracoval, tak by sme mohli na doladenie pohybov využiť aj nejakú formu umelej inteligencie, teda učenie. Potrebujeme zistiť, ako získať informácie o



učení a ako sa dá trénovať (navyše potrebujeme učenie aj logovať, aby sme potom mohli prezerat' priebeh učenia a vyberat' dobré pohyby).

- Na verziovanie zdrojových kódov sme sa rozhodli použiť SVN.

Úlohy z minulých stretnutí:

ID	Riešiteľ	Opis	Zadané	Termín	Status
2.3	Andrej	Zamysliet' sa nad rizikami tímového projektu	06.10.2010	13.10.2010	Splnené
3.4	všetci	Naštudovanie servera	13.10.2010	20.10.2010	Splnené
3.6	všetci	Spustenie si zvoleného hráča	13.10.2010	20.10.2010	Splnené
4.1	Andrej	Zaradiť externého člena do Google skupiny, dotProjectu	20.10.2010	25.10.2010	Splnené
4.2	Andrej	Dať na web link na Humboldt tím, editor XABSL a robota, ktorý je k dispozícii (3 linky)	20.10.2010	25.10.2010	Splnené
4.3	Juraj	Doplniť do plánu presné termíny odovzdania, zvýrazniť ich	20.10.2010	25.10.2010	Splnené
4.4	Andrej	Nahrať upravený plán na web stránku tímu	20.10.2010	25.10.2010	Splnené
4.5	Miro	Skúsiť spustiť logy z minulých rokov na 0.6.4 a dať vedieť ostatným, či to funguje	20.10.2010	27.10.2010	Rozpracované
4.6	všetci (Veronika)	Špecifikácia prototypu na základe výberu hráča spolu s predstavami, ako nápady zrealizovať	20.10.2010	27.10.2010	Splnené
4.7	Roman, Miro, Daniel	Pozrieť sa na verziovací systém (Team system? SVN? Ako to majú Critical Error?)	20.10.2010	27.10.2010	Splnené

Nové úlohy:

ID	Riešiteľ	Opis	Zadané	Termín	Status
5.1	všetci	Upraviť dokumenty k špecifikácii a návrhu a poslať ich Jurovi	27.10.2010	01.11.2010	Zadané
5.2	Juraj	Doplniť do dokumentácie niečo o multiagentových systémoch a stabilizácii hráčov	27.10.2010	01.11.2010	Zadané
5.3	Andrej	Vytvoriť dokument o gyroskope a poslať Jurovi	27.10.2010	02.11.2010	Zadané
5.4	Juraj	Zoskupiť dokumentáciu o produkte	27.10.2010	02.11.2010	Zadané
5.5	Veronika	Skontrolovať dokumentáciu o produkte	27.10.2010	02.11.2010	Zadané
5.6	Roman	Vytvoriť dokument o SVN a poslať Jurovi	27.10.2010	01.11.2010	Zadané
5.7	Andrej	Vytvoriť dokument k analýze rizík projektu a poslať Jurovi	27.10.2010	02.11.2010	Zadané
5.8	Miro	Vytvoriť hrubý návrh štábnej kultúry (pomenúvanie premenných, nejaké štandardy písania kódu, hrubý návrh metodiky vývoja)	27.10.2010	01.11.2010	Zadané
5.9	Juraj	Zoskupiť dokumentáciu o riadení	27.10.2010	02.11.2010	Zadané
5.10	Veronika	Skontrolovať dokumentáciu o riadení a vytvoriť preberací protokol	27.10.2010	02.11.2010	Zadané



5.11	Daniel	Vytlačiť dokumentáciu (Veronika mu ju pošle)	27.10.2010	03.11.2010	Zadané
5.12	Andrej	Nahrať dokumentáciu na web (Veronika mu ju pošle)	27.10.2010	03.11.2010	Zadané



Zápis 6. stretnutia tímu Androids

Autor zápisu:	Bc. Veronika Wolfová
Dátum:	03.11.2010
Čas:	18:00 – 19:10
Prítomní:	vedúci: Ing. Ivan Kapustík členovia tímu: Bc. Juraj Belanji Bc. Miroslav Hruška (30 min. meškal) Bc. Andrej Minárik Bc. Veronika Wolfová
Miestnosť:	softvérové štúdio
Téma:	odovzdanie dokumentácie, určenie ďalšieho postupu

Priebeh:

- Roman sa na stretnutie zo zdravotných dôvodov nedostavil, pričom sa dopredu korektne ospravedlnil.
- Odovzdali sme dokumentáciu k produktu a k riadeniu pre prvú fázu tímového projektu.
- Po zbežnom prezretí odovzdanej dokumentácie zatiaľ jedinou pripomienkou zo strany vedúceho, Ing. Ivana Kapustíka, bolo, že sme síce pod čiaru uviedli odkazy na stránky, z ktorých sme čerpali informácie, ale do použitej literatúry sme uviedli len jeden manuál, pričom tam v skutočnosti mali byť vymenované jednotlivé konkrétne publikácie zo spomínaných stránok, z ktorých sme na čerpali.
- Juraj dostal za úlohu upraviť použitú literatúru a odkazy na ňu podľa pripomienok.
- Ing. Ivan Kapustík nám priniesol vytlačенú informáciu o plánovanej prednáške Ing. Márie Virčíkovej z Technickej Univerzity v Košiciach, ktorá je plánovaná na pondelok 8.11.2010 o 14:00 v miestnosti bc300 a mala by byť zameraná na robotiku a umelú inteligenciu, pričom inžinierka Virčíková pravdepodobne prinesie fyzického robota NAO. Tejto prednáške by sme sa mali zúčastniť, ak je to aspoň trochu možné.
- Ďalej sme si prešli úlohy z minulého stretnutia, ktoré sme všetky splnili okrem úlohy vytvoriť návrh štábnej kultúry a pridať ho do dokumentácie k riadeniu. Túto úlohu sme odložili na neskôr, lebo súvisí s tvorbou metodiky v predmete MSI. V ďalšej odovzdávanej verzii dokumentácie k riadeniu už bude vypracovaná metodika zahrnutá, pričom hotová by mala byť v priebehu najbližších dvoch týždňov.
- Narazili sme na vážny problém s editorom pohybov, ktorý budeme musieť konzultovať v rámci úspory času a rýchlosti riešenia s našimi predchodcami. Dostupné verzie editora, ktoré sme našli, nie sú schopné spracovávať už vytvorené XML súbory pohybov. Problém je buď vo verzii/nesprávnom postupe inštalácie editora, alebo v samotných XML súboroch.
- Ďalej sme prebrali návrh prototypu, z ktorého vyplynuli úlohy na najbližší týždeň (sú uvedené v tabuľke úloh).
- Keďže kolízia cvičení a tímových stretnutí spôsobuje Mirove meškania na stretnutia, čo nie je žiaduce, rozhodli sme sa pokusne posunúť začiatok stretnutia v stredu 10.11.2010 na 18:30.



- Stretnutie zo stredy 17.11.2010, kedy je sviatok, sme predbežne presunuli na 16.11.2010 v čase 17:00-19:00.
- Dospeli sme k záveru, že ak chceme pracovať na automatickom testovacom frameworku (trénerovi), bude nutné zistiť, ako a kde presne treba zmeniť fyziku a ďalšie nastavenia v serveri a hráčovi, aby bolo možné pracovať so zrýchlenou simuláciou – toto si zobral na starosť do budúceho týždňa Miro a spolupracovať bude s Romanom.
- Predpokladáme problémy pri definovaní opisu presného postupu overenia výsledkov testov.
- Treba zistiť, ako hráč JIM v pohybovom module určuje svoj stav, teda či stojí alebo spadol, v akej pozícii sú jednotlivé kĺby, atď.
- Pravdepodobne by bolo vhodné používať dva záznamy – prvým je log servera (každých 200 ms) a druhým sú samotné správy zo servera (každých 20 ms).
- Treba zistiť, ako by sme dokázali zadať serveru presnú situáciu, z ktorej chceme vychádzať – rozmiestnenie hráčov, polohu/pohyb lopty a hráčov, atď.
- Ďalšiu prácu na projekte sme si približne rozdelili tak, že Veronika a Andrej budú pracovať na pohyboch hráča, Juraj na vylepšovaní editora a Miro s Romanom na testovacom frameworku.

Úlohy z minulých stretnutí:

ID	Riešiteľ	Opis	Zadané	Termín	Status
4.5	Miro	Skúsiť spustiť logy z minulých rokov na 0.6.4 a dať vedieť ostatným, či to funguje	20.10.2010	27.10.2010	Splnené
5.1	všetci	Upraviť dokumenty k špecifikácii a návrhu a poslať ich Jurovi	27.10.2010	01.11.2010	Splnené
5.2	Juraj	Doplniť do dokumentácie niečo o multiagentových systémoch a stabilizácii hráčov	27.10.2010	01.11.2010	Splnené
5.3	Andrej	Vytvoriť dokument o gyroskope a poslať Jurovi	27.10.2010	02.11.2010	Splnené
5.4	Juraj	Zoskupiť dokumentáciu o produkte	27.10.2010	02.11.2010	Splnené
5.5	Veronika	Skontrolovať dokumentáciu o produkte	27.10.2010	02.11.2010	Splnené
5.6	Roman	Vytvoriť dokument o SVN a poslať Jurovi	27.10.2010	01.11.2010	Splnené
5.7	Andrej	Vytvoriť dokument k analýze rizík projektu a poslať Jurovi	27.10.2010	02.11.2010	Splnené
5.8	Miro	Vytvoriť hrubý návrh štábnej kultúry (pomenúvanie premenných, nejaké štandardy písania kódu, hrubý návrh metodiky vývoja)	27.10.2010	01.11.2010	Rozpracované
5.9	Juraj	Zoskupiť dokumentáciu o riadení	27.10.2010	02.11.2010	Splnené
5.10	Veronika	Skontrolovať dokumentáciu o riadení a vytvoriť preberací protokol	27.10.2010	02.11.2010	Splnené
5.11	Daniel	Vytlačiť dokumentáciu (Veronika mu ju pošle)	27.10.2010	03.11.2010	Zrušené
5.12	Andrej	Nahrať dokumentáciu na web (Veronika mu ju pošle)	27.10.2010	03.11.2010	Splnené

**Nové úlohy:**

ID	Riešiteľ	Opis	Zadané	Termín	Status
6.1	Juraj	Upraviť odkazy na literatúru v dokumentácii a zoznam použitej literatúry	03.11.2010	10.11.2010	Zadané
6.2	Andrej, Veronika	Získať najaktuálnejšiu a hlavne plne funkčnú verziu editora pohybov, nainštalovať si ju a preskúmať aktuálne pohyby hráča JIM a agenta tímu RoboKit	03.11.2010	10.11.2010	Zadané
6.3	Roman, Miro	Aké nastavenia a zmeny je možné urobiť vo fyzike servera a hráča, ako a kde treba tieto zmeny v jednotlivých súboroch realizovať, čo všetko je možné ovplyvniť – získané poznatky spísať prehľadne do dokumentu, aby boli dostupné pre všetkých	03.11.2010	10.11.2010	Zadané
6.4	Juraj	Previazanie klbov v editore	03.11.2010	24.11.2010	Zadané



Zápis 7. stretnutia tímu Androids

Autor zápisu:	Bc. Miroslav Hruška
Dátum:	10.11.2010
Čas:	18:30 – 20:00
Prítomní:	vedúci: Ing. Ivan Kapustík členovia tímu: Bc. Juraj Belanji Bc. Miroslav Hruška Bc. Roman Kováč Bc. Andrej Minárik Bc. Veronika Wolfová
Miestnosť:	softvérové štúdio
Téma:	Progres v rámci zadaných úloh.

Priebeh:

- Stretnutie začalo rozpravou o avizovanej prednáške v súvislosti s RoboCup, ktorá sa konala v pondelok 8. 11. 2010.
- Následne bolo prediskutované stretnutie, ktoré sme mali s inžinierom Bartalosom ohľadom použitých technológií v našom projekte. Uvedené stretnutie sa konalo dňa 9. 11. 2010. Zo stretnutia bol vyvodený nasledovný záver: nakoľko projekt RoboCup3D je špeciálnym typom projektu, dôraz sa bude klásť hlavne na úroveň zdrojového kódu, JUnit testov a dokumentácie.
- Nasledovalo vyhodnotenie úloh zadaných v minulom týždni.
- Veronika s Andrejom prezentovali problémy spojené s editorom pohybov. Vzhľadom na nekompatibilitu vznikla potreba riešiť úlohu transformácie XML súboru týkajúceho sa pohybu. Andrej navrhol, že nástroj pre konverziu urobí.
- Dohodli sme sa, že každý bude mať do budúceho týždňa JIM-a v takej forme, aby ho mohol spúšťať z vývojového prostredia.
- Miro zhodnotil možnosti zmeny servera pre potreby testovacieho frameworku. Zistené výsledky sú pozitívne pre vývoj frameworku.
- Roman hovoril o spôsobe, akým sú vytvárané logy. Logy servera sú rovnaké pre verziu 0.6.3 a 0.6.4.
- Inžinier Kapustík navrhol kontaktovať Ivana Hujsiho v otázke parsovania logov zo servera.
- Rozprávali sme sa o spôsoboch, akými môže Juraj pokračovať vo vývoji editora. Naznačil, že môže byť problém vytvoriť netriviálnu previazanosť medzi klbmi.
- Zhodli sme sa, že s vysokou pravdepodobnosťou nebudeme pokračovať na tvorbe vyššej logiky s použitím jazyka XABSL. Juraj navrhol použitie produkčného systému. Inžinier Kapustík rozprával o produkčnom systéme, ktorý používali pri riešení RoboCup 2D.
- Rozdelili sme si úlohy.



