

FAKULTA INFORMATIKY
A INFORMAČNÝCH TECHNOLOGIÍ

Tímový projekt
Imagine Cup

<i>Vedúci projektu:</i>	<i>prof. Ing. Mária Bieliková PhD.</i>	<i>Autori :</i>	<i>Bc. Andrej Frlička</i>
<i>Študijný odbor</i>	<i>Softvérové inžinierstvo</i>		<i>Bc. Marek Tomša</i>
<i>Akademický rok:</i>	<i>2006/2007</i>		<i>Bc. Richard Veselý</i>
<i>Ročník:</i>	<i>1. Inžiniersky</i>		<i>Bc. Oto Vozár</i>
<i>Semester:</i>	<i>Letný</i>		

ANOTÁCIA

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ
Študijný program: Softvérové inžinierstvo

Autori: Andrej Frlička
Marek Tomša
Richard Veselý
Oto Vozár

Tímový projekt: Imagine cup
Vedenie tímového projektu: prof. Ing. Mária Bieliková PhD.
Máj 2007

Tvorba vzdelávacích systémov s presne vymedzenou funkcionalitou je dnes pomerne bežný jav. Väčšina z vytvorených produktov sa zameriava na úzku doménu vedomostí a tým aj úzku skupinu používateľov. Systém, ktorý navrhujeme má za cieľ poskytnúť vzdelanie deťom, ktoré prekročili brány základnej školy spôsobom, ktorý sa plne prispôsobí ich naturelu svojim obsahom aj formou.

Hra je formou a prostriedkom, ktorý presvedčí deti, aby sa učili. Preto ich v našom systéme zavedieme do herného sveta, kde budú spolu so svojim zvieratkom objavovať nové poznatky a tým rásť na poli poznania.

Dokument poskytuje porovnanie systémov, ktoré majú podobné zameranie alebo ktorých zložky sú podobné. Z porovnania vyplýva rámcová verzia požiadaviek na systém a požiadaviek na vývoj. Všetky analýzy sú pretavené v hrubý návrh herného systému, ktorý má byť novým pohľadom na vzdelávanie najmladších, a ktorý je zároveň podkladom ku vypracovanému prototypu.

Posledná časť dokumentu zahŕňa doplnenie špecifikácie a vypracovanie podrobného návrhu systému, a následne podáva správu o implementácii a testovaní výsledného produktu.

ANNOTATION

Slovak University of Technology, Bratislava
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES
Degree Course: Software engineering

Authors: Andrej Frlička
 Marek Tomša
 Richard Veselý
 Oto Vozár

Team project thesis: Imagine cup
Supervisor: prof. Ing. Mária Bieliková PhD.
Máj 2007

Creation of educational systems with specific functionality is quite common nowadays. Most of the existing products target small knowledge domain and in turn also small group of users. Our proposed system aims to educate children and adapt its form and content fully according to their personality and needs.

Games can act as a tool to persuade children to learn. Therefore we have included them in our game world, where the children explore and discover new things together with their pet companions while learning in the process.

Document offers comparison of systems, which have similar scope or set of components. Comparison also serves as a basic version of the set of system and development requirements. Our analysis is gradually transformed into rough draft of gaming system, which is used in process of prototype implementation.

Last part of the document comprises extended specification and detailed system design, and reports about implementation and testing of resulting product.

PREDSLOV

V modernej spoločnosti je dobré vzdelanie skoro až nevyhnutnosťou. Pri uchádzaní sa o lepšie ohodnotené pracovné miesta musia dnes ľudia ponúkať vysokú mieru kvalifikácie a odbornosti. A to sa nedá nadobudnúť z jedného dňa na druhý, ale len neustálym a systematickým sebazdokonaľovaním.

Istý múdry filozof raz povedal, že cesta dlhá tisíc míľ začína prvým krokom. A sú to práve prvé kroky, ktoré sú pre budúce smerovanie človeka najdôležitejšie. Veď práve obdobie školského veku je odrazovým mostíkom s prvými schodíkmi a prekážkami, ktoré musí mladý človek prekonať v úsilí za dosiahnutím svojich cieľov. No je to aj obdobím formovania si návykov a prístupu ku vzdelaniu, čo je ešte dôležitejšie.

A pritom, už od útleho detstva sa človek prirodzene učí. Nie cielene a formálne, ale skúmaním sveta a hrou. Jeho zvedavosť ho ženie k interakcii s vecami a ľuďmi okolo neho. Preto aj neskôr prijíma procesy, ktoré v sebe zahrňujú tieto prvky s oveľa väčším záujmom, ako len nejakú nudnú prezentáciu faktov.

Cieľom nášho projektu je navrhnúť systém pre deti mladšieho školského veku, ktorý by im umožnil vzdelávať sa prostredníctvom interaktívneho herného sveta. V ňom by sa mohli stretávať s ostatnými hráčmi a predmetmi a učiť sa prostredníctvom rôznych interaktívnych hier, prispôsobujúcich svoju náročnosť ich výkonom a vedomostiam. S postupom času ako by vzdelanostne rástli by sa stávali vo virtuálnom svete silnejšími, čo by ich motivovalo k ďalšej snahe a ovplyvňovalo ich možnosti v hre. A samozrejme aj v reálnom svete, čo je našou prvoradou snahou.

Projekt vznikol v rámci predmetu Tvorba softvérového systému v tíme s cieľom zúčastniť sa súťaže Imagine Cup 2007, ktorej hlavná téma je formulovaná vetou : „Predstavte si svet, kde by technológie dokázali zabezpečiť vzdelanie pre každého“. Vedúcou projektu je prof. Ing. Mária Bieliková PhD. a členmi tímu sú v abecednom poradí Andrej Frlička, Marek Tomša, Richard Veselý a Oto Vozár.

Dokumentácia k nemu je rozdelená na technickú dokumentáciu časť riadenia. V technickej dokumentácii sa nachádza samotný popis projektu chronologicky po jednotlivých etapách jeho vytvárania a v časti riadenia je opísaný samotný proces, ktorého výstupom je tento projekt.

ZADANIE TÍMOVÉHO PROJEKTU

Súťaž Imagine Cup, ktorú organizuje firma Microsoft si za niekoľko rokov získala pomerne veľa priaznivcov na univerzitách z celého sveta a stala sa vynikajúcim prostriedkom na zmeranie síl študentov pripravujúcich sa v profesii softvérového inžiniera. Súťaž má viacero kategórií, pre nás bude zaujímavá kategória "softvérový návrh". Témou ročníka 2007 je:

„Imagine a world where "technology enables a better education for all".

Téma je veľmi zaujímavá a aktuálna. Možno využiť a nadviazať aj na výskumné projekty, ktoré sa riešia na Ústave informatiky a softvérového inžinierstva, konkrétne v súvislosti s adaptívnymi webovými systémami v rôznych doménach a medzi nimi aj pre e-vzdelávanie. Úlohou je použiť tvorivosť, kreativitu a technické znalosti na riešenie toho, čo považujú za aktuálne problémy v rámci danej témy pomocou Microsoft nástrojov a technológií. Oceňujú sa nápady, kuriózne riešenia a talent pri vytváraní použiteľných softvérových aplikácií. V rámci tímového projektu bude treba:

- oboznámiť sa s projektami s minulých rokov, pravidlami a duchom súťaže
- vymyslieť nápad, ktorý bude ukazovať kreativitu študentov
- nápad realizovať formou softvérového prototypu v rámci ohraničení a podmienok súťaže Imagine Cup 2007

Súťaž je určená pre štvorčlenné tímy študentov, a keďže štandardne vytvárame 5-6 členné tímy, je nevyhnutné prejsť záujem spracovaním ponuky v skrátenom termíne, aby bolo možné ešte pred uzavretím zloženia tímov vybrať štvoricu, ktorá bude riešiť túto úlohu.

OBSAH

1	ÚVOD.....	4
1.1	ÚČEL A ROZSAH DOKUMENTU	4
1.2	PREHLAD DOKUMENTU.....	4
1.3	SLOVNÍK POJMOV PROBLÉMOVEJ OBLASTI.....	4
1.4	SKRATKY.....	5
1.5	ODKAZY A ZDROJE	5
2	OPIS RIEŠENÉHO PROBLÉMU	6
2.1	PREHLAD PROBLÉMOVEJ OBLASTI	6
2.1.1	Výukové hry.....	6
2.1.2	Výukový softvér.....	8
2.2	ANALÝZA ŠKOLSKÝCH OSNOV.....	9
2.3	ANALÝZA ONLINE HIER	10
2.4	CIELE PRODUKTU	11
2.5	VLASTNOSTI PRODUKTU.....	12
3	ŠPECIFIKÁCIA RÁMCA.....	13
3.1	PRINCÍP HRY	13
3.2	SVET.....	13
3.2.1	Viditeľný herný svet	13
3.2.2	Neviditeľný herný svet.....	14
3.2.3	Účastníci.....	14
3.2.4	Interakcie.....	16
3.2.5	Hodnotenie	17
3.2.6	Logovanie a štatistiky	18
3.3	KOMUNIKÁCIA	18
3.4	HRY.....	19
3.5	QUESTY (ÚLOHY).....	19
3.5.1	Línia učenia sa	19
3.5.2	Preverovacie questy	19
3.6	SCENÁRE POUŽITIA.....	20
3.6.1	Hráč.....	20
3.6.2	Mentor.....	21
3.7	POUŽÍVATEĽSKÉ ROZHRANIE.....	21
4	ŠPECIFIKÁCIA ČASTÍ SYSTÉMU	22
4.1	HERNÝ SVET.....	22
4.2	GADGET	22
4.3	PDA.....	22