Úlohy z minulých stretnutí:

ID	Riešiteľ	Opis	Zadané	Termín	Status
5.8	Miro	Vytvoriť hrubý návrh štabnej kultúry (pomenúvanie premenných, nejaké štandardy písania kódu, hrubý návrh metodiky vývoja)	27. 10. 2010	01. 11. 2010	Rozpracované
6.1	Juraj	Upraviť odkazy na literatúru v dokumentácii a zoznam použitej literatúry	03.11.2010	10. 11. 2010	Splnené
6.2	Andrej, Veronika	Získať najaktuálnejšiu a hlavne plne funkčnú verziu editora pohybov, nainštalovať si ju a preskúmať aktuálne pohyby hráča JIM a agenta tímu RoboKit.	03.11.2010	10. 11. 2010	Splnené
6.3	Roman, Miro	Aké nastavenia a zmeny je možné urobiť vo fyzike servera a hráča, ako a kde treba tieto zmeny v jednotlivých súboroch realizovať, čo všetko je možné ovplyvniť – získané poznatky spísať prehľadne do dokumentu, aby boli dostupné pre všetkých	03.11.2010	10. 11. 2010	Splnené
6.4	Juraj	Previazanie kľbov v editore	03.11.2010	24. 11. 2010	Rozpracované

Nové úlohy:

ID	Riešiteľ	Opis	Zadané	Termín	Status
7.1	Všetci	Mať JIM-a v skompilovateľnom stave.	10.11.2010	16.11.2010	Zadané
7.2	Juraj, Andrej	Umiestniť dokumentáciu k riadeniu na webový server.	10.11.2010	16.11.2010	Zadané
7.3	Andrej	Analyzovať potrebnú konverziu medzi rôznymi XML formátmi, prípadná implementácia riešenia.	10.11.2010	16.11.2010	Zadané
7.4	Juraj	Pokračovať v práci na editore pohybov, konkrétne v tvorbe symetrických pohybov.	10.11.2010	16.11.2010	Zadané
7.5	Roman, Miro	Navrhnuť architektúru a triedy testovacieho frameworku. Špecifikovať požadovanú funkcionálnu z dôvodu overenia návrhu systému.	10.11.2010	16.11.2010	Zadané
7.6	Veronika	Vybrať si a vytvoriť jeden konkrétny pohyb a odovzdať získané skúsenosti ostatným.	10.11.2010	16.11.2010	Zadané
7.7	Roman	Vytvoriť repozitár na SourceForge pre projekty.	10.11.2010	16.11.2010	Zadané



Zápis 8. stretnutia tímu Androids

Autor zápisu:	Bc. Roman Kováč
Dátum:	16.11.2010
Čas:	17:00 – 18:00
Prítomní:	vedúci: Ing. Ivan Kapustík členovia tímu: Bc. Juraj Belanji Bc. Miroslav Hruška Bc. Roman Kováč Bc. Andrej Minárik Bc. Veronika Wolfová
Miestnosť:	softvérové štúdio
Téma:	Progres v rámci zadaných úloh.

Priebeh:

- Stretnutie začalo zhrnutím predchádzajúceho stretnutia a následne pokračovalo prechádzaním stavu rozpracovaných úloh.
- Juraj, ako bolo vopred dohodnuté, bude pracovať na úlohe previazania klbov v editore.
- JIM-a sa podarilo spustiť všetkým členom tímu. Pre tento účel vytvoril Roman návod, ako hráča spustiť a urobil v Jimovi zmeny, aby bol ľahko spustiteľný.
- Juraj s Andrejom umiestnili dokumentáciu k riadeniu na web tímu.
- Andrej referoval o stave úlohy konverzie XML dokumentu, ktorý generuje editor pohybov, do nového formátu:
 - Andrej vytvorí konvertor, ktorý následne Juraj použije v editore pohybov pridaním tlačidla na spustenie konvertora.
 - Andrej spomenul, že zatiaľ analyzuje pôvodné XML súbory. Ing. Kapustík reagoval, že editor pohybov generuje XML súbory jednoznačne a nemali by byť s ich analýzou problémy.
 - Veronika pripomenula absenciu trvania pohybu a bude potrebné zistiť, ako nastaviť túto hodnotu v novom XML formáte (zatiaľ to riešila ručne).
 - Miro po rozhovore s členom druhého tímu, ktorý sa rovnako zaoberá RoboCup-om, zistil, že aj oni riešia rovnakú úlohu.
- Miro a Roman referovali o stave testovacieho frameworku:
 - Roman prezentoval architektúru spracovania logu. Ing. Kapustík upozornil, že je potrebné brať do úvahy dva rôzne časy - čas hry a druhý čas, ktorý nezávisí od toho, či je hra v móde PlayOn.
 - Miro prišiel k záveru, že opis situácie pre testovací framework nie je triviálny a bude potrebné tento opis reprezentovať skriptom.
 - Pre tento účel vytvoril skript v jazyku Lisp pre prihrávku.
 - Spomenul príklad, kedy nie je potrebné čakať na ukončenie celej situácie, ak sa vyhodnotí, že nespĺňa očakávané podmienky. Napríklad, keď lopta má zlý smer, nie je nutné čakať kým lopta zastaví.



- Algoritmus pre vyhodnotenie situácie bude musieť vracat' boolean hodnotu, ktorá určí, či test bol úspešný. Navyiac je potrebné vracat' aj hodnotu fitness, ktorá určí, v akej miere bol test úspešný.
- Mal by existovať spôsob, ako bude možné JIM-ovi povedať, čo má robiť, za účelom testovania situácie.
- Umelou inteligenciou sa na teraz nebudeme zaoberať, keďže sa dotýka aj vyšších schopností hráča.
- Ing. Kapustík reagoval na spomenuté skutočnosti. Spomenul, že tréner by mohol hráča umiestniť na ihrisko a sám hráč sa rozhodne, čo má robiť. Druhou možnosťou je upraviť priamo logiku hráča trénerom.
- Dospelo sa k názoru, že ideálny spôsob bude dodať hráčovi parametre, prípadne súbor s inicializáciou situácie.
- Ing. Kapustík navrhol ako voliteľný parameter pre inicializáciu testu presnosť, ktorá zabezpečí, že tréner nebude opakovať vždy rovnakú situáciu (napr. umiestni loptu na iné miesto).
- Ing. Kapustík navrhol ako ďalší voliteľný parameter hodnotu, ktorá určí, po akom čase sa testovanie začne vykonávať. Ďalším parametrom by mohol byť čas, do ktorého sa má testovanie vykonávať.
- Ing. Kapustík spomenul, že by bolo vhodné mať možnosť analyzovať situáciu aj bez testov, čo by mohlo mať zaujímavé využitie (napr. učenie sa agenta).
- Veronika referovala o stave úlohy vytvorenia konkrétneho pohybu:
 - Spomenula, že je potrebné upravovať súbor plan.rb, aby bolo možné pohyb spustiť.
 - Navyše musí byť hra v stave PlayOn.
 - Spomenula problém, ktorý nastane po skončení polčasu, kedy je hráč znova nabeamovaný na ihrisko a vykonávanie pohybu je prerušené.
 - Ing. Kapustík navrhol pozrieť sa na spôsob, ako vytvoriť nekonečný polčas.
 - Pomocou editora pohybov a úpravou vytvorených XML súborov vytvorila pohyb otáčania sa hráča, ktorý je stabilný a dokáže otočiť hráča približne o 75°, ale treba ho ešte trochu doladiť.
 - Vytvorila aj pohyb padania brankára nabok, pričom brankár zostane ležať naboku avšak občas padne pod mierne iným uhlom (nie rovno na bok).
 - Pokúsila sa zrýchliť pohyb, čo však viedlo k neočakávaným výsledným polohám hráča v dôsledku zvýšenia uhlovej rýchlosti. Podarilo sa jej tento problém postupným ladením odstrániť.
 - Všimla si, že JIM disponuje aj prázdnyimi fázami pre ustálenie stavu hráča. Editor nepodporuje vytváranie prázdnych fáz.
- Roman spomenul, že vytvoril projekt na stránke SourceForge a pridal členov tímu do tohto projektu.
- Ďalšie stretnutie tímu sa bude konať 24.11.2010 o **18:45**.



Úlohy z minulých stretnutí:

ID	Riešiteľ	Opis	Zadané	Termín	Status
5.8	Miro	Vytvoriť hrubý návrh štábnej kultúry (pomenúvanie premenných, nejaké štandardy písania kódu, hrubý návrh metodiky vývoja)	27. 10. 2010	01. 11. 2010	Rozpracované
6.4	Juraj	Previazanie kľbov v editore	03.11.2010	24. 11. 2010	Rozpracované
7.1	Všetci	Mať JIM-a v skompilovateľnom stave	10.11.2010	16.11.2010	Splnené
7.2	Juraj, Andrej	Umiestniť dokumentáciu k riadeniu na webový server	10.11.2010	16.11.2010	Splnené
7.3	Andrej	Analyzovať potrebnú konverziu medzi rôznymi XML formátmi, prípadná implementácia riešenia	10.11.2010	16.11.2010	Rozpracované
7.4	Juraj	Pokračovať v práci na editore pohybov, konkrétne v tvorbe symetrických pohybov	10.11.2010	16.11.2010	Rozpracované
7.5	Roman, Miro	Navrhnuť architektúru a triedy testovacieho frameworku. Špecifikovať požadovanú funkcionálnosť z dôvodu overenia návrhu systému	10.11.2010	16.11.2010	Splnené
7.6	Veronika	Vybrať si a vytvoriť jeden konkrétny pohyb a odovzdať získané skúsenosti ostatným	10.11.2010	16.11.2010	Splnené
7.7	Roman	Vytvoriť repozitár na SourceForge pre projekty	10.11.2010	16.11.2010	Splnené

Nové úlohy:

ID	Riešiteľ	Opis	Zadané	Termín	Status
8.1	Juraj	Spresniť plán	16.11.2010	24.11.2010	Zadané
8.2	Veronika	Pokračovať vo vytváraní pohybov	16.11.2010	24.11.2010	Zadané
8.3	Miro	Zistiť, ako povedať hráčovi, čo má robiť, v akom jazyku budú testy písané, ako prepojiť testovací framework s JIM-om, ako vytvoriť nekonečný polčas	16.11.2010	24.11.2010	Zadané
8.4	Roman	Začať implementovať parser logu	16.11.2010	24.11.2010	Zadané



Zápis 9. stretnutia tímu Androids

Autor zápisu:	Bc. Andrej Minárik
Dátum:	24.11.2010
Čas:	18:45 – 20:45
Prítomní:	vedúci: Ing. Ivan Kapustík členovia tímu: Bc. Juraj Belanji Bc. Miroslav Hruška Bc. Roman Kováč Bc. Andrej Minárik Bc. Veronika Wolfová ostatní: Daniel Erban
Miestnosť:	softvérové štúdio
Téma:	Zhrnutie hodnotenia dokumentácie k riadeniu, previazanie kľbov v editore a XML export v JIM-ovom formáte, forma dokumentácie, ktorá sa bude odovzdávať, JIM-ove pohyby a plan.rb, postup prác na testovacom frameworku a nápady do budúcnosti

Priebeh:

Dokumentácia k riadeniu

- Na úvod oboznámenie vedúceho s hodnotením dokumentácie k riadeniu na predmete MSI, otázka dvoch tabuliek úloh v zápisnici. Opravenie jednotlivých zápisníc. Pridáme metodiky a ďalšie dokumenty do dokumentácie.
- Dokumentácia by mala byť v jednom súbore, ak sú rôzne dokumenty v rôznych formátoch, malo by sa zjednotiť do Word-u.

Zhrnutie minulého stretnutia

- Roman zhrnul minulé stretnutie a čo sa na ňom dialo.
- XML export, testovací framework, pohyby (otáčanie, pád), náš SourceForge.
- Ďalej sme začali prechádzať úlohy z minula:
 - Štábná kultúra od Mira je hotová, pošle dokument.
 - Štábnu kultúru pre editor (C#) netreba, pretože nezasahujeme priamo do kódu editora, vytvárame nové funkcionality ako zásuvné moduly (aplikujeme našu štábnu kultúru).