4.4	NÁSTROJ PRE RODIČOV A UČITEĽOV	22
5	ANALÝZA TECHNOLOGIÍ PRE VÝVOJ.....	23
5.1	GRAFICKÁ REPREZENTÁCIA.....	23
5.1.1	OpenGL.....	23
5.1.2	DirectX.....	23
5.1.3	WPF (Windows Presentation Foundation)	23
5.1.4	Microsoft XNA	24
5.2	SIEŤOVÁ KOMUNIKÁCIA	24
5.2.1	Sockety.....	24
5.2.2	.NET remoting	24
5.2.3	WCF (Windows Communication Foundation).....	24
5.3	MANIPULÁCIA SO ZVUKOM.....	25
5.3.1	Wavein/waveout API	25
5.3.2	DirectSound	25
5.3.3	XACT.....	25
6	HRUBÝ NÁVRH	26
6.1	NÁVRHOVÉ ROZHODNUTIA.....	26
6.1.1	Atribúty návrhu	26
6.1.2	Model komunikácie.....	27
6.1.3	Štruktúra systému.....	27
6.2	ARCHITEKTÚRA SYSTÉMU	28
6.2.1	Infraštruktúrne služby	29
6.2.2	Služby aktivít	29
6.2.3	Dátové služby.....	29
6.2.4	Služby doménových znalostí	29
7	PROTOTYP	30
7.1	IMPLEMENTÁCIA ČASTÍ PROTOTYPU	30
7.1.1	Server	30
7.1.2	Herný klient.....	33
7.1.3	Editor máp.....	36
7.1.4	PDA klient.....	39
7.1.5	Zhrnutie.....	41
7.2	ZÁVERY PROTOTYPOVANIA	41
8	SPRESNENIE A DOPLNENIE ŠPECIFIKÁCIE.....	42
8.1	ZHODNOTENIE PROTOTYPOVANIA	42
8.2	HERNÉ AKTIVITY	42

8.2.1	Strokes.....	42
8.2.2	SmartLines	43
8.2.3	Matematické balóny	44
8.2.4	Zvuky	44
8.2.5	Opakovanie melódie	45
8.2.6	Malý hudobník	45
8.3	ODDYCHOVÉ AKTIVITY	46
8.3.1	Čítanie rozprávok.....	46
8.3.2	Dlhé slová	47
8.4	SLUŽBA ZVIERATKO.....	47
9	PODROBNÝ NÁVRH	48
9.1	SLUŽBY	48
9.1.1	Služba svet	48
9.1.2	Rámec pre vytváranie služieb aktivít	54
9.1.3	Rámec pre vytváranie úložísk doménových znalostí	55
9.1.4	Služba StoryTeller.....	56
9.1.5	Služba Zvieratko	56
9.2	METÓDY A PRÍSTUPY	57
9.2.1	Item Response Theory	57
9.2.2	Metóda častejšieho opakovania nesprávne zodpovedaných otázok	58
9.2.3	Stroke metrika	59
9.2.4	Rozoznávanie výšky tónov a opakovanie melódie	60
9.2.5	Adaptívnosť pri SmartLines.....	60
9.3	APLIKÁCIE.....	61
9.3.1	Aplikácia hostujúca služby.....	61
9.3.2	Editor máp.....	61
9.3.3	Herný klient.....	65
9.3.4	Hra Strokes.....	75
9.3.5	Hra Matematické balóny	76
9.3.6	Hra Zvuky	77
9.3.7	Hra Opakovanie melódie	78
9.3.8	Hra Malý hudobník	79
9.3.9	Hra SmartLines	80
9.3.10	Zvieratko v PDA	83
9.3.11	Zvieratko	86
9.3.12	Zvieratko v Sidebar Gadgete.....	87

10	IMPLEMENTÁCIA	89
10.1	VÝBER IMPLEMENTAČNÉHO JAZYKA A PROSTREDIA	89
10.2	KOMPONENTY SYSTÉMU	89
11	OVERENIE RIEŠENIA	96
11.1	TESTOVANIE	96
11.1.1	Plán testovania	96
11.1.2	Testovacie dáta	96
12	ZÁVER	101
A	POUŽÍVATEĽSKÁ PRÍRUČKA	1
A.1	HLAVNÉ MENU	1
A.2	ZVIERATKO	1
A.3	HERNÉ AKTIVITY	2
A.3.1	Obťahovanie	2
A.3.2	Matematické balóny	4
A.3.3	Zvuky	4
A.3.4	Opakovanie melódie	5
A.3.5	Malý hudobník	6
A.3.6	Chytré čiary	7
A.3.7	Rozprávky	9
A.3.8	Dlhé slová	9
A.4	HERNÝ KLIENT	10
A.5	PDA KLIENT	13
B	SYSTÉMOVÁ PRÍRUČKA	1
B.1	SYSTÉMOVÉ PREREKvizITY	1
B.2	POSTUP INŠTALÁCIE	1
B.2.1	Postup inštalácie na klientovi	1
B.2.2	Postup inštalácie na PDA	2
B.2.3	Postup inštalácie na serveri	2
B.2.4	Konfigurácia na serveri	2
B.2.5	Konfigurácia na klientovi	2
B.3	NAPŔŇANIE DÁTAMI	3
B.3.1	Manipulácia s dátami v službách doménových znalostí	3
B.3.2	Použitie editorov	4
C	OBSAH MÉDIA	1

1 Úvod

Kapitola popisuje základné informácie o dokumente výsledkov projektu. Predstavuje účel a rozsah dokumentu, ponúka slovník pojmov problémovej oblasti prehľad použitých skratiek, informáciu o použitých zdrojoch a prehľad odkazov a zdrojov.

1.1 Účel a rozsah dokumentu

Dokument slúži ako podkladový materiál ku tvorbe a hodnoteniu tímového projektu. Rozsahom kopíruje požiadavky plynúce z obsahu vyžadovaného na odovzdanie v prvom a druhom kontrolnom bode. Je kostrou analýzy problémovej oblasti a poskytuje základné piliere pre ďalší návrh. Informuje o výsledkoch implementácie prvého prototypu

1.2 Prehľad dokumentu

Informácie, ktoré sú poskytnuté v dokumente je možné rozdeliť do nasledujúcich ucelených kapitol:

Tabuľka 1-1: Kapitoly vystupujúce v systéme

Kapitola	Popis kapitoly
Úvod	Popisuje základné syntaktické a sémantické vlastnosti dokumentu
Opis riešeného problému	Pojednáva o problémovej oblasti a postavení produktu v nej. Zhŕňa požiadavky kladené na projekt a ciele projektu.
Špecifikácia rámca	Podrobne popisuje požiadavky kladené na informačný systém.
Špecifikácia častí systému	Bližšie popisuje jednotlivé súčasti systému .
Analýza technológií pre vývoj	Analyzuje vývoj z pohľadu technológií. Ponúka analýzu niektorých uvažovaných technológií , navzájom ich porovnáva a zdôvodňuje vhodnosť ich použitia v procese vývoja.
Hrubý návrh	Poskytuje pohľad na architektúru systému v hrubých obrysoch, a zdôvodňuje rozhodnutia, ktoré boli vzaté v úvahu pri procese tvorby hrubého návrhu.
Prototyp	Informuje o výsledkoch implementácie 1. prototypu ku druhému kontrolnému bodu
Spresnenie a doplnenie špecifikácie	Rozširuje špecifikáciu o popis herných aktivít
Podrobný návrh	Poskytuje pohľad na architektúru komponentov systému
Implementácia	Dokumentuje vytvorené projekty v systéme
Overenie riešenia	Opisuje overovanie riešenia. Zhŕňa postup overovania

1.3 Slovník pojmov problémovej oblasti

peer-to-peer komunikácia – Komunikácia medzi dvomi uzlami, ktoré nemajú prostredníka. V systéme bude s jej pomocou realizovaná audiovizuálna komunikácia.

dungeon – V hrách predstavuje dungeon-“jaskyňa“ nebezpečnú oblasť, ktorú môžu hráči skúmať, v ktorej môžu bojovať a obvykle v nej nachádzajú rôzne poklady a odmeny.

quest - úloha, ktorú hráči dostávajú, ktorá má buď postranný význam, alebo posúva dejovú líniu. Obvykle býva sprevádzaná istým motivujúcim prvkom, a jej naplnenie znamená pre hráča zisk.

avatar – virtuálna postava, ktorá reprezentuje fyzickú osobu, ktorá ju riadi v hernom svete. Avatari majú obvykle niektoré vlastnosti, ktoré fyzické osoby nemajú.

kontrakt – legálne viazaná výmena sľubov medzi stranami

1.4 Skratky

WCF – Windows Communication Foundation je moderné komunikačné rozhranie vytvorené firmou Microsoft, ktoré je nástupcom technológie webových služieb, ktorých koncept rozširuje.

MMORPG - Massively Multiplayer Online Role Playing Game. Hra na hrdinov, v ktorej hráči prežívajú osudy alternatívnych postáv, postupne sa snažia presadiť v rozsiahlom hernom svete a komunikujú resp. bojujú s ostatnými hráčmi.

NPC – Postava v hre ktorá nie je riadená hráčom (nehráčska postava)

1.5 Odkazy a zdroje

1. CALTAGIRONE, S., KEYS, M., SCHLIEF, B., AND WILLSHIRE, M. J. 2002. Architecture for a massively multiplayer online role playing game engine. J. Comput. Small Coll. 18, 2 (Dec. 2002), 105-116.
2. KARDAN, K. 2006. Computer role-playing games as a vehicle for teaching history, culture, and language. In Proceedings of the 2006 ACM SIGGRAPH Symposium on Videogames (Boston, Massachusetts, July 30 - 31, 2006). sandbox '06. ACM Press, New York, NY, 91-93. DOI= <http://doi.acm.org/10.1145/1183316.1183328>
3. KIM, J., CHOI, J., CHANG, D., KWON, T., CHOI, Y., AND YUK, E. 2005. Traffic characteristics of a massively multi-player online role playing game. In Proceedings of 4th ACM SIGCOMM Workshop on Network and System Support For Games (Hawthorne, NY, October 10 - 11, 2005). NetGames '05. ACM Press, New York, NY, 1-8. DOI= <http://doi.acm.org/10.1145/1103599.1103619>
4. PAPARGYRIS, A. AND POULYMENAKOU, A. 2005. Learning to fly in persistent digital worlds: the case of Massively Multiplayer Online Role Playing Games. SIGGROUP Bull. 25, 1 (Jan. 2005), 41-49. DOI= <http://doi.acm.org/10.1145/1067699.1067706>

2 Opis riešeného problému

2.1 Prehľad problémovej oblasti

V počítačovom vzdelávaní existujú dve skupiny produktov: Výukové hry, ktorých cieľom je vzdelávanie formou hry a programy, ktoré možno označiť ako výukový softvér, ktorý priamo podporuje štandardný proces učenia. Podrobnejší opis sa nachádza v nasledujúcich dvoch podkapitolách.