K previazaniu kľbov (nielen) v editore, XML export editora

- Previazanie kľbov editora a symetrické pohyby – Juro nás informoval, že je problém previazať jednotlivé kľby – Ako prepočítavať uhly? Ktoré funkcie treba meniť?
- Symetriu a zrkadlový pohyb implementoval (pre ruky a nohy).
- Veronikin nápad – zladenie pohybu ruky a protíľahlej nohy. Nejde však o symetriu, je to skôr previazanie kľbov. Ing. Kapustík podotkol, že toto by mohlo byť zaujímavé implementovať ako logiku priamo do hráča (vyvažovanie rukami pri chôdzi, behu, kopoch...), je to však náročnejšia úloha na samostatnú (diplomovú) prácu.



- Andrejova úloha s XML exportom hotová, zostáva ešte do editora doplniť extra tlačítko (v spolupráci s Jurajom) a opraviť atribút pri finálnej fáze (Andrej vie).
- Import XML nefunguje a nebudeme ho opravovať. Import pohybov z .rmo ide, ale nenastaví kĺby robota do aktuálnej polohy.
- Juro podotkol, že práca na editore ukazuje, že bol zrejme robený pomerne narýchlo – ťažko sa nadväzuje.
- Editor je fajn na vytvorenie základného pohybu, ale na editáciu pohybov nie je vhodný.

Odovzdávanie dokumentácie o tri týždne

- O tri týždne sa odovzdáva finálna dokumentácia k zimnému semestru! Bude treba implementačnú dokumentáciu – 3 časti: testovací framework, pohyby agenta (opis ku každému pohybu a prípadne screenshoty), úpravy v editore pohybov.
- Andrej navrhol spraviť do budúcnosti pre ďalšie tímy krátky návod na tvorbu pohybov a pridať tento do dokumentácie.
- Na záver TP bude prezentácia prototypu – bude vhodné mať aj ukážky (video s akčnou hudbou = **bonus!**, sľúbil Ing. Kapustík).
- Juro spresnil plán.

Pohyby JIM-a a plan.rb

- Veronika nevytvorila tento týždeň nijaký nový pohyb, ale pripravila 3 pomocné pohyby (padanie a pod.), ktoré využíva pri vývoji vstávania.
- Pohyb *stand_back* funguje na vstávanie z chrbta aj z brucha – chceme ho zachovať pre obe situácie (ušetríme čas pri rozhodovaní a preplánovaní) – ak je robot na boku, nevie vstať na jedenkrát – treba doladiť (kvôli rýchlosti).
- Máme pomocný pohyb preklopenia na chrbát (po vstaní tak stojí bokom k lopte, čo je výhodnejšie ako keby stál chrbtom).
- Otázka na trvanie pohybu od Ing. Kapustíka - odmerať čas jednotlivých pohybov (do dokumentácie).
- Potrebujeme prísť na to, ako sa plánuje viac pohybov za sebou (JIM vykoná iba jeden a preplánuje).
- Súbor *plan.rb* by sa mal prerobiť – aby sa pozrel, či je naplánovaný ešte ďalší pohyb.
- Ďalší problém *plan.rb* – v stoji sa pokúša postaviť, resp. v ľahu sa pokúša vykonať pohyb iný ako vstávanie (možno problém s *me.on_ground*).
- Miro sa pozrie na *plan.rb*.

Testovací framework, pokračovanie práce a čo ďalej (požiadavky na pohyby)

- Ako povedať hráčovi, čo má robiť – *plan.rb*, Ruby je vhodný jazyk (skripty sa dajú rýchlo a jednoducho písať a spúšťať) a ďalej ho budeme používať.
- Čo sa očakáva od XMLka, aby sa dalo využiť učenie v našom testovacom frameworku:
 - Využitie konštánt pri písaní pohybov – do budúcnosti.
- Môže sa stať, že pri prudkých pohyboch jedného kĺbu sa iný kĺb posunie o pár stupňov – nás to však asi nijak neovplyvňuje.



- Prepojenie JIM-a a frameworku – doplniť do JIM-a server, ktorý umožní každému hráčovi získať plán od frameworku – využiť špeciálny protokol (FTP? Miro analyzuje možnosti).
- Vytvorí sa ešte jedna vrstva nad súčasným plánom, aby sme mohli jednoduchšie písať testy.
- Zásahy v JIM-ovi – high-level ruby skripty pohybov tvoriť tak, aby obsahovali čo najviac zmysluplných parametrov (napr. schopnosť dostať sa niekam – parametre cieľ, presnosť (okolie), uhol natočenia vzhľadom na cieľ).
- Na základe Mirových predstáv prebehla debata o testovaní schopnosti dostať sa niekam – pohybujeme sa na rôznych úrovniach.
- Dorobiť otáčanie po menších uhloch, vytvoriť dva druhy chôdze (s dlhými krokmi a drobnú, presnú) – na pohyboch naďalej pracuje Veronika s Andrejom.
- Dĺžka kroku určuje presnosť chôdze (priblíženie k cieľu), druhá vec je uhlová presnosť chôdze (tá nás trápi pri väčších vzdialenostiach).
- Miro nám ukázal, že nastaviť dlhšie trvanie polčasu je možné v konfiguračných súboroch servera – v adresári servera v súbore soccersim.rb, premenná rulehalftime.

Intermezzo k novej úlohe na ďalší semester - rovnováha

- Veronika – nápad sledovať Zero Moment Point, aby sme zistili pre pohyb, ako často agent vybočí zo stabilnej polohy – samotné vyklonenie z rovnovážnej polohy však niekedy môže byť legitímnou súčasťou pohybu, preto sledovanie nemôže byť triviálne (počítanie pohybov ťažiska – aproximácia, stačí približne, zanedbať niektoré časti tela a pod.) -> úloha na ďalší semester?
- Dano – pošle nám dokument od RoboCup svetového tímu, ktorí riešili výpočet ťažiska.
- Potrebné výpočty je možné zjednodušovať a nájsť vhodný pomer zjednodušenie – presnosť výpočtu.

Komunikácia frameworku a servera

- Romanova úloha – implementovať parser logu – zmenilo sa, potrebujeme informácie real-time a formát logu tiež nie je úplne ideálny – server na porte 3200 posiela informácie monitoru – budeme sa pripájať priamo sem, tu beží všetko, ako potrebujeme.
- Máme implementovaný vlastný monitor, ktorý prijíma správy od servera, parsovanie je urobené do objektového modelu.
- V parsovaní sú implementované aj testy.
- Existuje port pre trénera, ktorý môže hovoriť serveru, kam umiestniť hráča, loptu, atď.
- Ďalšie stretnutie sme opäť dohodli na **18:45**



Úlohy z minulých stretnutí:

ID	Riešiteľ	Opis	Zadané	Termín	Status
5.8	Miro	Vytvoriť hrubý návrh štabnej kultúry (pomenúvanie premenných, nejaké štandardy písania kódu, hrubý návrh metodiky vývoja)	27.10.2010	01.11.2010	splnené
6.4	Juro	Previazanie kľbov v editore	03.11.2010	24.11.2010	Nerealizujeme
7.3	Andrej	Analyzovať potrebnú konverziu medzi rôznymi XML formátmi, prípadná implementácia riešenia	10.11.2010	16.11.2010	splnené
7.4	Juraj	Pokračovať v práci na editore pohybov, konkrétne v tvorbe symetrických pohybov	10.11.2010	16.11.2010	splnené
8.1	Juro	Spresniť plán	16.11.2010	24.11.2010	splnené
8.2	Veronika	Pokračovať vo vytváraní pohybov	16.11.2010	24.11.2010	splnené
8.3	Miro	Zistiť, ako povedať hráčovi, čo má robiť, v akom jazyku budú testy písané, ako prepojiť testovací framework s JIM-om, ako vytvoriť nekonečný polčas	16.11.2010	24.11.2010	splnené
8.4	Roman	Začať implementovať parser logu	16.11.2010	24.11.2010	splnené

Nové úlohy:

ID	Riešiteľ	Opis	Zadané	Termín	Status
9.1	Andrej	HowTo k tvorbe pohybov	24.11.2010	01.12.2010	zadané
9.2	Miro	Pozrieť sa na <i>plan.rb</i> a prípadne prerobiť	24.11.2010	01.12.2010	zadané
9.3	Veronika	Vytvoriť drobnú, presnú chôdzu (napr. na základe JIM-ovej)	24.11.2010	01.12.2010	zadané
9.4	Roman	Dokončiť parsovanie	24.11.2010	01.12.2010	zadané
9.5	Andrej	Dať link na naše SVN na web	24.11.2010	01.12.2010	zadané
9.6	Roman	Poslať Ing. Bartalovi link na SVN	24.11.2010	01.12.2010	zadané
9.7	Andrej	Vytvoriť pohyb kop stranou chodidla	24.11.2010	01.12.2010	zadané
9.8	Juro	Update dokumentácie	24.11.2010	01.12.2010	zadané
9.9	Dano	Poslať link na dokument k práci s ťažiskom	24.11.2010	01.12.2010	zadané
9.10	Andrej, Juro	Zpracovať možnosť exportu XML v JIM-ovom formáte do editora pohybov	24.11.2010	01.12.2010	zadané
9.11	Veronika, Andrej	Dorobiť drobné otáčanie, vytvoriť (dopracovať) chôdzu v dvoch druhoch (dlhé kroky a drobná)	24.11.2010	01.12.2010	zadané



Zápis 10. stretnutia tímu Androids

Autor zápisu: Bc. Miroslav Hruška
Dátum: 01.12.2010
Čas: 18:45 – 20:00
Prítomní: vedúci: Ing. Ivan Kapustík
členovia tímu: Bc. Juraj Belanji
Bc. Miroslav Hruška
Bc. Roman Kováč
Bc. Andrej Minárik
Bc. Veronika Wolfová
Miestnosť: softvérové štúdio
Téma: Progres v rámci jednotlivých úloh. Tvorba predbežných plánov pre letný semester. Informácie o ukončení predmetu a prezentovaní výsledkov práce v zimnom semestri.

Priebeh:

- Na začiatku stretnutia Andrej, posledný zapisovateľ, zhrnul obsah predchádzajúceho stretnutia.
- V rámci rozhovoru o dokumentácii k riadeniu sa Miro zaviazal, že do dnešného večera odovzdá finálnu verziu štábnej kultúry vo formáte doc.
- Po pomerne krátkom úvode sme prešli konkrétne k úlohám.

Vyhodnotenie úloh

- Andrej prezentoval pokrok vo svojej úlohe - vizualizačnú pomôcku pre ostatné tímy pre urýchlenie procesu oboznámenia sa s editorom pohybov a modelom robota NAO vo všeobecnosti.
- Miro rozoberal stav, v akom sa nachádza *plan.rb*. Spoločnou diskusiou sa identifikovali pravdepodobné problémy a načrtlo sa, čo je možné s nimi urobiť. Konkrétne ide o riešenie vyššej logiky, pričom sa diskusiou dospelo, že bude realizované až v budúcom semestri a teda aj riešenie problému sa odkladá.
- Po diskusii s inžinierom Kapustíkom sme dospeli k záveru, že je nutné podrobne analyzovať možnosti, ako zapojiť do plánu viaceré akcie za sebou, nakoľko nie je implicitne určený koniec pohybu.
- Veronika načrtla problém, že agent občas pri použití možnosti *Replan* z GUI naplánuje a vykonáva inú akciu, ako má podľa podmienok v *plan.rb* a svojho aktuálneho stavu naplánuvať. Problém sa vyskytoval nepravidelne, náhodne a zatiaľ sa ho podarilo odstrániť len opätovným spustením hráča.
- Andrej si dal za úlohu spýtať sa Ivana Hujsiho, akým spôsobom vytvárali videá. Pomocou videí je totiž možné do dobrej miery prezentovať výsledky nášho úsilia v zimnom semestri.
- Veronika prezentovala skúsenosti s tvorbou chôdze, konkrétne rozprávala o pohybe *walk_fine*. Zároveň povedala, že sa jej podarilo vytvoriť extrémne drobnú chôdzu dopredu aj



dozadu. Veronika načrtla problém, ktorý spočíva v nemožnosti vytvoriť priamočiaru chôdzu vzhľadom na systém, ktorým sú riešené pohyby a vplyv náhodných chýb, ktoré do simulácie zavádza server. Z tohto zistenia vyplynula aj predstava dlhodobej práce v letnom semestri. Drobná chôdza (cupitanie) sa nepodobá na bežnú ľudskú chôdzu, preto sa Veronika rozhodla vo svojej ďalšej práci venovať aj vytvoreniu ľudskej chôdze. Taktiež prezentovala, že veľmi malé zmeny v XML súbore (niekoľko stupňov alebo milisekúnd) majú veľký vplyv na upravovanú chôdzu *walk_fine*, ktorú prebrala od tvorcov JIM-a a ďalej upravovala. Problémom je chôdzu vyladiť.

- Andrej prezentoval výsledky o tvorbe kopu do boku.
- Roman prezentoval výsledky o parsovaní, dotvoril parsovanie správ zo servera. Načrtol ďalšiu úlohu, ktorá spočíva vo vyššej objektovej reprezentácii parsovaných údajov, ku ktorej sa prihlásil. Nakoľko úloha nie je triviálna, nebolo zatiaľ stanovené, kedy bude splnená. Pravdepodobne sa úloha vyrieši až v letnom semestri. Roman by mal vytvoriť dokument o parseri a o tom, akým spôsobom bude reprezentovať údaje.
- Rozprávali sme na tému záverečnej prezentácie z predmetu Tímový projekt. Prezentácia dosiahnutých výsledkov má mať technický charakter. Je nutné dohodnúť sa na termíne prezentácie s druhým RoboCup 3D tímom. Inžinier Kapustík hovoril, že by prezentácia mohla prebehnúť 15.12.2010 alebo 16.12.2010 ráno, kedy má druhý tím stretnutie. Termín závisí aj od plánov vedúceho druhého tímu.
- Andrej s Jurajom hovorili o úlohe, ktorú mali splniť ohľadom editora pohybov. Úloha je zatiaľ v rozpracovanom stave, ale Juraj sa zaviazal, že ju do budúceho týždňa dokončí.
- Miro rozprával o prepojení testovacieho frameworku a JIM-a cez Trivial FTP server.
- Inžinier Kapustík hovoril o tom, až si finálne uzavrieme zimný semester a rozhodneme sa, čo máme dať do prezentácie.

Práca v letnom semestri

- Dospeli sme k hrubému obrysu úloh, ktorým sa budeme venovať v letnom semestri:
 - Analýza vyššej logiky (*plan.rb*).
 - Systém pohybov, stabilizačný modul.

Dokumentácia

- Juraj určil deadline na dodanie čiastkových výstupov do dokumentácie na 10.12.2010 (piatok).
- Určili sme, že dokumentáciu k prototypu v jednotlivých častiach zameriame na zodpovedanie otázok: „Čo sme robili? Ako sme to robili? Prečo sme to robili práve tak? Čo je výsledkom snaženia a čo sú jeho výhody?“

Začiatok budúceho stretnutia

- Neurčili sme explicitne začiatok stretnutia na 18:45, ale bude potrebné, aby znova bol jeho začiatok posunutý na tento čas. Miro si dal za úlohu kontaktovať v tejto veci vedúceho projektu a oboznámiť ostatných členov tímu.

**Úlohy z minulých stretnutí:**

9.1	Andrej	HowTo k tvorbe pohybov	24.11.2010	01.12.2010	Splnené
9.2	Miro	Pozrieť sa na plan.rb a prípadne prerobiť	24.11.2010	01.12.2010	Splnené
9.3	Veronika	Vytvoriť drobnú, presnú chôdzu (napr. na základe JIM-ovej)	24.11.2010	01.12.2010	Splnené
9.4	Roman	Dokončiť parsovanie	24.11.2010	01.12.2010	Splnené
9.5	Andrej	Dať link na naše SVN na web	24.11.2010	01.12.2010	Splnené
9.6	Roman	Poslať Ing. Bartalosovi link na SVN	24.11.2010	01.12.2010	Splnené
9.7	Andrej	Vytvoriť pohyb kop stranou chodidla	24.11.2010	01.12.2010	Splnené
9.8	Juro	Update dokumentácie	24.11.2010	01.12.2010	Splnené
9.9	Dano	Poslať link na dokument k práci s ťažiskom	24.11.2010	01.12.2010	Splnené
9.10	Andrej, Juro	Zapracovať možnosť exportu XML v JIM-ovom formáte do editora pohybov	24.11.2010	01.12.2010	Rozpracované
9.11	Veronika, Andrej	Dorobiť drobné otáčanie, vytvoriť (dopracovať) chôdzu v dvoch druhoch (dlhé kroky a drobná)	24.11.2010	01.12.2010	Splnené

Nové úlohy:

ID	Riešiteľ	Opis	Zadané	Termín	Status
10.1	Juraj	Dokumentácia k editoru	01.12.2010	10.12.2010	Zadané
10.2	Andrej	Dokumentácia k editoru, časť o konvertovaní XML	01.12.2010	10.12.2010	Zadané
10.3	Roman	Dokumentácia k parsovaniu. Prípadne rozšírenie v zmysle návrhu tried pre objektovú reprezentáciu sveta	01.12.2010	10.12.2010	Zadané
10.4	Veronika	Dokumentácia k pohybom	01.12.2010	10.12.2010	Zadané
10.5	Miroslav	Dokumentácia k prepojeniu testovacieho frameworku a JIM-a	01.12.2010	10.12.2010	Zadané
10.6	Juraj	Zapracovať zmeny do editora pohybov. Pozrieť sa bližšie na logiku v <i>plan.rb</i>	01.12.2010	08.12.2010	Zadané
10.7	Miroslav	Implementovať TFTP klienta v testovacom frameworku	01.12.2010	08.12.2010	Zadané
10.8	Andrej	Zistiť, ako tvoriť videá na prezentáciu	01.12.2010	08.12.2010	Zadané



Zápis 11. stretnutia tímu Androids

Autor zápisu:	Bc. Juraj Belanji
Dátum:	08.12.2010
Čas:	18:50 – 19:50
Prítomní:	vedúci: Ing. Ivan Kapustík členovia tímu: Bc. Juraj Belanji Bc. Miroslav Hruška Bc. Roman Kováč Bc. Andrej Minárik Bc. Veronika Wolfová ostatní: Daniel Erban (meškal 15 min.)
Miestnosť:	softvérové štúdio
Téma:	Sumarizácia a zhodnotenie stavu projektu po zimnom semestri, plány pre letný semester, organizačné záležitosti ohľadom prezentácie prototypu.

Priebeh:

- Toto je posledné regulárne stretnutie tímu v zimnom semestri.
- Na začiatku stretnutia nás Ing. Kapustík oboznámil s tým, že si všimol, že je časť web stránky s odkazmi zle kódovaná. Andrej povedal, že to upraví.
- Ing. Kapustík nám oznámil, že pôvodne dohodnutý termín prezentácie už nie je možné dodržať, keďže v tom čase rieši iné záležitosti (disciplinárna komisia). Predbežné nové termíny stretnutia sú:
 - streda do 16:00
 - štvrtok po 16:00
- Samotná prezentácia prototypu najčastejšie trvá cca. 20 min + 10 min diskusia.
- Miro následne zhrnul priebeh minulého stretnutia.
- Ing. Kapustík hovoril o jeho stretnutí s diplomantmi, ktorí pracujú na JIM-ovi, v súvislosti s problémami s plánovačom (plan.rb). Navrhol, aby sme im opísali problém a riešenie nechali na nich, keďže majú lepší prehľad, čo ako funguje (sami to vytvorili), a aj pre nich je dôležité, aby sa odhalená chyba odstránila.
- Andrej hovoril o možnostiach nahrávania pohybov. Skontaktoval Ivana Hujsiho, ktorý mu povedal, s akým nástrojom ich minuloročný tím nahrával pohyby. Program je voľne dostupný, takže môžeme nahrávať naše vytvorené pohyby aj my rovnakým nástrojom.
- Roman hovoril o pokroku v implementovaní parsera správ. Na parsovaní sa bude pokračovať v letnom semestri. Zatiaľ má rozanalyzované transformačné matice, z ktorých sa budú získavať relevantné informácie, ktoré budeme používať pri testovaní a tréňovaní agenta.
- Vystala otázka ohľadom začiatku súradníc. Roman zistil, že sa v matici absolútne súradnice lopty v strede ihriska nezhodujú s predpokladanými súradnicami stredu hracej plochy. Potrebné je zistiť, ako sa tieto súradnice vypočítavajú.
- Veronika spomenula, že jej nefungovali niektoré skratky na ovládanie monitora.



- 14.12.2010 (utorok) do 14:00 treba odovzdať dokumentáciu - elektronicky na stránku tímu aj v papierovej verzii vedúcemu. Zhodli sme sa, že v utorok prinesieme vytlačенú verziu dokumentácie Ing. Kapustíkovi po prednáške z Kódovania.
- Miro hovoril o TFTP serveri. Bude sa prenášať súbor so špeciálnym názvom a v JIM-ovi sa spustí Ruby skript s plánom, ktorý chceme, aby sa vykonal.
- Dohodli sme sa, čo budeme prezentovať o prototypu:
 - Veronika bude hovoriť o implementovaných pohyboch
 - Andrej bude hovoriť o kódoch a pomôcke pri tvorbe pohybov
 - Juro bude hovoriť o editore a zmenách v ňom
 - Roman bude hovoriť o parsovaní správ zo servera
 - Miro bude hovoriť o testovacom frameworku, vlastnom monitore a TFTP protokole
- Do prezentácie potrebujeme dať na záver naše plány do budúcnosti (letný semester).
- Zhodli sme sa, že prezentáciu dajú dokopy Veronika a Juro, pričom si každý člen osobne pripraví slajdy k svojej časti prezentácie.
- Vhodné je v úvode prezentácie povedať, na koho sme nadväzovali a čo sme prebrali, a na záver, čo sme konkrétne zistili pri implementácii prototypu a ako plánujeme pokračovať.
- Následne sme pokračovali vo voľnej diskusii. Hovorilo sa o RoboCup Junior, o spôsobe hodnotení a určili sme zmenu vedúceho pre nasledovný semester. Vedúci bude Juro a Veronika preberie úlohu zástupcu vedúceho. Zatiaľ sme sa zhodli, že si ostatné manažérske roly nebudeme vymieňať.

Úlohy z minulých stretnutí:

ID	Riešiteľ	Opis	Zadané	Termín	Status
9.10	Andrej, Juraj	Zapracovať možnosť exportu XML v JIM-ovom formáte do editora pohybov	24.11.2010	01.12.2010	Splnené
10.1	Juraj	Dokumentácia k editoru	01.12.2010	10.12.2010	Rozpracované
10.2	Andrej	Dokumentácia k editoru, časť o konvertovaní XML	01.12.2010	10.12.2010	Rozpracované
10.3	Roman	Dokumentácia k parsovaniu. Prípadne rozšírenie v zmysle návrhu tried pre objektovú reprezentáciu sveta	01.12.2010	10.12.2010	Rozpracované
10.4	Veronika	Dokumentácia k pohybov	01.12.2010	10.12.2010	Rozpracované
10.5	Miroslav	Dokumentácia k prepojeniu testovacieho frameworku a JIM-a	01.12.2010	10.12.2010	Rozpracované
10.6	Juraj	Zapracovať zmeny do editora pohybov. Pozrieť sa bližšie na logiku v <i>plan.rb</i>	01.12.2010	08.12.2010	Splnené
10.7	Miroslav	Implementovať TFTP klienta v testovacom frameworku	01.12.2010	08.12.2010	Splnené
10.8	Andrej	Zistiť, ako tvoriť videá na prezentáciu	01.12.2010	08.12.2010	Splnené



Nové úlohy:

ID	Riešiteľ	Opis	Zadané	Termín	Status
11.1	Miro	Napísať ukázkový skript v Ruby pre účel prezentácie	08.12.2010	14.12.2010	Zadané
11.2	Juraj, Veronika	Dohodnúť termín prezentácie s vedúcimi, druhým tímom a technologickým konzultantom	08.12.2010	14.12.2010	Zadané
11.3	Andrej	Opraviť chyby v kódovaní na stránke	08.12.2010	14.12.2010	Zadané
11.4	všetci	Pripraviť slajdy k prezentácii	08.12.2010	14.12.2010	Zadané
11.5	Veronika, Juraj	Pripraviť prezentáciu k prototypu	08.12.2010	14.12.2010	Zadané



Metodiky a návody na použitie – časť C

V tejto časti sa nachádzajú návody a metodiky, ktoré pomáhajú pri riešení úloh a riadení tímu.