V ďalšej časti vystupuje do popredia analýza školských osnov pre prvý stupeň základných škôl a aj možnosť realizácie jednotlivých vyučovacích disciplín pomocou hier s ohľadom na pútavosť ale aj užitočnosť pre daný predmet.

Projekt sa tematicky dotýka nielen oblasti počítačového vzdelávania ale aj domény hromadných online hier (MMORPG) a preto sa posledná podkapitola zaoberá opisom ich základných princípov a konceptov.

2.1.1 Výukové hry

Pri analýze sme sa zamerali na hry s výukovým charakterom, ktoré sú určené pre deti prvého stupňa základných škôl. Z hľadiska realizácie môžeme hry rozdeliť na desktopové a webové aplikácie.

Desktopové aplikácie

Predstavujú ich „krabicové“ produkty, ktoré je možné bežne kúpiť v obchode a nainštalovať na počítač. Hry sú väčšinou tematicky zamerané na vybranú mentálnu oblasť akou je precvičovanie pamäti, predstavivosti, alebo sa priamo viažu na niektorý školský predmet napríklad na písanie, čítanie alebo vlastivedu. Okrajovo je možné k nim priradiť aj výukové encyklopédie s lákavými témami (dinosoury alebo vesmír).

Softvér sa obvykle skladá z väčšieho množstva minihier, z ktorých každá sa zameriava na určitý výsek domény a ktoré nie sú alebo sú len voľne prepojené. Charakterovo sa niektoré podobajú simulácii, kde môže dieťa skúšať používať rôzne predmety ako napríklad mikroskop alebo skúmavky.

U všetkých produktov, ktoré sme analyzovali nebola zabezpečená podpora hrania pre viacerých hráčov prostredníctvom siete, prevažuje ovládanie jedným alebo dvoma hráčmi sediacimi za jedným počítačom.

Nasleduje opis niektorých analyzovaných hier.

Pamäť a logika

Hra je zameraná na precvičovanie logického myslenia a pamäte, kde náročnosť jednotlivých úloh rastie postupne od najľahších až ku zložitejším.

Začíname s vlastivedou

Produkt je tematicky viazaný priamo na predmet vyučovaný na školách, a používa sa ako doplnok ku školskému učivu. Venuje sa zemepisu a dejinám vlasti a informácie sprístupňuje dieťaťu pomocou multimediálnych prostriedkov.

Animal Hospital: Pet Vet 3D

Program je príkladom simulačnej hry, ktorá stavia hráča do pozície veterinára a zábavnou formou buduje jeho znalosti o zvieratách a vzťah k nim. Pohľad do hry sa nachádza na obrázku (Obrázok 2-1).



Obrázok 2-1 Obrázok z hry Animal Hospital

Webové aplikácie

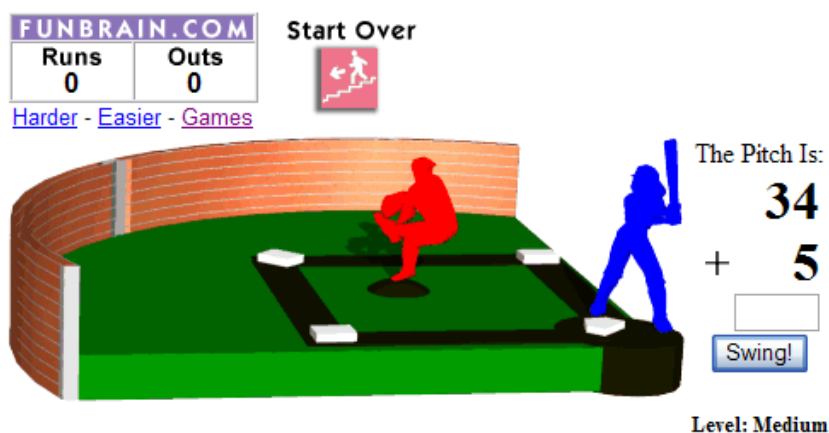
Existuje ich pomerne veľké množstvo variabilného rozsahu a kvality. Zvyčajne sú zoskupené na samostatných portáloch určených výhradne pre deti alebo len ako sekcia v portáloch s viacerými druhmi hier. Výhodou riešenia je, že sú dostupné bez inštalácie a sú takmer všetky voľne prístupné. Nevýhoda spočíva hlavne v obmedzenej možnosti interakcie s používateľom a v oslabení prezentácie informácií.

Šírka precvičovanej oblasti býva väčšinou užšie profilovaná ako pri desktopových aplikáciách, rozsahovo približne len ako jedna z jej minihier. Ďalším rozdielom oproti prvej skupine je neexistencia priamej väzby na vyučovací predmet.

Nasleduje opis niektorých analyzovaných hier:

Math Baseball

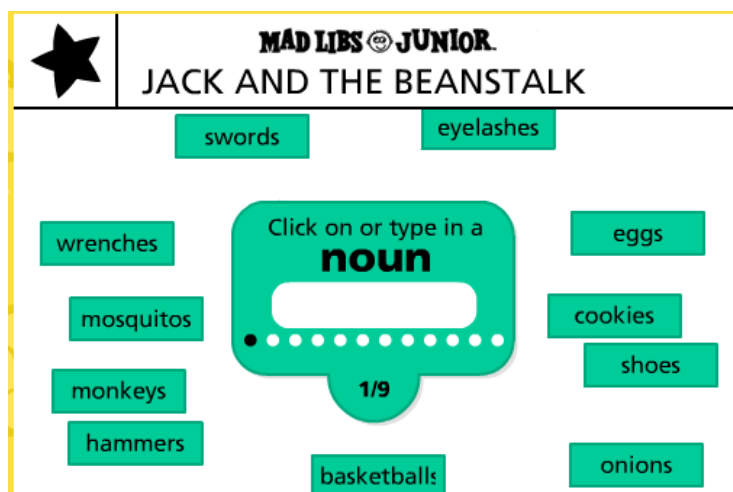
Jednoduchá hra, v ktorej hráč ovplyvňuje hru pomocou výsledkov výpočtu matematických príkladov s nastaviteľnou obtiažnosťou (Obrázok 2-2)



Obrázok 2-2 Obrazovka z hry Math Baseball

Jack and the Beanstalk

Hra sa zameriava na výber správneho slovného druhu z navrhovaných slov (Obrázok 2-3)



Obrázok 2-3 Obrazovka z hry Jack and Beanstalk

2.1.2 Výukový softvér

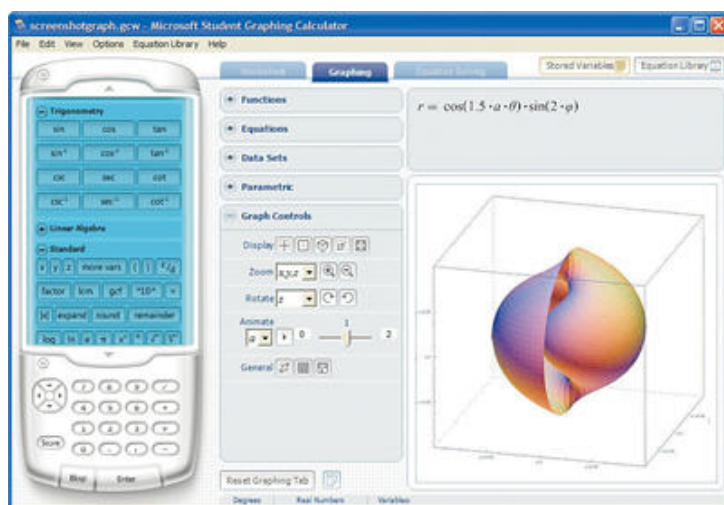
Ťažiskom sú hlavne programy uľahčujúce proces učenia v reálnom živote. Príkladom môže byť podpora vyhľadávania informácií, tvorba domácich úloh alebo vizualizácia určitých informácií hlavne v matematickej doméne. Pri analýze nás zaujalo hlavne riešenie Microsoft Student.

Microsoft Student

Softvér je rozdelený na niekoľko častí.

- Microsoft Math: Zaoberá sa rovnicami, plošnou geometriou (triangle solver), prevodmi jednotiek a je možné si doobjednať službu asistencie pri domácich úlohách. Prostredie je znázornené na obrázku (Obrázok 2-4)
- Language and Literature Tools: Obsahuje veľké množstvo citátov a obsahov najznámejších literárnych diel, postupy ako písať práce a nástroje pre prácu s cudzími jazykmi.

- Research Tools: Funguje v spolupráci s produktom Microsoft Encarta kde pomáha vyhľadávať a spracovať požadované informácie.
- Šablóny, návody a nástroje: Pomôcky pre vytváranie dokumentov a prezentácií



Obrázok 2-4 Obrazovka časti prostredia Microsoft Math

2.2 Analýza školských osnov

Po preštudovaní syllabov prvého stupňa základných škôl považujeme za perspektívne z hľadiska tvorby interaktívnych hier nasledujúce výukové oblasti:

Doména Slovenského jazyka:

Výcvik čítania:

- počúvanie čítaného textu
- očná gymnastika

Ľudová slovesnosť - riekanky

- vyčítanky,
- rapotanky,
- hádanky,
- rozprávky (úpravy)

Doména Prvouky:

- Poznávanie rastlín, plodov a semien

Doména Matematiky:

- Prirodzené čísla 1 až 5, numerácia
- Sčítanie a odčítanie prirodzených čísel v obore do 5
- Čísla 6, 0, 7, 8, 9, 10, sčítanie a odčítanie
- Numerácia v obore prirodzených čísel do 20. Sčítanie a odčítanie
- Geometria
- Riešenie rovníc
- Získanie prvých skúseností z číselnou osou
- Riešenie nepriamo sformulovaných úloh

Doména Informatickej vedy

- práca s detskými programami
- práca s grafikou
- práca s internetom
- jednoduché programy
- hry

Doména Výtvarná výchova

- Farba
- Základy konštruovania

Doména Hudobná výchova

- V rôznom tempe a dynamike deklamovať, akusticko-hmatovým pohybom realizovať jednotlivé rytmy v 2-, 3-, 4-štvrt'ovom takte
- Sluchom rozlíšiť rôzne zvuky, tón vyšší, nižší, rovnaký, ukončenosť a neukončenosť melódie, totožnosť, podobnosť, rozdielnosť, zo známych rytmicko-melodických útvarov riešiť rôzne hádanky
- Sluchom rozoznať náladový, dynamický, tempový kontrast, klavír, husle, flautu, kontrabas, detský a miešaný zbor, orchester, rozlíšiť melódiu a sprievod
- Riešiť rytmické a melodické dopĺňovačky. K danému rytmu tvoriť opačný rytmus (podľa dohody, napr. k štvrt'ovej note sú opačné 2 osminové

2.3 Analýza online hier

Základným princípom online hier je existencia herného sveta v ktorom sa nachádza vysoké množstvo ľuďmi kontrolovaných postáv. Každý účastník ovláda jednu postavu, ktorá ho reprezentuje. Obvyklý názov pre herného hrdinu je avatar. Hráči môžu pomocou nich interagovať s predmetmi vo svete, navzájom medzi sebou alebo s počítačom riadenými postavami.