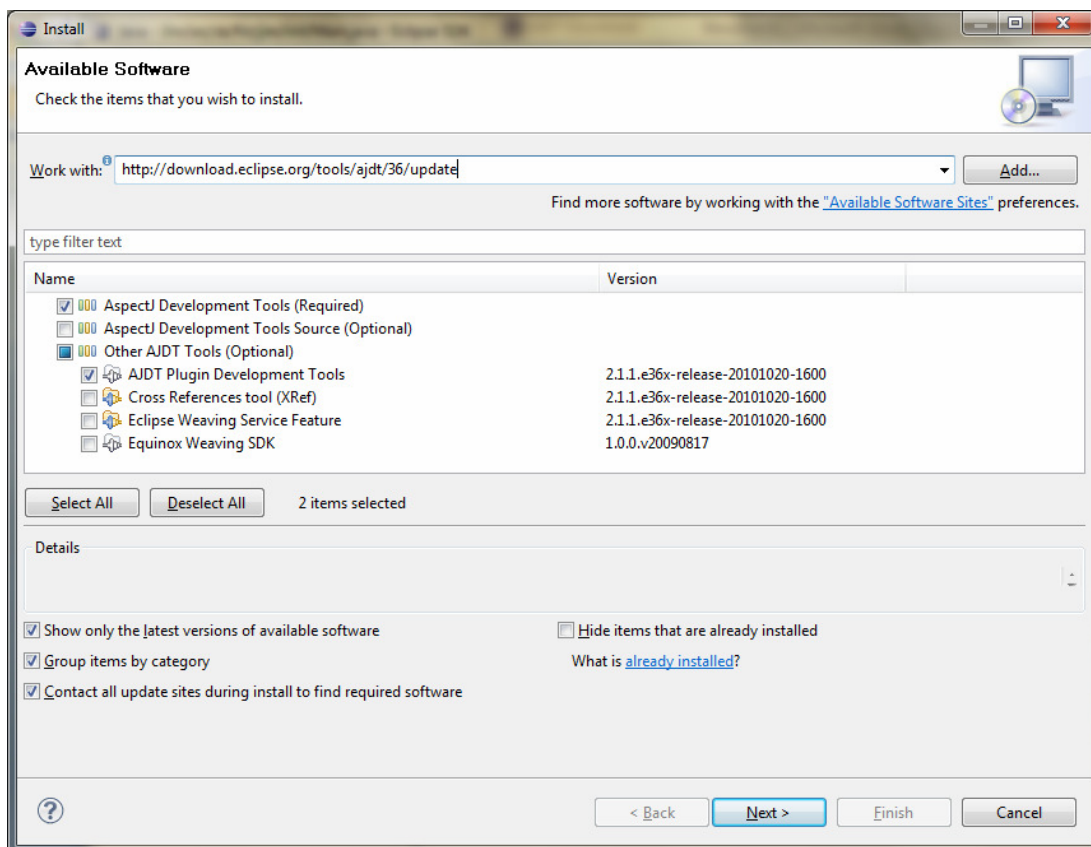


Návod na nastavenie SVN

AspectJ

Potrebné kvôli GUI v JIM-ovi.

1. Compiler
 - a. Stiahnuť zo stránky <http://www.eclipse.org/aspectj/downloads.php> aspectj-1.6.10.jar
 - b. Nainštalovať príkazom `java -jar <umiestnenie stiahnutého jar>` (v prípade Windowsu treba spustiť cmd ako admin)
 - c. Postupovať podľa inštrukcií
2. Plugin do eclipseu
 - a. Spustiť Eclipse
 - b. Ísť do Help -> Install new software, tam zadať <http://download.eclipse.org/tools/ajdt/36/update>
 - c. Zaškrtnúť časti zobrazené na tomto obrázku



- d. Preklikat', potvrdiť všetko, reštartnúť eclipse

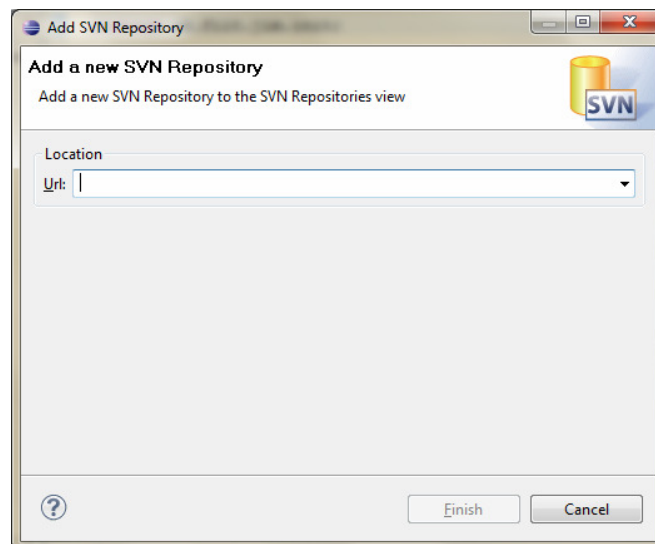


Subclipse

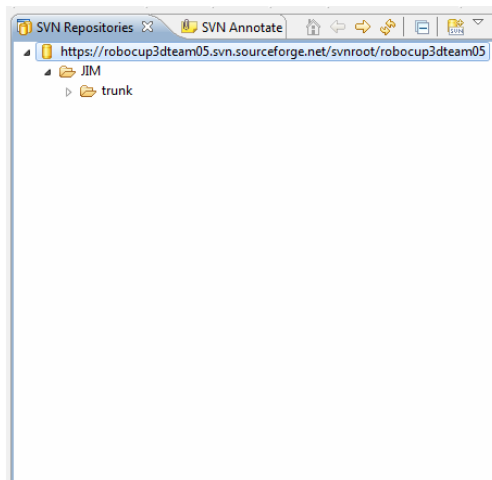
1. Spustiť Eclipse
2. Ísť do Help -> Install new software, tam zadať http://subclipse.tigris.org/update_1.6.x
3. Zaškrtnúť Subclipse
4. Preklikáť, potvrdiť všetko, reštartnúť eclipse

SVN repozitár

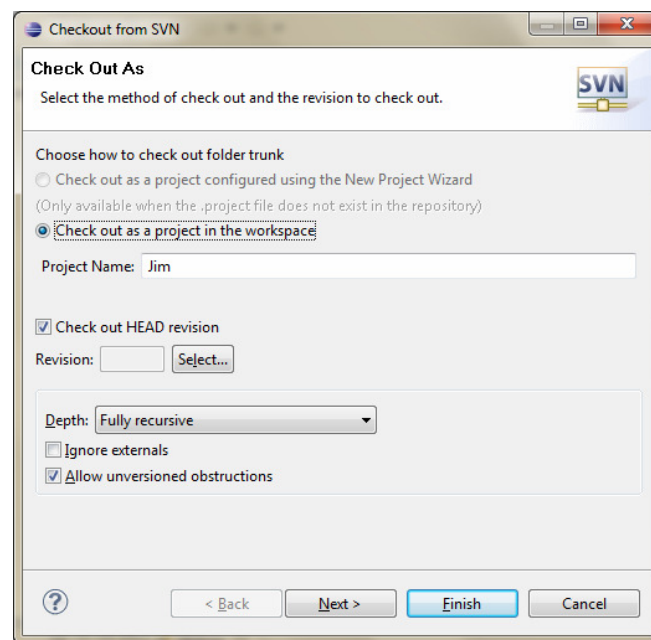
1. Spustiť Eclipse
2. Vpravo hore kliknúť na Open perspective -> Other -> SVN Repository
3. Keď sa otvorí perspektíva, vľavo kliknúť na Add SVN Repository
4. Zobrazí sa



5. Zadať <https://robocup3dteam05.svn.sourceforge.net/svnroot/robocup3dteam05>
6. Čakať
7. Dostať do takéhoto stavu



8. Pravým kliknúť na adresár *trunk* a dať Checkout
9. Zobrazí sa



10. Nemeniť nič (jedine názov projektu, keď chcete)
11. Finish
12. Dlhو čakať
13. Môže sa stať, že to zamrzne. V tom prípade vypnúť Eclipse a opakovať Checkout.
14. Ak to nezamrzlo, prepnete si späť perspektívu na Java a mali by ste tam mať funkčného JIM-a aj s GUI, ktoré sa zobrazí pri spustení JIM-a.

Ak budete niečo meniť, zmenu potvrdíte tak, že kliknete pravým na celý projekt, Team -> Commit. Možno na vás vyskočí okno, že máte zadať meno a heslo, tak zadajte meno a heslo zo SourceForge.



Ak potvrdzujete zmeny, zadajte k nim stručný opis po anglicky. Nezabúdajte si dávať update predtým, ako idete niečo meniť (Team -> Update to HEAD).



Návod na zverejnenie zápisníc

1. Napísanie zápisnice podľa šablóny dostupnej na webovej stránke tímu v časti Súbory
2. Nahradenie faktov v {} reálnymi informáciami
3. Na druhý deň po stretnutí nazdieľať tímu editovateľnú formu zápisnice (formát MS Word) v nástroji Google Docs, aby bola ešte pred zverejnením dostupná všetkým
4. Pomenovanie zápisnice podľa konvencie <cislo_stretnutia>_zapisnica_<yyyymmdd>.doc, teda napr. 01_zapisnica_20100929.doc
5. Kontrola gramatickej a informačnej stránky iným členom tímu (editor - spravidla Veronika) podľa možnosti ešte v deň vytvorenia
6. Prekonvertovanie zápisnice do formátu PDF a upload na web tímu (editorom alebo správcom webstránky) podľa možnosti najneskôr na druhý deň po tímovom stretnutí
7. Súbor sa nahráva na server do adresára files
8. Je potrebná zmena práv súboru zápisnice (chmod) na hodnotu 774, aby bolo umožnené ostatným členom tímu zápisnicu v prípade potreby nahradiť aktuálnou verziou
9. Pre zverejnenie zápisnice je potrebné pridanie HTML kódu do súboru subory.html (tabuľka „Zápisnice zo stretnutí“) a vyplnenie potrebných informácií, predovšetkým krátkeho obsahu stretnutia hlavnými preberanými bodmi:

```
<tr>
  <td class="cislo">{cislo_stretnutia}</td>
  <td class="datum">{datum_stretnutia}</td>
  <td>{obsah_stretnutia}</td>
  <td class="stiahnut">
    <a href="files/{nazov_suboru}">PDF</a>
  </td>
</tr>
```

Poznámka: V prípade nesprávneho zobrazovania znakov stránky so súbormi pri editovaní HTML súboru môže pomôcť zmena kódovania súboru (kódovanie na serveri labss nastavené na ISO-8859-2).



Metodika pre manažment úloh v distribuovanom projekte prostredníctvom nástroja dotProject

Úvod

Táto metodika určuje postup manažmentu úloh v malom tíme a bola vytvorená pre potreby tímu 05 (Androids), ktorý sa ňou riadi pri práci na projekte RoboCup 3D v predmete Tímový projekt v akademickom roku 2010/2011. Na vyššej úrovni rozoberá postup manažmentu úloh v tíme všeobecne a na nižšej úrovni určuje postup a formálnu stránku pre zadávanie a prideľovanie úloh v nástroji dotProject.

1. Manažment úloh v distribuovanom projekte

Manažment úloh je veľmi dôležitý pre úspešné riadenie projektu. Umožňuje manažérovi projektu sledovať rýchlosť napredovania tímu v práci na projekte a vyťaženosť jednotlivých členov tímu, čím podporuje riadenie tímu.

1.1. Postup

Manažment úloh pozostáva z fáz a činností, ktorých nadväznosti ilustruje diagram na *Obr. 1*.

Identifikácia úlohy

Vstup: myšlienky, návrhy, požiadavky, problémy, zmeny

Výstup: identifikovaná úloha

Zodpovedný: správca úloh

Potenciálnu novú úlohu tím najčastejšie identifikuje počas tímového stretnutia alebo pri riešení inej úlohy. Tím úlohu analyzuje a vyhodnotí, či skutočne ide o novú úlohu a či je zmysluplná a potrebná. Ak ide o zmysluplnú novú úlohu, správca úloh ju zaradí do zoznamu úloh, inak je úloha zrušená.

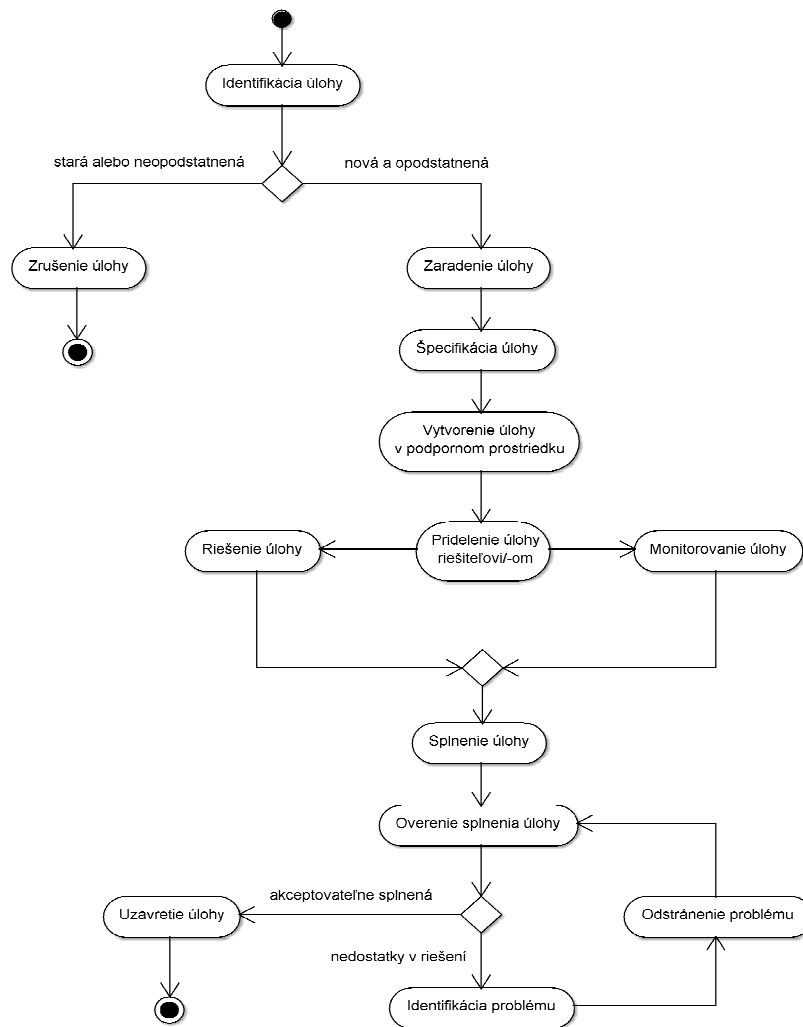
Zrušenie úlohy

Vstup: identifikovaná úloha

Výstup: protokol o zamietnutí úlohy

Zodpovedný: správca úloh

Správca úloh vypracuje protokol o zamietnutí úlohy, ktorý obsahuje opis úlohy a dôvod zamietnutia.