Herné dianie spočíva najmä v súbojoch postáv, pričom jednotlivé postavy môžu disponovať rôznymi ochrannými, útočnými a podpornými prostriedkami definovanými svojim typom a silou postavy. Súboj môžu taktiež do istej miery ovplyvniť predmety, ktoré má avatar pri sebe.

Motiváciou pre hranie takéhoto typu hier je hlavne **rozvoj postavy**, ktorý sa niekedy kvantifikuje číslami v podobe skúsenostných bodov alebo úrovní a ostatnými číselnými štatistikami (Obrázok 2-6). Jej sila a možnosti rastú v prevažnej miere s počtom súbojov, ktoré absolvovala. Ľudský hráč okrem zvládnutia ovládania a orientácie vo svete nezdokonaľuje žiadne svoje schopnosti ani znalosti a väčšinu času zvyšuje úroveň svojej postavy množstvom súbojov. Činnosť môže často prerásť až do stereotypu, čo býva problémom veľkého množstva takýchto hier.

Najzaujímavejším aspektom online hier je práve interakcia medzi hráčmi a vznik určitého sociálneho prostredia, vzťahov a komunít.

Na obrázku (Obrázok 2-5) sa nachádza obrazovka hry Rose Online.



Obrázok 2-5 Obrazovka hry Rose Online



Obrázok 2-6 Číselne kvantifikované vlastnosti postavy a predmety inventára v hre Diablo

2

2.4 Ciele produktu

Hlavným cieľom produktu je priniesť jeho hlavným používateľom, čiže učiacim sa deťom v predškolskom alebo rannom školskom veku, nasledujúce možnosti:

- Spojenie učenia sa a zábavy, čiže získavanie znalostí a zručností pomocou hier

- Prispôsobenie úrovne zložitosti v závislosti od vstupov hráča tak, aby nebola náročnosť hry privysoká.
- Vytvorenie pozitívneho vzťahu k učeniu a vedomostiam
- Rozvoj osobnostných vlastností ako súťaživosť, zodpovednosť, kooperácia a sebavedomie
- Popri schopnosti učiť sa poskytnúť možnosť učiť niekoho iného a získať pohľad na výučbový proces aj z opačnej strany
- Poskytnúť dieťaťu spätnú väzbu ako dobré sú jeho znalosti v porovnaní s ostatnými ako aj motiváciu tieto znalosti rozvíjať
- Možnosti spoznania sa, komunikácie a interakcie s inými deťmi
- Vyplnenie voľného času zmysluplnými aktivitami rozvíjajúcimi dieťa
- Zvládnutie ovládania počítača pomocou štandardných vstupov (myš, klávesnica) ako aj pre deti prístupnejšími vstupmi (hlas, pohyb tela)

Okrem toho poskytuje rodičom alebo učiteľom

- Možnosť kontroly znalosti a nastavenia určitých parametrov hry
- Kooperáciu s dieťaťom v rámci hry

2.5 Vlastnosti produktu

Špecifikom produktu má byť práve kombinácia pozitívnych vlastností z oblasti výučby ako aj z oblasti herného priemyslu. Kombináciou týchto dvoch aspektov vzniká niekoľko zaujímavých vlastností, ktoré sa podpisujú pod celkový charakter produktu.

Inherentné vlastnosti (Vyplývajúce z domén)

- Produkt je vysoko motivujúci: Faktor nepoznaného, zábavy a spoločnosti znamená vždy zvýšenie motivácie zotrvať.
- Produkt je veľmi návykový: Dieťa si rýchlo zvykne, že herný svet mu poskytne veľa možností a smerov, v ktorých sa môže pohybovať, pomerne rýchlo si vytvorí návyk vzdelávať sa takýmto spôsobom.
- Produkt hráča rozvíja: Dieťa sa nevedomky svojou existenciou v hernom svete rozvíja či už v sociálnych interakciách alebo v jednotlivých doménových oblastiach.

Požadované vlastnosti (Vyplývajúce z ďalšej špecifikácie)

- Dieťa musí byť s produktom v neustálom kontakte: Produkt nesmie byť „zadrôtovaný“ len na jedno konkrétne umiestnenie. Dieťa ku nemu musí mať neustále prístup.
- Rodič (učiteľ) musí mať možnosť upravovať správanie systému podľa potrieb dieťaťa.

3 Špecifikácia rámca

Kapitola zhrňa špecifikáciu rámca systému. Je rozdelená na tri podkapitoly. Prvá opisuje základný princíp hry, druhá špecifikuje herný svet a ďalšia možnosti komunikácie medzi účastníkmi hry. Ďalšie podkapitoly uvádzajú koncept podhier a questov v rámci špecifikovanej hry. Nasleduje popis priebehu hry z pohľadu hráča a mentora vo forme scenárov použitia. Posledná podkapitola špecifikuje používateľské rozhrania aplikácie.

3.1 Princíp hry

Výučba je realizovaná prostredníctvom hry typu MMORPG. Herný svet je rozdelený na podsvety, ktoré sú navzájom prepojené. Hráči sa pohybujú po svete a vidia avatarov, ostatných hráčov, s ktorými môžu komunikovať. Prechádzajú líniami učenia sa, môžu tvoriť s ostatnými hráčmi tímy, získavať vyznamenania a skúsenosti, ktoré môžu vymeniť za iné druhy odmeny, a po zvládnutí tém istej oblasti učiť ostatných hráčov.

3.2 Svet

Herný priestor predstavuje svet s voľným pohybom hráčov, rozdelený na oblasti. Priestor poskytuje dva stupne voľnosti (pohyb v smere osi X, Y).

Herný svet pozostáva z:

- Viditeľného herného sveta (fakty)
 - Statických objektov
 - Dynamických objektov
 - Hráčov
 - Interakcií
 - Hodnotenia
- Neviditeľného herného svetla (súvislosti)
 - fyzikálne zákony (globálne)
 - sociálne zákony (zónovo závislé)

3.2.1 Viditeľný herný svet

Statické herné objekty

Sú objekty, ktoré buď pri kontakte s hráčom nevyvolávajú špeciálne udalosti. Môžu ale zabráňovať hráčovi v pohybe.

- Statické objekty miest
 - Budovy a príslušenstvo ku budovám (Nepriechodné)
 - Mestská zeleň (Niektoré priechodné)
 - Ulice (priechodné)
 - Osvetlenie (nepriechodné, špeciálne)
 - Hradby resp. Iné oddeľovače (nepriechodné)
- Statické objekty prírody
 - Zeleň
 - Vodné plochy
 - Prirodzené prekážky (rieky, pohoria)

- o Umelé prekážky(postavené človekom)
- Statické objekty lokálnych svetov
 - o Miestnosti
 - o Zariadenie a príslušenstvo

Dynamické herné objekty

Sú objekty, ktoré po použití vyvolajú nejakú akciu, prípadne s ktorými sa dá manipulovať

- Predmety
- herné objekty špecifické pre konkrétnu úlohu (quest)
- Dvere
- Prepínače a páky, vypínače
- Spojnice
 - o Medzi svetmi (menia svet)
 - Vchody do dungeonov, východy z dungeonov (brány, schody)
 - Dopravné prostriedky
 - Portály (jednosmerné, viacsmerne)
 - o V rámci svetú (menia súradnice v svete)
 - Teleporty (viacadresové spojnice)
 - Portály

3.2.2 Neviditeľný herný svet

Svet má svoje zákonitosti, ktoré nie sú na prvý pohľad zrejmé. Zákonitosti, ktoré platia v jednotlivých častiach sú reprezentované ako neviditeľné vrstvy nad mapou krajiny, alebo ako globálne zákonitosti, ktoré nie sú lokalizované.

Fyzikálne zákony

Fyzikálne zákony sú v hernom svete nastavené globálne ale môžu existovať lokálne anomálie.

- Vrstva rýchlosti pohybu
- Optické vrstvy (Zelený svet, Modrá krajina...)

Sociálne zákony

Sociálne zákony sú lokálne závislé. Sú silne závislé od zón, v ktorých sa dieťa nachádza. Sú reprezentované vrstvami.

- Vrstva bezpečnosti oblasti (Miera intenzity výskytu príšer v oblasti)
- Vrstvy úrovne znalosti

3.2.3 Účastníci

Účastníkmi hry sú hráči (učiace sa deti), mentori, NPC postavy, zvieratka osvojené hráčmi a sprevádzajúce ich vo svete (pet) a tímy tvorené viacerými hráčmi.

Nasleduje zoznam všetkých entít, ktoré majú špeciálny charakter v systéme a ich význam spočíva v interakcii:

- NPC
- Hráč
- Mentor
- Pet
- Tím (Tvorený mentorom, hráčmi a ich pet)

Hráči

Hlavným typom účastníka je hráč – učiace sa dieťa. Každý hráč má svoj profil, v ktorom sú zaznamenané nasledovné informácie

- vek
- pohlavie
- rodný jazyk
- príslušnosť k skupine
- ďalšie informácie
 - o žiacka knižka

Hráč je vo svete reprezentovaný svojim avatarom vo vizuálnej forme. Má možnosť pohybovať sa po svete a interagovať s ostatnými objektmi v hre – s hráčmi a NPC postavami.

Používateľské skupiny

Aplikácia poskytuje možnosť definovať používateľské skupiny. Do skupiny patrí množina hráčov. Skupina môže mať definovaných mentorov, ktorí majú právo dohliadať nad aktivitami hráčov patriacich do danej skupiny.