Obr.1 Diagram činností v procese manažmentu úloh

Zaradenie úlohy

Vstup: identifikovaná úloha

Výstup: aktualizovaný zoznam úloh

Zodpovedný: správca úloh

Správca úloh zaradí identifikovanú úlohu do zoznamu úloh tímu.

Špecifikácia úlohy

Vstup: identifikovaná úloha

Výstup: podrobná analýza úlohy



Zodpovedný: správca úloh

Ak správca úloh úlohu zaradil do zoznamu úloh na vypracovanie, táto úloha sa bližšie špecifikuje, teda určí sa, čo úloha obnáša, koľko človeko-hodín bude trvať, čo je potrebné pre jej splnenie, koľko členov tímu ju bude riešiť a aká je jej priorita.

Vytvorenie úlohy v podpornom prostriedku

Vstup: podrobná analýza úlohy

Výstup: úloha v podpornom prostriedku

Zodpovedný: správca úloh

V tomto kroku sa predpokladá, že tím používa podporný prostriedok pre manažment úloh. Správca úloh zadá úlohu do tohto prostriedku, pričom vyplní všetky potrebné údaje o úlohe. Medzi hlavné údaje patria opis a priorita úlohy, odhadované trvanie a stanovený termín splnenia.

Pridelenie úlohy riešiteľovi/riešiteľom

Vstupy: úloha v podpornom prostriedku, údaje o členoch tímu

Výstup: dvojica úloha-riešiteľ/-ia

Zodpovedný: správca úloh

Správca úloh pridelí úlohu členovi tímu na vyriešenie. Člen tímu musí o pridelení vedieť a súhlasiť s ním. Člen tímu je vybraný na základe predpokladov o jeho schopnosti vyriešiť úspešne danú úlohu a na základe jeho pracovného vyťaženia. Ak je úloha časovo a rozsahovo príliš náročná pre pridelenie jednému členovi tímu, správca úloh ju pridelí viacerým členom, pričom jedného z nich poverí zodpovednosťou za jej úspešné vyriešenie. Zodpovednosť za úlohu zahŕňa koordináciu ostatných členov, ktorým bola úloha pridelená na vyriešenie, a podávanie hlásení o plnení úlohy. Členovia tímu, ktorým bola úloha pridelená, sú ďalej nazývaní „riešitelia“, riešiteľ zodpovedný za skupinu riešiteľov je nazývaný „vedúci riešiteľ“.

Riešenie a monitorovanie úlohy

Vstup: podrobná analýza úlohy

Výstup: správy o stave riešenia úlohy

Zodpovední: (vedúci) riešiteľ, správca úloh

Riešiteľ/riešitelia pracujú na vyriešení pridelenej úlohy, pričom podávajú pravidelne správy o stave riešenia úlohy správcovi úloh. Správca úloh monitoruje postup prác na úlohe.

Splnenie úlohy



Vstup: úloha

Výstup: riešenie úlohy a prislúchajúce dokumenty

Zodpovedný: (vedúci) riešiteľ

Riešiteľ (vedúci riešiteľ) oznámi správcovi úloh, že úloha je vyriešená a odovzdá mu riešenie úlohy spolu s prislúchajúcimi dokumentmi.

Overenie splnenia úlohy

Vstup: podrobná analýza úlohy, riešenie úlohy a prislúchajúce dokumenty

Výstup: vyhodnotenie riešenia úlohy

Zodpovedný: správca úloh

Správca úloh overí správnosť a úplnosť riešenia úlohy. Ak je úloha akceptovateľne splnená, správca úloh ju v prostriedku na manažment úloh označí za splnenú a oficiálne ju uzavrie. Ak správca úloh pri overovaní splnenia úlohy odhalí nedostatky v jej riešení, vráti ho riešiteľom na prepracovanie spolu s poznámkami, aké nedostatky identifikoval.

Uzavretie úlohy

Vstup: vyhodnotenie riešenia úlohy

Výstup: protokol o uzavretí úlohy

Zodpovedný: správca úloh

Správca úloh vypracuje protokol o uzavretí úlohy, ktorý obsahuje opis úlohy, hodnotenie riešenia a dátum uzavretia.

Identifikácia problému

Vstup: vyhodnotenie riešenia úlohy

Výstup: opis problému

Zodpovedný: správca úloh

Správca úloh identifikuje a opíše problém v riešení úlohy alebo v prislúchajúcich dokumentoch. Riešenie úlohy vráti na prepracovanie riešiteľovi/-om spolu s opisom problému.

Odstránenie problému

Vstup: opis problému

Výstup: nové riešenie úlohy a/alebo prislúchajúce dokumenty

Zodpovedný: (vedúci) riešiteľ



Riešiteľ/-ia odstránia identifikovaný problém v riešení úlohy a/alebo prislúchajúcich dokumentoch.

1.2. Roly

Z vyššie uvedeného postupu manažmentu úloh vyplývajú nasledovné roly:

- správca úloh
- riešiteľ
- vedúci riešiteľ

Správca úloh

Člen tímu: projektový manažér

Zodpovednosti:

- identifikácia úlohy
- zrušenie a zaradenie úlohy
- špecifikácia úlohy
- vytvorenie úlohy v podpornom prostredí
- priradenie úlohy riešiteľovi/-om
- monitorovanie úlohy
- overenie splnenia úlohy
- uzavretie úlohy
- identifikácia problému

Riešiteľ

Člen tímu: ľubovoľný podľa úlohy

Účastník:

- identifikácia úlohy
- špecifikácia úlohy

Zodpovednosti:

- riešenie úlohy
- splnenie úlohy
- odstránenie problému

Povinnosti:

- pravidelne podávať hlásenia o stave riešenia úlohy nadriadenému (správca úloh/vedúci riešiteľ)

Vedúci riešiteľ



Pozri časť *Riešiteľ* vyššie.

Rozšírenie povinností:

- koordinácia tímu riešiteľov
- podávanie súhrnných správ správcovi úloh

2. Zadávanie a pridelovanie úloh v prostredí dotProject

Táto konkretizácia procesu zadávania a pridelovania úloh je určená pre malé tímy v predmete Tímový projekt, ktoré používajú podporný prostriedok dotProject. Vzhľadom na charakteristiky tímových projektov nie je nutné a ani možné korektne vyplniť niektoré nástrojom poskytnuté polia, preto polia, ktorých vyplňanie nie je v metodike explicitne opísané, si zachovávajú svoju prednastavenú hodnotu.

Pred samotným zadávaním úloh si nástroj dotProject vyžaduje vytvorenie spoločnosti a projektu, ktorým budú úlohy patriť. Opis postupu pre tieto činnosti nie je predmetom tejto metodiky. Vychádza z predpokladu, že sú splnené všetky podmienky pre úspešné vytvorenie novej úlohy a jej pridelenie riešiteľom.

2.1. Vytvorenie úlohy

Ako bolo uvedené v *kap. 1.1* a *1.2*, úlohy v podpornom prostriedku vytvára správca úloh. Úlohy sa vytvárajú a pridelujú čo najskôr po ich špecifikácii.

Inicializácia vytvárania úlohy:

1. prihlási sa do podporného prostriedku prostredníctvom internetového prehliadača
2. prejde na zobrazenie projektov
3. vyberie projekt, ktorému chce pridať úlohu
4. zvolí možnosť *new task* (*nová úloha*) v pravej hornej časti obrazovky

Vyplnenie základných polí:

1. Task Name (Názov úlohy)
 - stručný a výstižný
 - ak sa nachádza úloha v zápisnici zo stretnutia tímu, názov začína identifikačným číslom zo zápisnice
 - dva spôsoby formulácie
 - a. obsahuje slovesné podstatné meno/sloveso v neurčitku
 - b. obsahuje podstatné meno, ktoré pomenúva ucelený problém, ktorého riešenie je predmetom úlohy



2. Priority (Priorita)

- tri dostupné stupne:
 - a. high (vysoká)
 - i. ak na úlohu nadväzuje iná úloha
 - ii. ak najneskorší termín splnenia úlohy treba nutne dodržať
 - b. low (nízka)
 - i. časovo nenáročná úloha, ktorej termín ukončenia nenastane do dvoch týždňov od vytvorenia
 - ii. okrajová úloha, ktorá neovplyvňuje dodržanie termínov, fungovanie tímu ani kvalitu výsledného produktu
 - c. normal (normálna)
 - i. všetky ostatné úlohy

3. Milestone? (Míľnik?)

- pre úlohu zo skupiny 2. a) ii.

Po vyplnení vyššie uvedených základných údajov o úlohe správca úloh vyplní polia prvých troch záložiek – *Details (Detaily)*, *Dates (Dátumy)* a *Dependencies (Závislosti)*.

Detaily

Pre uvažovaný typ projektov správca úloh vyplní:

1. Description (Opis)

- a. musí riešiteľovi presne vymedziť zadanie úlohy
- b. všetky požiadavky, ktoré má spĺňať riešenie úlohy
- c. forma, rozsah a hĺbka očakávaného výstupu
- d. vedúci riešiteľ, ak úlohu rieši tím riešiteľov

2. Task Type (Typ úlohy)

- a. Administrative (Administratívna)
 - riadiaca/organizačná úloha, písanie dokumentácie, podávanie hlásení, manažérska úloha
- b. Operative (Operatívna)
 - implementačná/testovacia úloha
 - prispieva ku kvalite a rozsahu výsledku projektu
- c. Unknown (Neznáma)
 - ak správca úloh nevie úlohu zaradiť do skupiny a) alebo b)

3. Task Parent (Rodičovská úloha)

- a. ak je úloha podúlohou inej úlohy, správca úloh nastaví rodičovskú úlohu



Dátumy

V tejto záložke správca úloh vyplní polia:

1. Start Date (Dátum začiatku)
 - z kalendára vyberie dátum, kedy bola úloha špecifikovaná
 - nastaví čas, kedy bola úloha špecifikovaná
2. Finish Date (Dátum ukončenia)
 - z kalendára vyberie dátum najneskoršieho prípustného ukončenia úlohy
 - nastaví čas, kedy musí byť v deň ukončenia úloha vyriešená
 - v prípade úloh s vysokou prioritou dátum ukončenia nastaví tak, aby vznikla dostatočná časová rezerva
3. Expected Duration (Očakávané trvanie)
 - orientačný údaj
 - odhad počtu človeko-hodín/človeko-dní, ktoré riešenie úlohy spotrebuje
 - vyplní na základe konzultácie s riešiteľom/-mi

Závislosti

Správca úloh pre uvažovaný typ projektov vyplní len pole *Task Dependencies (Závislosti úlohy)*, a to v prípade, že riešenie vytváranej úlohy závisí od vyriešenia inej úlohy. Obmedzujúcu úlohu správca úloh vyberie zo zoznamu všetkých úloh.

2.2. Pridelenie úlohy riešiteľom

Pridelenie úlohy riešiteľom v podpornom prostriedku vykoná správca úloh na základe rozhodnutia z fázy *Pridelenie úlohy riešiteľovi/-om* z kap. 1.1. Pre pridelenie úlohy je určená posledná záložka formuláru pre zadávanie úloh v dotProject *Human Resources (Ľudské zdroje)*.

Pridelenie úlohy v dotProject:

1. výber 1 alebo viac riešiteľov zo zoznamu členov tímu
2. nastavenie percentuálneho podielu množstva práce na úlohe, ktorého vykonanie sa očakáva od konkrétneho riešiteľa pre každého z vybraných riešiteľov

Po ukončení pridelenia úlohy riešiteľom správca úloh potvrdí vytvorenie úlohy. V prípade potreby údaje o úlohe počas jej riešenia na základe konzultácie s riešiteľmi upravuje. Všetci riešitelia musia byť na prípadnú zmenu náležite upozornení.



Metodika pre manažment verzií zdrojového kódu

Manažment verzií zdrojového kódu

Účelom tejto metodiky je definovať postup pri manažmente verzií zdrojového kódu softvérového projektu. Metodika sa zaoberá procesmi zobrazenými v nasledujúcej tabuľke.