Mentori

Špeciálny typ účastníkov sú mentori, ktorí majú v hre vymedzenú sadu právomocí. Mentor má právo zasahovať do procesu učenia sa detí, ktoré patria do skupiny, ktorú môže riadiť.

NPC (Non-player character)

NPC postavy v hre predstavujú softvérové agenty, reagujúce pri oslovení iným účastníkom hry (typicky hráčom). NPC zadávajú hráčom questy a pomáhajú im pri nasledovaní línií učenia sa. Pri nasledovaní línie učenia sa má žiak pridelenú špeciálnu NPC (tútora), ktorý ho sprevádza svetom. Tieto NPC budú tematicky ladené (Harry Potter, anime postavička...) a dieťa si bude môcť vybrať.

Ďalším typom NPC je zvieratko (pet), ktoré chodí všade s hráčom a môže mu pomáhať riešiť niektoré úlohy.

Tímy

Dôležitým rozmerom hry je podpora vytvárania tímov. Tímy sa vytvárajú niekoľkými spôsobmi:

- spoluhráči sa nájdu sami
- hra vyberie spoluhráčov do tímu
 - o snaha odporučiť vyvážený tím
- s ovplyvnením mentora

Tím figuruje pri plnení úloh ako jedna entita a získané body (skill pointy) sa započítavajú jednotlivým členom rovným dielom. Vytvorený tím môže prejsť do režimu „jeden učí ostatných“. Toto sa môže uskutočňovať v špeciálnej lokácii v hre – „team nest“.

Zvieratko (pet)

Ústrednou postavou hry, ktoré bude hráča sprevádzať počas celej doby trvania je zvieratko (pet). Po splnení prvej úlohy si bude môcť hráč vybrať z viacerých druhov, predpokladajú sa minimálne dva, jeden pre chlapcov a druhý pre dievčatá.

Zvieratko bude mať podobne ako hráč určité vedomosti, ktoré bude môcť počas hry rozvíjať tak, že ho bude hráč učiť. Zároveň mu bude môcť hráč kupovať rôzne predmety, ktoré budú mať vplyv na činnosti, ktoré bude vedieť vykonávať. Okrem znalostí bude mať hráčov spoločník aj svoj charakter a momentálny stav kvantifikovaný náladou, pričom od neho sa bude rozvíjať napríklad jeho ochota poradiť, pomôcť alebo inak interagovať.

Hráč bude môcť mať svoje zvieratko neustále pri sebe počas:

- Hry: kde bude figurovať ako samostatná postava sprevádzajúca hráčovho avatara.
- Súboja/Výučby: kde bude pôsobiť ako poradca, alebo ako “nezbedný lapaj”
- Operačný systém, plocha (Windows Vista): pomocou technológie gadgets bude môcť byť zvieratko zobrazené pri bežnej práci s počítačom, okrem základnej interakcie bude figurovať ako pomyselný most medzi herným a reálnym svetom pričom bude schopné spustiť hru (preniesť dieťa do iného sveta), prijať správy od iných hráčov v hernom svete a sledovať a zapisovať hráčove úspechy (žiacka knižka). Môžu naň byť napojené RSS feedy, prípadne môže byť prostredníkom komunikácie medzi hráčmi v hre a tými, ktorí sú mimo hry, ale prítomnými pri počítači.
- Mobilné zariadenie: Hráč si môže svoje zvieratko vystupujúce v hre nahráť do PDA a tam ho učiť (hraním sa špecifických hier) a starať sa oň (koncept tamagotchi). Keď sa príde znova hrať pri počítači, zvieratko si nahrá do online hry a tam mu robí spoločnosť a pomáha mu.

Nasleduje návrh predmetov, ktoré môže zvieratko používať:

- Hudobný nástroj (gitara, flauta...) – môže na požiadanie alebo keď má dobrú náladu zahrať – kvalita by bola podľa úrovne hudobnej znalosti
- Knižky – zvieratko môže čítať rozprávky
- Kalkulačka – môže počítat príklady
- Teplomer – vie povedať, aké je počasie mimo hry, pomocou externej webovej služby alebo vo vnútri hry(pomocou vrstvy počasia).

3.2.4 Interakcie

Zoznam interakcií medzi objektmi, ktoré sú schopné interagovať (Hráči, Dynamické herné objekty)

- Interakcia Hráč, Hráč
 - Duel
 - Komunikácia pomocou kanálu (Bez obmedzení)
 - Hry na jednom klientovi
 - Hry po sieti
- Interakcia Hráč, NPC
 - Duel
 - Zadávanie úloh
 - Hry
 - Zmena pozície hráča (Presun v deji, teleportácia)
 - Nákup, predaj
- Interakcia Hráč, Dynamický objekt
 - Závislé od typu predmetu
- Interakcia Hráč, Mentor
 - Výukový duel
 - Komunikácia pomocou kanálu
 - Hry
 - Zmena pozície hráča

- Interakcia Hráč, pet
 - o Výukové hry
 - o Zábavné hry
 - o Trénovanie
 - o Diaľkové ovládanie pet (Podľa poslušnosti)
- Interakcia pet, pet
 - o Spoločná hra
 - o Spoločný duel s pomocou hráča (alebo bez pomoci – Počas turnaja)
- Interakcia Tím, Tím
 - o Spoločná tímová hra
 - o Tímový duel (Rôzne druhy)
 - o Duel petov
 - o Zlúčenie tímov
 - o Rozloženie tímu
 - o Vytvorenie spoločného kanála
- Interakcia Tím, Hráč
 - o Prijatie hráča do tímu
 - o Vytvorenie tímu
 - o Vytvorenie komunikačného kanálu
- Interakcia Tím, NPC
 - o Nie je definovaná
- Interakcia Tím, pet
 - o Adoptovanie zvieratka Tímom
- Interakcia Tím, Mentor
 - o Pripojenie ku tímu
 - o Vedenie tímu
 - o Zadávanie úloh v tíme
 - o Opustenie tímu (Dočasný mentor)
 - o Rozpustenie tímu (Nevhodný, neslušný tím)
 - o Tvorba tímu mentorom

3.2.5 Hodnotenie

Úsilie hráčov je ohodnocované rôznymi spôsobmi. Plnením úloh zvyšujú úroveň svojich skillov. Tiež môžu dostať odmenu vo forme hernej meny, ktorú majú možnosť v hre vymieňať za veci. Spojením hodnotenia v reálnom svete a v hernom svete je koncept žiackej knižky, ponúkajúci možnosť integrovať hodnotenie v škole s hrou.

Skilly (Znalosti)

Skilly reprezentujú jednotlivé oblasti učenia sa (viď ďalej). Každý hráč má pre oblasti, v ktorých sa zúčastnil výučby alebo overovania znalostí, asociovanú hodnotu reprezentujúcu jeho úroveň v danej oblasti (skill pointy). Pri dosiahnutí istého počtu skill pointov a vyznamenaní (viď ďalej) dostáva hráč právo v danej oblasti vyučovať iných hráčov.

Herná mena

Herná mena vo svete predstavuje výmenný prostriedok, pomocou ktorého hráč môže prispôbiť niektoré vlastnosti sveta podľa svojich želaní.

Žiacka knižka

Koncept žiackej knižky dáva možnosť napojiť svet hry na proces učenia sa v škole. Učitelia by do elektronickej žiackej knižky zapisovali známky deťom a pri vedomostných oblastiach, ktoré sa učia v hre, by sa zohľadňovali školské výsledky dieťaťa. Počas procesu trénovania jednotlivých znalostí môže byť prihliadané na dosiahnutú úroveň znalostí deklarovanú v žiackej knižke. Tiež môžu byť prehľbované znalosti v oblastiach, v ktorých dieťa v škole zaostáva.

Rozšírením konceptu žiackej knižky sú domáce úlohy zadávané učiteľom na hodinách, ktoré riešia deti doma s pomocou aplikácie Pelikán.

Ďalším možným rozšírením je integrácia aplikácie priamo do vyučovacieho procesu.

3.2.6 Logovanie a štatistiky

Aplikácia by mala ukladať čo najväčšie množstvo informácií o priebehu hry. Informácie je možné ďalej spracovať a poskytnúť ich štatistické vyhodnotenie za účelom prispôsobenia výučbového procesu, zhodnotenia úspechov hráča, porovnania s úspechmi ostatných

Budú zaznamenané najmä nasledovné informácie:

- dĺžka pobytu hráča v jednotlivých herných zónach
 - dĺžka učenia sa pre jednotlivé vedomostné oblasti
- rýchlosť postupu pri učení sa
- dosiahnutá úroveň vedomostí pri overovaní vedomostí v jednotlivých hrách
- konkrétne odpovede na otázky v hrách formou otázky

Vyhodnotením zaznamenaných informácií budú k dispozícii nasledovné informácie

- globálna úroveň dosiahnutých znalostí (percento správnych odpovedí/úspešných pokusov)
- relatívna úroveň dosiahnutých znalostí (percentil v rámci herného sveta)
- vyhodnotenie úrovne znalostí relatívne pre jednotlivé vedomostné oblasti v hre

3.3 Komunikácia

Základným rámcom komunikácie sú kanály. Hráč pripojený na kanál (účastník kanálu) prijíma všetky správy vyslané ostatnými účastníkmi kanálu.

Existujú nasledovné formy realizácie komunikačných kanálov:

- textová
- hlasová
- video (webcam)

Z hľadiska hernej logiky môžu byť došpecifikované ľubovoľné kanály, ako príklad uvádzame nasledovné kanály:

- help – ak si niekto nevie rady
- „hľadá sa partia“
- „hľadá sa učiteľ“
- Globálny kanál

3.4 Hry

Aplikácia rozlišuje medzi režimom učenia sa a módom preverovania znalostí. V móde učenia sa sú aplikované adaptívne prístupy za účelom dosiahnutia čo najvyššieho stupňa osvojenia si znalostí. Herný svet obsahuje oblasti určené na učenie (realizácia línií učenia sa) a „nebezpečné“ oblasti s voľným pohybom, kde môže byť hráč počas pohybu skúšaný.

3.5 Questy (Úlohy)

Questy v hre predstavujú línie nadobúdania znalostí. Hráč počas hry napreduje naplňaním questov.

Príklad: musí úspešne vykonať quest overujúci mieru osvojenia si sčítania aby sa mohlo ísť učiť násobilku.

Existujú dva druhy questov:

V závislosti od procesu výuky

- línia učenia sa (učenie)
- preverovacie questy (skúšanie)

V závislosti od počtu avatarov potrebných na splnenie úlohy

- individuálne
- tímové

3.5.1 Línia učenia sa

Línia učenia sa predstavuje definovanú postupnosť úloh, ktoré by mal hráč splniť za účelom nadobudnutia znalostí v danej oblasti. Počas línie učenia sa hráča sprevádza NPC alebo iná postava (mentor, hráč), ktorá je jeho tútorom.

Vyznamenania

Za splnenie nejakej línie učenia sa dostáva hráč vyznamenanie, zohľadňujúce mieru osvojenia si preverovaných znalostí (zlatá, strieborná, bronzová medaila). Za splnenie každej úlohy v línii učenia sa dostáva hráč skúsenostné body v danej oblasti.

Ovplyvnenie výberu línie učenia sa

- systém autonómne určuje
 - o definícia profilu pre hráča – vyvážený profil, profil do hĺbky v jednej oblasti, možnosť zakázať jednotlivé oblasti
- definovanie línie učenia sa mentorom
- hráč sa rozhoduje samo výberom questov od NPC účastníkov

V každom momente počas učenia sa systém sleduje výkonnosť hráča, a ponúka zmenu aktivity, zníži obtiažnosť, prejde do módu vysvetľovania alebo vykoná inú vhodnú akciu, ak usúdi, že je to v danom momente pre zlepšenie progresu v procese učenia sa.

3.5.2 Preverovacie questy

Preverovacie questy predstavujú pre deti možnosť uplatniť nadobudnuté znalosti za účelom získania bodov. Preverovacie questy zadávajú NPC účastníci hry. Rozlišujú sa tímové a individuálne questy.

Quest je reprezentovaný postupnosťou úloh, ktoré je potrebné vykonať, aby tím splnil danú úlohu. Jednotlivé úlohy môžu byť z rôznych alebo z tej istej oblasti. Pri výbere questu má hráč k dispozícii informáciu o tom, z ktorej oblasti daný quest je.

3.6 Scenáre použitia

Táto kapitola obsahuje opis priebehu hry z pohľadu jednotlivých druhov účastníkov. Prvá podkapitola popisuje priebeh hry z pohľadu hráča - učiaceho sa dieťaťa a druhá z pohľadu mentora – rodiča alebo učiteľa.

3.6.1 Hráč

Odvážny prieskumník

Dieťa si prvýkrát zapína hru. Sprievodca ho prevedie úvodnou obrazovkou kde ho počas vytvárania avatara naučí princípom hlasového ovládania hry. Po vytvorení avatara sa ocitá v hlavnom meste. Tu je pri ňom jeho prvý tútor, ktorý mu vysvetlí, aké má v hernom svete možnosti. Akonáhle sa dieťa cíti dostatočne poučené, vyraža skúmať svet okolo. Stretáva NPC postavy, ponúkajúce mu v rámci príbehov, že ho naučia vedomosti z rôznych oblastí. Dieťa si vyberie jedného z tútorov a ten ho prevedie jednotlivými hrami a dieťa tak získava skúsenosti a ocenenia v rámci hry. Počas procesu učenia sa systém snaží rôznymi prostriedkami adaptovať individuálnemu tempu učenia sa dieťaťa výberom obtiažnosti úloh ako aj výberom samotných tém úloh. Línie učenia sa obsahujú niekoľko kontrolných bodov – skúšok. Ak dieťa preukáže dostatočnú úroveň znalostí na skúške, pokračuje k ďalším úrovňam kým nezloží záverečnú skúšku. Po završení prvej línie učenia sa a dosiahnutím najvyššieho statusu získava ako odmenu zvieratko. Toto mu robí spoločníka a môže mu pomáhať pri zvládaní úloh v hre. Zároveň sa zvieratko stáva vstupným bodom do herného sveta, pretože len ono má možnosť kedykoľvek vytvárať spojenie so svetom.

Po nadobudnutí skúseností sa dieťa spolu so svojim spoločníkom púšťa prebádať krajinu. Stretáva rôzne „príšerky“, ktoré mu dávajú úlohy a za ich zvládanie dostáva body/peniaze, ktoré môže vymieňať za predmety v hre.

Skúsené dieťa sa zúčastňuje súťaží v aréne a snaží sa dosiahnuť čo najvyššie skóre v rámci herného sveta. Hra vyhodnocuje štatistiku jeho úspešnosti.

„Ve dvou se to lépe táhne“

Dvaja alebo viacerí skúsení hráči, každý dobrý v inej oblasti, sa spoja a vytvoria tím (alianciu, partiu). Spoločne sa môžu zúčastniť hier v aréne, alebo prebádať svet a spoločnými silami zdolávať protivníkov, na ktorých by ako jednotlivci nemali šancu. Systém sa snaží motivovať hráčov, aby tvorili vyvážené tímy, alebo aby ich v prípade že tím nie je vyvážený (každý je veľmi dobrý v inej oblasti) motivoval k tomu, aby si navzájom pomohli (vzájomné učenie) dosiahnuť vyššiu úroveň vo všetkých oblastiach. Toto je zabezpečené penalizáciou (znížením odmeny), prirodzenou motiváciou hráčov byť lepší a poskytnutím prostriedkov, ktoré im umožnia vzájomne zvyšovať úroveň znalostí.

Školská aktivita

Deti dostali okrem bežnej domácej úlohy za úlohu získať v hre isté množstvo bodov. Mentor definoval v hre quest, prípadne vypísal turnaj s účasťou obmedzenou na členov jeho triedy. Deti majú možnosť pripraviť sa na úlohu na cvičisku prípadne absolvovaním celej línie

učenia sa. Po absolvovaní úloh sú vyhodnotené a mentor má k dispozícii výsledky ich snaženia.

3.6.2 Mentor

Mentor (učiteľ/rodič) po prihlásení do hry získava prehľad o štatistikách úspešnosti svojich zverencov. Systém odhadne prípady, ktoré dá mentorovi do pozornosti

- skokan dňa – ten, kto dosiahol najväčší progres
- víťaz najväčšieho počtu turnajov
- hráč, ktorý má problémy so zvládaním istých úloh

Mentor môže prispôbovať línie učenia sa jednotlivcov alebo celej skupiny hráčov. Tiež môže modifikovať úroveň otázok kladených jednotlivcom alebo skupine.

Ďalšou možnosťou je vypisovanie turnajov. Mentor v aréne na to určenej vypíše turnaj a určí ceny. Turnaj môže byť uzavretý pre určitú skupinu hráčov (napríklad detí z triedy) a deti môžu dostať účasť na turnaji napríklad ako školskú aktivitu (prípadne domácu úlohu).

3.7 Používateľské rozhranie

Používateľské rozhranie smerom k používateľovi je grafické a zvukové. Ovládanie hry je zabezpečené myšou, klávesnicou, gamepadom, hlasom, alebo rôznou kombináciou týchto metód.

Na ovládanie pohybu svojho avatara a interakciu s hernými objektmi používa hráč hlasové príkazy. Ku každému hlasovému príkazu existuje možnosť dosiahnuť danú akciu myšou. Konkrétne úlohy vo forme malých hier rieši hráč s použitím špecifického zariadenia vhodného pre interakciu pri danom type úlohy (hudobné najmä s mikrofónom, písanie s tabletom/snímaním pera na papieri)

Ako rozšírenie sa uvažuje s ovládaním určitých typov podhier vo forme rozoznávania obrazu.

4 Špecifikácia častí systému

4.1 Herný svet

Hlavná časť hry, pozostáva predovšetkým z herného enginu a komunikačného rámca. Pre hráča predstavuje virtuálny priestor, po ktorom sa môže pohybovať a stretávať s inými hráčmi. V tejto časti sa odohráva väčšina výuky.

4.2 Gadget

Časť, ktorá rezidentne beží v operačnom systéme a ktorá umožňuje hráčovi interagovať so svojim zvieratkom a s niektorými offline aspektmi hry ako žiacka knižka alebo správy od používateľov v hre.

4.3 PDA

Predstavuje ju aplikácia bežiaca na mobilnom zariadení umožňujúca kontakt so zvieratkom ako aj možnosť hrať s ním nejaké jednoduché hry a tým ho učiť. Konceptne je táto časť doplnková k hernému svetu v čase, keď sa dieťa nenachádza doma.

4.4 Nástroj pre rodičov a učiteľov

Predstavuje systém, v ktorom je možné sledovať a kontrolovať proces výučby. Sledovanie je možné hlavne pomocou určitých štatistických veličín napríklad počet správnych odpovedí alebo náročnosť otázok. Proces výučby bude možno modifikovať napríklad pridaním nových otázok a úloh k rôznym výukovým oblastiam.

5 Analýza technológií pre vývoj

Kapitola poskytuje prehľad a analýzu technológií, ktorých použitie bolo zvažované v rámci realizácie projektu. Medzi základné kritéria patrila najmä jednoduchosť použitia, podpora rámca .NET a vysoká miera abstrakcie bez nutnosti ovládať architektúru nízkej úrovne. Analýza zahŕňa všetky aspekty potrebné pre realizáciu projektu, od grafickej reprezentácie, cez sieťovú komunikáciu až po prehrávanie a zaznamenávanie zvuku.

5.1 Grafická reprezentácia

Rámec .NET Framework poskytuje základnú grafickú funkcionálnosť prostredníctvom knižnice operačného systému gdi32.dll, ktorá štandardne zodpovedá za kreslenie grafického rozhrania, ovládacích prvkov, formulárov a iných grafických prvkov v prostredí Microsoft Windows. Neposkytuje však podporu pre akcelerovanú grafiku, ktorú umožňujú moderné grafické karty a použitie tejto knižnice je teda vhodné iba pre základné grafické operácie. Jednoduchá grafická reprezentácia je pomocou nej realizovateľná, no rýchlosť je pre potreby projektu nedostatočná.