TABUĽKA 1. PROCESY

Názov	Kapitola
Založenie projektu	3.1
Prevzatie aktuálnej verzie zdrojového kódu	3.2
Odobzdanie zmien v zdrojovom kóde	3.3
Zamknutie súboru so zdrojovým kódom	3.4
Vetvenie projektu	3.5
Riešenie konfliktov v zdrojovom kóde	3.6

1. Slovník pojmov

Pojem	Opis
Ťaví štýl	Ťaví štýl formátovania textu spája slová v slovnom spojení do jedného, kde každé slovo začína veľkým písmenom a všetky ostatné písmená v tomto slove sú písané malým písmenom. V tomto štýle sa nepoužíva interpunkcia. Toto pravidlo platí aj pre skratky. Príkladom je slovné spojenie SVN repozitár, ktoré by bolo v tomto štýle formátované ako reťazec SvnRepozitar.

2. Roly a zodpovednosti

V procesoch, ktorými sa metodika zaoberá sú nevyhnutné nasledujúce roly spolu s ich zodpovednosťami.

TABUĽKA 2. ROLY A ICH ZODPOVEDNOSTI

Rola	Zodpovednosť
Manažér vývoja	<ul style="list-style-type: none"> • Založenie projektu • Vytvorenie novej vetvy • Návrat k jednej z predchádzajúcich verzií zdrojového kódu • riešenie vážnych konfliktov pri odovzdávaní zmien v zdrojovom kóde jedným z programátorov • Spojenie vedľajších vetiev s hlavnou
Programátor	<ul style="list-style-type: none"> • Kontrola nových verzií zdrojového kódu • Prevzatie aktuálnej verzie zdrojového kódu • Zamknutie súboru so zdrojovým kódom • Odovzdanie zmien v zdrojovom kóde • Požiadanie o vytvorenie novej vetvy • riešenie jednoduchých konfliktov pri odovzdávaní zmien v zdrojovom kóde



3. Procesy

3.1. Založenie projektu

Vstup: požiadavka na založenie nového projektu

Výstup: založený projekt vo verziovacom systéme

Zodpovedný: manažér vývoja

Manažér vývoja dostane požiadavku na založenie nového projektu. Následne postupuje podľa nasledujúceho postupu:

1. Určí názov projektu, ktorý bude mať nasledujúcu formu: <NazovSpolocnosti>_<NazovProjektu>, kde reťazec <NazovSpolocnosti> bude nahradený názvom spoločnosti, pre ktorú sa projekt vyvíja a reťazec <NazovProjektu> bude nahradený názvom projektu. Pri oboch názvoch bude použitý ťaví štýl.
2. Vytvorí projekt vo verziovacom systéme.
3. Nastaví práva projektu tak, že sám bude administrátorom projektu so všetkými právami a programátori zúčastnení v projekte budú môcť prehliadať, pridávať, meniť a mazať súbory so zdrojovým kódom.
4. Vytvorí adresárovú štruktúru projektu nasledujúcim spôsobom:
 - a. Vytvorí podadresár s názvom *trunk*, ktorý predstavuje hlavnú vetvu projektu
 - b. Vytvorí podadresár s názvom *branches*, ktorý bude obsahovať vedľajšie vetvy projektu

3.2. Prevzatie aktuálnej verzie zdrojového kódu

Výstup: aktuálna verzia zdrojového kódu u programátora

Zodpovedný: programátor

Programátor pred začatím práce na projekte prevezme aktuálnu verziu zdrojového kódu, čo vykoná nasledujúcim postupom:

1. Ubezpečí sa, že nemá neodovzdané zmeny na projekte. V prípade, že takéto zmeny má neodovzdané, postupuje sa podľa kapitoly 3.3.
2. Prevezme aktuálnu verziu zdrojového kódu pre celý projekt.
3. V prípade, že pri prevzatí aktuálnej verzie zdrojového kódu nastanú konflikty v zdrojovom kóde v dôsledku chyby pri kroku číslo 1, postupuje sa podľa kapitoly 3.6.



3.3. Odovzdanie zmien v zdrojovom kóde

Vstup: kompilovateľná verzia zdrojového kódu

Výstup: aktualizovaný projekt vo verziovacom systéme

Zodpovedný: programátor

Programátor po vykonaní zmien v zdrojovom kóde odovzdá tieto zmeny do verziovacieho systému. Postupuje podľa nasledujúceho postupu:

1. Ubezpečí sa, že aktuálna verzia zdrojového kódu je kompilovateľná.
2. Spustí testy.
3. V prípade, že testy neboli úspešné, odhalí chybu a opraví ju. Odovzdanie zmien nemôže prebehnúť.
4. Označí celý projekt so zdrojovým kódom a v závislosti od zvoleného verziovacieho systému stlačí akciu pre odovzdanie projektu.
5. Opíše zmeny v zdrojovom kóde v anglickom jazyku.
6. Potvrdí vykonané zmeny.
7. V prípade, že nastanú konflikty postupuje podľa kapitoly 3.6.

3.4. Zamknutie súboru so zdrojovým kódom

Vstup: súbor so zdrojovým kódom

Výstup: zamknutý súbor so zdrojovým kódom

Zodpovedný: programátor

Programátor potrebuje výhradne pracovať so zdrojovým kódom a nechce, aby na tej istej časti zdrojového kódu pracoval aj iný programátor. Dodržiava pri tom pravidlo, že uzamyká čo možno najmenej súborov a nemá ich uzamknuté dlhšie ako 3 dni. Ak je nevyhnutné uzamknúť súbor na dlhšiu dobu, kontaktuje manažéra vývoja a požiada o povolenie. Postupuje nasledujúcim spôsobom:

1. Zvolí jeden alebo viac súborov so zdrojovým kódom, ktorý chce uzamknúť.
2. Pred uzamknutím súborov napíše dôvod v anglickom jazyku, prečo ich uzamyká. Formát tohto dôvodu je voľný.
3. Uzamkne zvolené súbory.



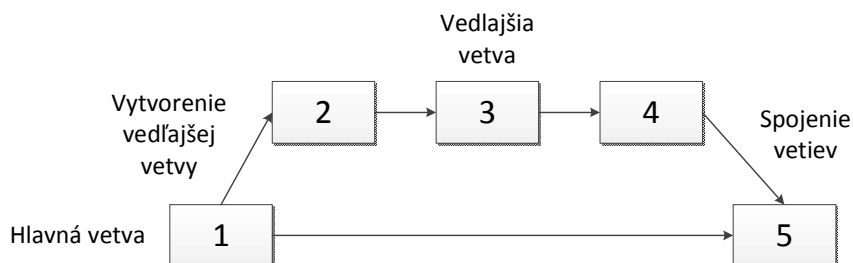
3.5. Vetvenie projektu

Vstup: požiadavka na vytvorenie vetvy projektu

Výstup: vytvorená vetva projektu

Zodpovedný: manažér vývoja

Manažér vývoja dostane požiadavku na vytvorenie novej vedľajšej vetvy projektu alebo sám usúdi, že projekt si vyžaduje novú vetvu. Nová vetva projektu sa vytvorí iba vtedy, ak sa očakáva dlho trvajúci vývoj, ktorý by mohol ovplyvniť funkcionality projektu. Štruktúru tohto procesu zobrazuje nasledujúci obrázok.



OBRÁZOK 1. ŠTRUKTÚRA PROCESU VETVENIA PROJEKTU

Manažér vývoja postupuje pri vytváraní novej vetvy podľa nasledujúceho postupu:

1. Vytvorí názov vetvy, ktorý bude pozostávať z čísla novej verzie projektu vyvíjanej v novej vetve projektu. Formát názvu je nasledovný <HlavnaVerzia>.<VedlajsiaVerzia>. Vedľajšia verzia projektu pozostáva z 2 čísiel. Príkladom názvu vetvy je 1.04.
2. Vytvorí adresár s názvom vetvy pod adresárom *branches*.
3. Prekopíruje celý obsah adresára *trunk* pod novovytvorený adresár s názvom vetvy.
4. Oboznámi programátorov s existenciou novej vetvy projektu.

Pri spájaní vedľajšej vetvy projektu s hlavnou sa manažér vývoja riadi nasledujúcim postupom:

1. Zvolí vedľajšiu vetvu, ktorú chce spojiť s hlavnou.
2. Spustí príkaz na spojenie vetiev.
3. V prípade konfliktov postupuje podľa kapitoly 3.6.
4. Vymaže vedľajšiu vetvu z adresára *branches*.
5. Oboznámi programátorov so spojením vedľajšej vetvy s hlavnou.



3.6. Riešenie konfliktov v zdrojovom kóde

Vstup: konflikt v zdrojovom kóde

Výstup: vyriešený konflikt v zdrojovom kóde

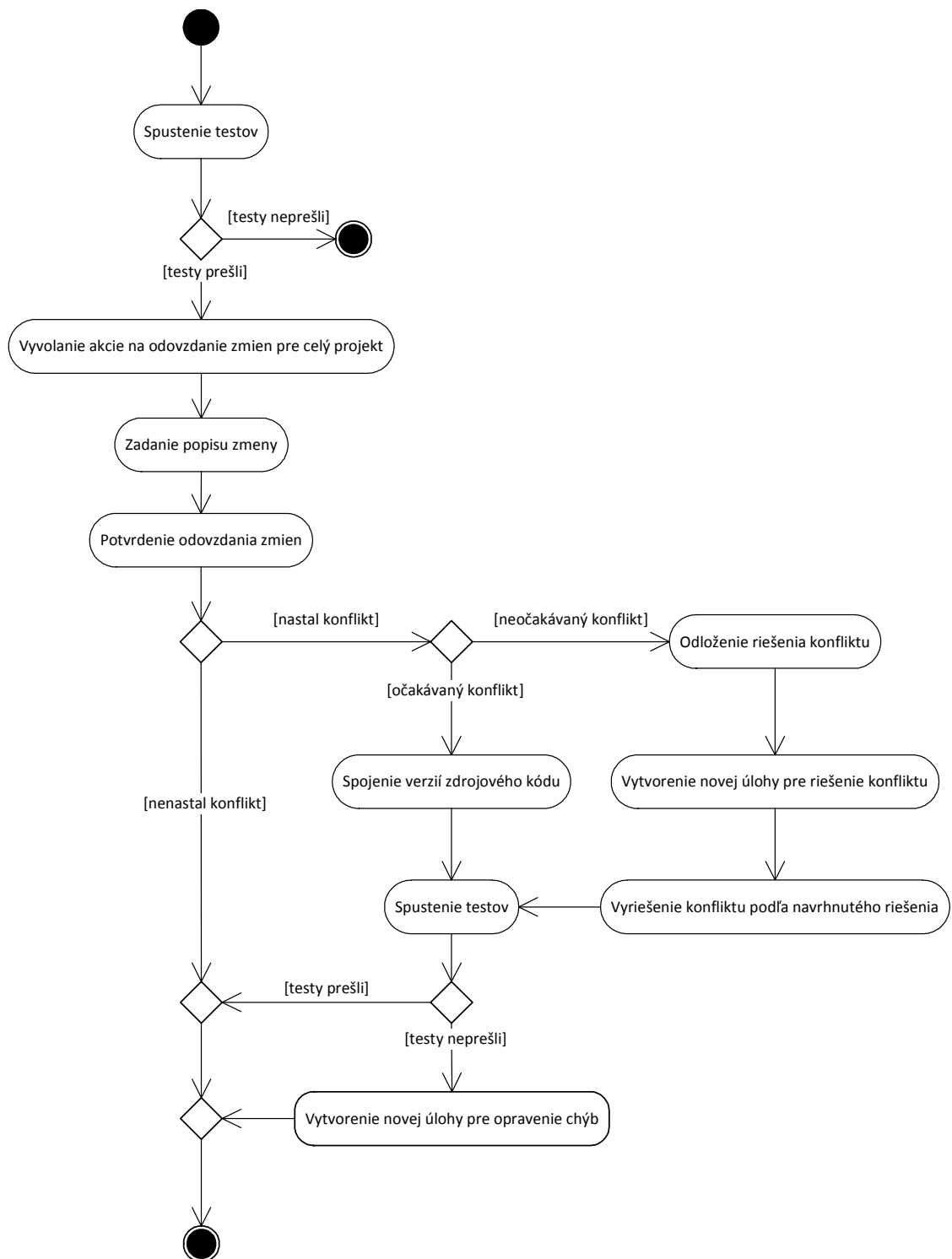
Zodpovedný: manažér vývoja, programátor

V prípade, že pri jednej z operácií dôjde ku konfliktu v zdrojovom kóde, je povinnosť tento konflikt vyriešiť alebo odložiť a kontaktovať manažéra vývoja (v prípade, že konflikt nenastane manažérovi vývoja). Takýto konflikt môže nastať buď manažérovi vývoja, alebo programátorovi. Zodpovedný za vyriešenie konfliktu je človek, u ktorého nastal konflikt. Postupuje sa nasledujúcim postupom:

1. Zistenie dôvodu konfliktu a porovnanie konfliktných súborov.
2. V prípade, že konflikt bol očakávaný z dôvodu práce viacerých programátorov na tej istej časti zdrojového kódu a rozdiely v rôznych verziách zdrojového kódu sa neovplyvňujú, pristúpi sa k spojeniu 2 verzií do výslednej tak, aby boli všetky zmeny zachované.
3. V opačnom prípade sa postupuje nasledovne:
 - a. Ak je zodpovedná osoba programátor, odloží konflikt a vytvorí novú úlohu v systéme pre správu úloh pre manažéra vývoja, ktorý tento konflikt vyhodnotí a zadá riešenie konfliktu. Programátor na základe zadaného riešenia vyrieši konflikt.
 - b. Ak je zodpovedná osoba manažér vývoja, rieši tento konflikt podľa vlastného uváženia a v prípade potreby kontaktuje zainteresovaných programátorov.

Odovzdanie zmien v zdrojovom kóde v prostredí Eclipse použitím pluginu Subclipse

Táto časť sa zaoberá odovzdávaním zmien v zdrojovom kóde v prostredí Eclipse použitím pluginu Subclipse. Činnosti, ktoré musí v tomto procese programátor vykonať zobrazuje nasledujúci diagram činností.



OBRÁZOK 2. DIAGRAM ČINNOSTÍ PRE PROCES ODOVZDANIA ZMIEN V ZDROJOM KÓDE



1. Spustenie testov

Pred samotným odovzdaním zmien v zdrojovom kóde sa programátor ubezpečí, či vykonané zmeny neovplyvnili funkčnosť jednotlivých súčiastok softvéru. Táto činnosť je dôležitá z pohľadu celkovej funkčnosti systému a zmeny, ktoré by ju narušili, nemôžu byť odovzdané do verziovacieho systému. Možný výsledok testov a následný postup programátora je zobrazený v nasledujúcej tabuľke.

TABUEKA 3. VÝSLEDOK TESTOV A POSTUP PROGRAMÁTORA

Výsledok	Postup programátora
Všetky testy prebehli úspešne	Programátor postupuje podľa kapitoly 2.
Aspoň jeden test neprebehol úspešne	Proces odovzdania zmien bude ukončený a programátor opraví chyby v zdrojovom kóde alebo v testoch. Následne opäť postupuje podľa kapitoly 1.

2. Vyvolanie akcie na odovzдание zmien pre celý projekt

Vykonané zmeny budú vždy odovzdávané v rámci celého projektu, keďže aj testy boli vykonávané pre celý projekt a predíde sa tým situáciám, kedy programátor zabudne odovzdať časť vykonaných zmien. Z tohto dôvodu programátor postupuje nasledovne:

1. Pravým tlačidlom myši klikne na projekt v prostredí Eclipse.
2. Zobrazí sa mu kontextová ponuka, kde prejde na položku *Team*.
3. Následne sa mu zobrazia podpoložky vybranej položky a zvolí položku *Commit...*
4. Zobrazí sa nové okno *Commit* a programátor postupuje podľa kapitoly 3.

3. Zadanie opisu zmien a ich potvrdenie

Zadanie opisu zmeny alebo viacerých zmien je nevyhnutné. Cieľom správneho opisu zmeny je najmä zabezpečiť prehľadnosť vo vykonaných zmenách v zdrojovom kóde. Záznam o zmene má stanovený formát, ktorý je nutné dodržiavať a je nasledovný:

[P:<NazovBalicka>][C:<NazovTriedy>][M:<NazovMetody>][T:<IdUlohy>] <OpisZmeny>

Význam parametrov a ich povinnosť je zobrazená v nasledujúcej tabuľke.

TABUEKA 4. PARAMETRE OPISU ZMENY V ZDROJOVOM KÓDE

Parameter	Význam	Povinnosť
<NazovBalicka>	Názov balíčka, v ktorom bola vykonaná zmena	povinný
<NazovTriedy>	Názov triedy v rámci balíčka, v ktorej bola vykonaná zmena	povinný v prípade, že existuje parameter <NazovMetody>
<NazovMetody>	Názov metódy v rámci triedy	dobrovoľný

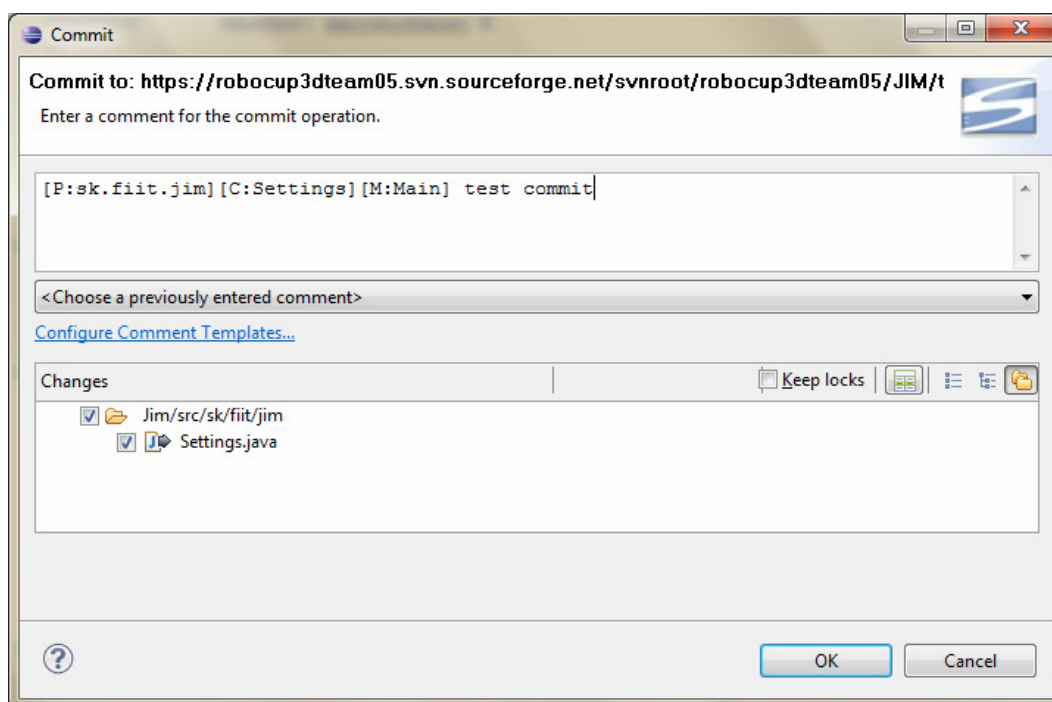


<IdUlohy>	ID úlohy zo systému pre správu úloh	povinný v prípade, že existuje úloha pre vykonanú zmenu
<OpisZmeny>	Opis, ktorý vystihuje vykonanú zmenu	povinný

V prípade, že boli vykonané zmeny vo viacerých balíčkoch, vytvorí sa viacero záznamov pod sebou oddelených novým riadkom. V prípade, že sa zmena týka viacerých úloh v systéme pre správu úloh, časť [T:<IdUlohy>] sa môže vyskytovať viac ako raz. Opis zmeny bude uvedený v angličtine.

Programátor postupuje nasledujúcim spôsobom:

1. Po zobrazení okna *Commit* zadá v hornej časti okna do textového poľa záznam o zmene podľa stanoveného formátu.
2. Okno bude následne vyzerať nasledovne:



OBRÁZOK 3. OKNO COMMIT PO ZADANÍ ZÁZNAMU O ZMENE

3. Stlačením tlačidla *OK* programátor potvrdí vykonané zmeny.



4. Riešenie konfliktov

Po potvrdení vykonaných zmien môže vzniknúť konflikt v zdrojovom kóde. Ak konflikt nevznikol, odovzdanie zmien prebehlo úspešne a proces je ukončený. V opačnom prípade je nutné takýto konflikt riešiť. Konflikt môže nastať z viacerých dôvodov:

1. Práca na neaktuálnom zdrojovom kóde.
2. Práca viacerých programátorov na tej istej časti zdrojového kóde.
3. Neuzamknutie súboru so zdrojovým kódom.

Cieľom je predísť vzniku konfliktov v zdrojovom kóde, keďže ich riešenie je často zložité. Z tohto dôvodu je nevyhnutné, aby bolo dodržiavané pravidlo o práci na aktuálnom zdrojovom kóde, pravidlo o čo možno najčastejšom odovzdávaní zmien a pravidlo, podľa ktorého je nutné uzamykať súbory so zdrojovým kódom, keď má programátor pracovať sám na časti zdrojového kódu.

V prípade, že konflikt nastal z nedodržania jedného z týchto pravidiel a teda konflikt je neočakávaný, programátor postupuje nasledovne:

1. Vyberie možnosť odloženia riešenia konfliktu a vytvorí novú úlohu v systéme pre správu úloh manažérovi vývoja s opisom konfliktu zodpovedajúcim opisu z verziovacieho nástroja. Názov úlohy bude začínať reťazcom [CF] a bude nasledovaný názvom konfliktu podľa uváženia programátora. Úloha bude mať vysokú prioritu. Týmto sa presúva zodpovednosť za riešenie konfliktu na manažéra vývoja.
2. Manažér vývoja vyhodnotí konflikt, v prípade nejasností kontaktuje zainteresovaných programátorov pre ozrejmienie dôvodu konfliktu a podľa vlastného uváženia zvolí riešenie, ktoré konflikt vyrieši, či už spojením 2 verzií zdrojového kódu s úpravami v zdrojovom kóde, úplným prepísaním zdrojového kódu v repozitári alebo zrušením nových zmien.
3. Manažér vývoja zmení zodpovednú osobu úlohy na osobu, ktorá úlohu vytvorila a pridá opis riešenia konfliktu spôsobom zobrazeným v nasledovnej tabuľke. Týmto sa presúva zodpovednosť za riešenie konfliktu späť na programátora.

TABUĽKA 5. ZÁPIS RIEŠENIA KONFLIKTU V ZDROJOVOM KÓDE

Riešenie konfliktu	Opis v systéme pre správu úloh	Príloha k úlohe
Spojenie 2 verzií zdrojového kódu	[m]	Výsledný zdrojový kód
Prepísanie zdrojového kódu v repozitári	[mf]	
Zrušenie nových zmien	[tf]	

4. Programátor dostane upozornenie o zmene stavu úlohy súvisiacej s konfliktom. Pristúpi k riešeniu konfliktu podľa nasledovnej tabuľky.



TABUĽKA 6. RIEŠENIE KONFLIKTU V ZDROJOVOM KÓDE

Opis v systéme pre správu úloh	Riešenie
[m]	Programátor zvolí akciu manuálneho spojenia verzií zdrojových kódov a do editora presunie zdrojový kód z prílohy pre danú úlohu.
[mf]	Programátor zvolí akciu, ktorá prepíše verziu v repozitári jeho verziou.
[tf]	Programátor zvolí akciu, ktorá zruší jeho vykonané zmeny.

V prípade, že konflikt bol očakávaný a bol vopred dohodnutý spoločný postup pri úprave zdrojového kódu, programátor pristúpi k manuálnemu spojeniu 2 verzií zdrojového kódu podľa vopred dohodnutého postupu. V prípade, že sa 2 verzie zdrojového kódu nedajú spojiť podľa očakávaného postupu, riešený konflikt sa zmení na neočakávaný a rieši sa spôsobom opísaným vyššie.

Po vyriešení konfliktu sa opäť spustia všetky testy a v prípade, že aspoň jeden z testov nebude úspešný, vytvorí sa úloha v systéme pre správu úloh pre opravenie chýb. Úloha bude mať štandardné parametre, ktoré sa používajú pri úlohe pre opravenie chyby.



Preberacie protokoly – časť D

Táto časť obsahuje preberacie protokoly tímu.



Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

POTVRDENIE O ODOVZDANÍ PROJEKTU

Tímový projekt I

Tím: Androids

Členovia tímu: Bc. Juraj Belanji

Bc. Miroslav Hruška

Bc. Roman Kováč

Bc. Andrej Minárik

Bc. Veronika Wolfová

Projekt: RoboCup 3D

Odovzdané dokumenty:

Dokumentácia k produktu v rozsahu 56 strán

Dokumentácia k riadeniu v rozsahu 37 strán

Pedagogický vedúci tímu, Ing. Ivan Kapustík, potvrdzuje prevzatie práce

Dňa:

člen tímu

Ing. Ivan Kapustík



Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

POTVRDENIE O ODOVZDANÍ PROJEKTU

Tímový projekt I

Tím: Androids

Členovia tímu: Bc. Juraj Belanji

Bc. Miroslav Hruška

Bc. Roman Kováč

Bc. Andrej Minárik

Bc. Veronika Wolfová

Projekt: RoboCup 3D

Odovzdané dokumenty:

Dokumentácia k produktu v rozsahu 81 strán

Dokumentácia k riadeniu v rozsahu 88 strán

Pedagogický vedúci tímu, Ing. Ivan Kapustík, potvrdzuje prevzatie práce

Dňa:

člen tímu

Ing. Ivan Kapustík