5.1.1 OpenGL

Knižníc poskytujúcich funkcie grafických akceleratorov je veľké množstvo, väčšina však stojí na sade knižníc OpenGL alebo Microsoft DirectX. Existuje iba malý počet knižníc, ktoré poskytujú funkcie OpenGL a stavajú na .NET Framework, okrem iných napríklad knižnice Tao alebo CsGL, mnohé z nich majú momentálne na neurčito pozastavený vývoj.

5.1.2 DirectX

Rámec DirectX je široko podporovaný a používaný na vývoj akcelerovaných herných grafických prostredí a má plnú podporu rámca .NET v podobe sady knižníc Managed DirectX, ktoré sa pravidelne aktualizujú, vrátane veľkého množstva dostupných príkladov a aktívnej komunity. V minulosti sa grafická podpora členila na dve časti – knižnica podporujúca 2D funkcionálnosť zvaná DirectDraw a 3D s názvom Direct3D, od verzie DirectX 9 sú tieto knižnice zlúčené a Microsoft naďalej podporuje už iba Direct3D, pomocou ktorého sa kreslí aj 2D grafika. Vzhľadom na zložitosť modelu Direct3D je použitie na realizáciu kreslenia jednoduchej 2D grafiky nevhodné. Niekoľko konceptov spojených s 3D grafikou je abstrahovaných v pomocnej knižnici Direct3DX, ktorá poskytuje vyššiu úroveň abstrakcie.

5.1.3 WPF (Windows Presentation Foundation)

V súčasnosti sa Microsoft pokúša zakryť komplexnosť návrhom zrozumiteľnejšieho objektového modelu stavajúcim na DirectX, ktorého použitie by bolo na realizáciu grafických operácií jednoduchšie. Preto vznikol ako súčasť rámca .NET Framework 3.0 nový set knižníc nazvaný Windows Presentation Foundation (WPF, známa tiež pod kódovým označením „Avalon“), ktorý používa služby DirectX a poskytuje zrozumiteľnejší model na tvorbu bohatých grafických používateľských rozhraní. Model bol navrhnutý veľmi všeobecne s vysokou úrovňou abstrakcie a je preto vhodný na ľubovoľnú úlohu spojenú so znázorňovaním grafiky, či už 2D rastrovej alebo vektorovej, prípadne dáva možnosť vytvoriť plne trojdimenzionálne rozhrania. Napriek zrozumiteľnosti je však objektový model pomerne rozsiahly a nie je priamo stavaný na vývoj grafických herných prostredí.

5.1.4 Microsoft XNA

V rámci novej iniciatívy Microsoft vyvinul sadu knižníc s názvom XNA, ktorá má slúžiť ako rámec na rýchly vývoj grafických herných prostredí. Okrem rámca XNA Framework Microsoft poskytuje aj integrované prostredie na vývoj herných aplikácií, ktoré je k dispozícii hneď v niekoľkých verziách, v závislosti od potrieb používateľa. Základným prostredím je XNA Game Express Studio, ktoré okrem samotného XNA rámca obsahuje aj niekoľko predpripravených herných šablón, ktoré demonštrujú použitie novej technológie a ďalej zjednodušujú vývoj herného prostredia. Ďalšou výhodou rámca XNA je platformová nezávislosť, vďaka ktorej je možné vytvorené aplikácie spustiť okrem operačného systému Windows aj na hernej konzole Xbox 360. Vzhľadom na to, že je rámec XNA priamo určený na vývoj herných prostredí, poskytuje vysokú mieru abstrakcie a je k nemu k dispozícii integrované vývojové prostredie, ktoré umožňuje vytvárať platformovo nezávislé aplikácie, je preto vhodný na realizáciu riešeného projektu.

5.2 Sieťová komunikácia

Analýza komunikačných technológií má za úlohu nájsť a určiť takú, ktorá spĺňa požiadavky na rozsah riešeného projektu. Technológia by mala poskytovať komunikáciu na vyššej úrovni abstrakcie, spoľahlivé doručovanie a dostatočný výkon a škálovateľnosť. Rámec .NET poskytuje niekoľko komunikačných variantov od najnižšej komunikačnej úrovne na báze komunikačných socketov po najvyššiu na úrovni webových služieb.

5.2.1 Sockety

Základný spôsob komunikácie obsiahnutý v .NET rámci je komunikácia pomocou socketov. Možno je použitie UDP ako aj TCP socketov, ich použitie však vyžaduje riešenie mnohých súvisiacich problémov ako je asynchrónne spracovanie prichádzajúcich dát, návrh vlastného komunikačného protokolu pokrývajúceho potreby projektu na nízkej úrovni a mnohé ďalšie. Poskytujú tiež najvyššiu výkonnosť.

Ďalšou možnosťou je použitie objektov s vyššou mierou abstrakcie, ktoré zaoberajú funkcionality socketov, poskytujú rovnakú výkonnosť, no problémy sú totožné ako pri komunikácii na úrovni socketov.

5.2.2 .NET remoting

Navyššiu úroveň poskytuje .NET remoting a webové služby. Pomocou nich je možné vytvárať komunikačné kanály a komunikovať na úrovni objektov volaním vzdialených metód namiesto dátových balíkov posielaných na nízkej úrovni. Výhodou je vysoká miera abstrakcie za cenu nízkeho výkonu, zníženého kvôli réžii komunikačného protokolu a veľkosti posielaných správ. Tieto metódy zároveň vystupujú ako najjednoduchšie riešenie.

5.2.3 WCF (Windows Communication Foundation)

S príchodom .NET rámca verzie 3.0 pribudla nová sada komunikačných knižníc s názvom Windows Communication Foundation (WCF, pôvodne s kódovým označením „Indigo“), ktorá prináša služby ešte vyššej úrovne ako poskytuje staršia verzia .NET rámca. WCF je navrhnutá ako architektúra orientovaná na služby a poskytuje možnosť definovania komunikačných rozhraní na princípe tzv. kontraktov. Kontrakty definujú metódy, ktoré budú prístupné na koncových bodoch služby, pričom každý koncový bod môže poskytovať inú množinu metód. Komunikácia medzi klientom a serverom môže byť duplexná, pričom si obe strany navzájom vymieňajú SOAP správy, ktoré sú, na rozdiel od bežných webových služieb,

v binárnom formáte, ak spolu komunikujú dve WCF služby. Táto komunikačná metóda poskytuje najvyššiu úroveň abstrakcie a pri realizácii komunikácie pomocou TCP kanálov poskytuje aj vysokú výkonnosť aj napriek režii SOAP protokolu. Vďaka svojim vlastnostiam sa pre realizáciu komunikácie v rámci projektu vybrala táto komunikačná metóda.

5.3 Manipulácia so zvukom

Pre potreby implementácie hudobnej domény je potrebné použitie technológie umožňujúcej manipuláciu so zvukom, okrem iného napríklad záznam pomocou mikrofónu alebo aplikovanie zvukových efektov.

5.3.1 Wavein/waveout API

Nahrávanie zvuku umožňuje niekoľko štandardných knižníc, ktoré poskytuje operačný systém Microsoft Windows. Medzi základné patrí wavein/waveout API poskytnuté knižnicou winmm.dll. Táto knižnica však poskytuje nízkoúrovňové rozhranie, malú mieru abstrakcie a nemá podporu v .NET Framework 2.0.

5.3.2 DirectSound

Alternatívou k nízkoúrovňovým knižniciam operačného systému je súbor multimediálnych knižníc DirectX, ktorý obsahuje knižnicu DirectSound. Táto má priamu podporu .NET Framework v podobe Managed DirectX a poskytuje požadovanú funkcionálnosť a zároveň vyššiu mieru abstrakcie ako systémové knižnice.

5.3.3 XACT

Okrem DirectSound v súčasnosti existuje aj nová technológia zvaná XACT, ktorá je súčasťou DirectX SDK a dodáva sa tiež k integrovanému vývojovému prostrediu XNA Game Studio Express. Táto technológia poskytuje vyššiu úroveň abstrakcie ako DirectSound, umožňuje priamo tvorbu zvukov a ich organizovanie do väčších celkov, ktoré môžu okrem iného obsahovať inštrukcie ako jednotlivé zvuky prehrávať.

Pre realizáciu projektu je vhodná kombinácia technológií DirectSound a XACT, vďaka vlastnostiam a funkcionalite ktorú poskytujú, pričom sa navzájom dopĺňajú.

6 Hrubý návrh

Kapitola hrubý návrh obsahuje popis hrubého návrhu systému. Prvá podkapitola uvádza návrhové rozhodnutia, ktoré sme vykonali v procese jeho navrhovania a druhá časť popisuje architektúru navrhnutého systému.

6.1 Návrhové rozhodnutia

Podkapitola obsahuje návrhové rozhodnutia, ktoré sme zvážili pri navrhovaní systému. Prvá jej logická časť uvádza atribúty návrhu a preferenciu jednotlivých atribútov v prípade navrhovaného systému s ohľadom na požiadavky na systém. Druhá podkapitola uvádza rozhodnutia o modeli komunikácie v rámci systému a tretia o štruktúrovaní systému.

6.1.1 Atribúty návrhu

Pre aplikácie založené na sieťovej komunikácii je dôležité navrhnuť architektúru systému a model komunikácie s ohľadom na nasledovné vlastnosti:

- škálovateľnosť
- výkonnosť
- rozšíriteľnosť
- elegantnosť návrhu
- bezpečnosť
- jednoduchosť

Vzhľadom na fakt, že sa v projekte prikladáme dôraz na rýchle prototypovanie a zároveň je našim cieľom dosiahnuť a zdokumentovať vysokú kvalitu výsledku, deklarujeme požiadavky na vlastnosti systému v nasledujúcom poradí:

1. jednoduchosť
2. rozšíriteľnosť
3. elegantnosť návrhu
4. výkonnosť
5. škálovateľnosť
6. bezpečnosť

Keďže je našim cieľom navrhnuť funkčný prototyp, je prvoradou hodnotou navrhnuť systém tak, aby sme ho zachovali jednoduchým a ľahko rozšíriteľným. Vďaka týmto dvom vlastnostiam budeme môcť navrhovaný systém počas nasledujúcich fáz vývoja rozširovať o dodatočnú funkcionality.

S ohľadom na požiadavky a hodnotiace kritériá súťaže Imagine Cup a tiež všeobecné požiadavky na kvalitu softvérových systémov je vhodné systém navrhnuť tak, aby preukazoval vysokú elegantnosť návrhu.

Výkonnosť je v prototype systému vyžadovaná len do tej miery, aby systém preukazoval dostatočnú mieru odozvy na to, aby bola zachovaná požiadavka dojmu plynulej odozvy.

Bezpečnosť sa neocitla na poslednom mieste z dôvodu, že by sme ju chceli podceňiť, ale preto, že uvažované použité komunikačné a implementačné technológie poskytujú vstavaný deklaratívny model bezpečnosti. Bezpečnosť komunikácie je zaistená použitím technológie WCF, ktorá poskytuje vstavanú bezpečnosť na úrovni správ a/alebo transportných kanálov a autentizácie.

6.1.2 Model komunikácie

Rozlišujú sa dva základné modely komunikácie – push a pull model. V prípade push modelu server iniciuje volania smerom ku klientovi a notifikuje ho o udalostiach, zatiaľ čo pull model predpokladá, že komunikácia prebieha výlučne na žiadosť klientov.

Push model podporuje jednoduchší návrh a prináša vyššiu elegantnosť, vzhľadom na fakt, že umožňuje iniciovanie komunikácie ako zo strany klienta, tak aj zo strany servera smerom ku klientom. Spomínaná vlastnosť zároveň vyžaduje existenciu duplexných komunikačných kanálov.

Abstrakcia na úrovni WCF, ktoré sme si vybrali pre realizáciu komunikačnej vrstvy systému, umožňuje volanie vzdialených metód. WCF tiež ponúka duplexný model vzdialeného volania metód, podmienený skutočnosťou, že je *ServiceHost* nakonfigurovaný tak, že používa duplexný *Binding*.

Zo vstavaných poskytujú duplexnú komunikáciu *NetTcpBinding*, *WSDualHttpBinding*, *NetPeerTcpBinding* a *NetNamedPipeBinding*. Vzhľadom na fakt, že základný model komunikácie v danom type aplikácií je klient-server a klient a server sa zo zásady nachádzajú na rozdielnych fyzických počítačoch, nie je možné použiť *NetPeerTcpBinding* ani *NetNamedPipeBinding*, ktoré podporujú *peer-to-peer sessionless* model komunikácie, prípadne vyžadujú aby bežali komunikujúce aplikácie na tom istom počítači.

Obe zvyšné možnosti poskytujú deklaratívny model bezpečnosti či už na úrovni správ alebo transportných kanálov.

Vzhľadom na vyššiu mieru interoperability sme sa rozhodli použiť *WsDualHttpBinding* s tým, že v prípade zvýšených nárokov na výkonnosť ho je možné jednoducho zameniť za *NetTcpBinding*.

6.1.3 Štruktúra systému

Základná štruktúra systému daná jeho špecifikáciou predpokladá existenciu serverovej a klientskej časti systému. Z hľadiska štruktúry systému je možné rozlíšiť dva základné modely:

- monolitická serverová časť (čisté klient-server riešenie)
- distribuovaná serverová časť (architektúra orientovaná na služby)

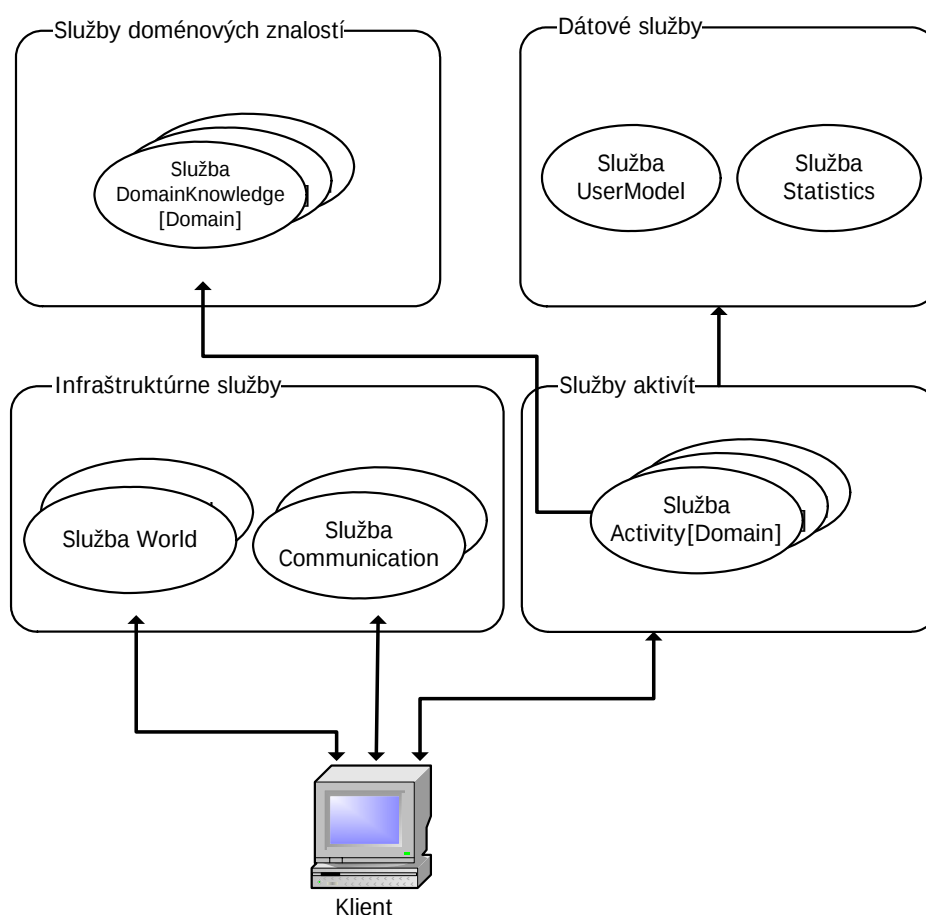
Monolitický model predpokladá existenciu jediného servera, ktorý poskytuje všetku serverovú funkcionality systému. Distribuovaný model naopak uvažuje so serverovou časťou systému ako s množinou nezávislých služieb. Práve vďaka jednoduchosti predstavuje monolitický model vhodnejšiu architektúru, keďže odpadá dodatočná komplexnosť spojená s koordináciou služieb a definovaním dodatočných rozhraní nutných pre zdieľanie dát.

Na druhej strane vykazuje distribuovaný model nepomerne silnejšiu mieru kohézie a prináša jednoduchšiu rozšíriteľnosť a vyššiu elegantnosť návrhu.

6.2 Architektúra systému

Podkapitola obsahuje popis architektúry systému. Diagram na obrázku č. 6-1 vyobrazuje prehľad systému na úrovni jednotlivých služieb. Služby sú logicky rozdelené do niekoľkých častí:

- infraštruktúrne služby
 - poskytujú základný model sveta a rámec pre komunikáciu hráčov
- služby aktivít
 - modelujú jednotlivé doménovo špecifické aktivity (napríklad konkrétne hry alebo turnaje)
- dátové služby
 - slúžia ako úložný priestor statických dát , štatistík, prípadne na správu dynamických dát (model používateľa)
- služby doménových znalostí
 - poskytujú úložný priestor pre doménovo špecifické znalosti, využívané doménovo špecifickými službami aktivít (napríklad matematické príklady)



Obrázok č. 6-1: Architektúra systému

6.2.1 Infraštruktúrne služby

Infraštruktúrne služby predstavujú jadro systému. Klient komunikuje výhradne s týmito službami. Komunikácia s týmito službami prebieha cez duplexné kanály.

Služba World

Ústrednou časťou systému je služba *World*, ktorá poskytuje pre klientov model objektívnej reality. Poskytuje vzťahnú sústavu a v nej príslušné koordináty jednotlivých objektov. V systéme môže existovať väčšie množstvo inštancií služby *World*, modelujúcich rôzne fyzikálne zákony pre jednotlivé podsvety alebo alternatívne svety (dungeony, nebo, peklo).

Služba Communication

Služba *Communication* slúži na sprostredkovanie komunikácie medzi klientmi. Kontrakty služby *Communication* na strane klienta umožňujú aj *peer-to-peer* komunikáciu medzi jednotlivými klientmi, užitočnú najmä pri tých typoch kanálov, ktoré majú vysoké požiadavky na prenášaný objem dát. Bežné komunikačné kanály (textové) sú klientom poskytované sprostredkované cez službu *Communication*.

6.2.2 Služby aktivít

Služby aktivít predstavujú jednotlivé konkrétne doménovo špecifické aktivity. Príkladom takejto aktivity môže byť hra v rámci plnenia nejakého *questu* alebo účasť na turnaji v konkrétnej disciplíne. Služby aktivít spolupracujú so službami doménových znalostí a získavajú od nich inštancie príkladov na základe charakteristík používateľa získaných zo služby *UserModel*.

6.2.3 Dátové služby

Dátové služby slúžia primárne ako úložisko dát. Poskytujú typicky metódy na pridávanie, výber, zmenu a mazanie dát. Dáta zahŕňajú štatistické dáta a dáta o používateľoch (hráčoch) vo forme modelu používateľa.

Služba Statistics

Služba *Statistics* slúži na zaznamenávanie údajov o histórii a tiež vyhodnotenie týchto dát.

Služba UserModel

Služba *UserModel* slúži na správu modelov používateľa. S pomocou tejto služby si používateľ vytvára na začiatku základný profil. V spolupráci so službou *Statistics* služba *UserModel* vyhodnocuje jednotlivé charakteristiky používateľa a tieto sprostredkuje ostatným službám.

6.2.4 Služby doménových znalostí

Služby doménových znalostí predstavujú množinu služieb sprostredkujúcich znalosti z konkrétnych problémových domén, slúžiace ako výukové príklady do hier. Okrem poskytovania znalostí umožňujú tiež pridávanie a správu inštancií príkladov. Ich rozhranie je verejné a prístupné aj pre iné systémy.

7 Prototyp

Kapitola obsahuje výsledky tvorby prototypu počas zimného semestra. Tam, kde to bolo možné sme uprednostnili implementáciu tých častí, ktoré spolu tvoria súrodý celok, prípadne na seba svojou funkcionalitou nadväzujú.

Na strane servera sme tvorili prototyp rámca komunikácie, ktorého cieľom bolo overiť technológiu Windows Communication Foundation, ktorú plánujeme v projekte nasadiť. Súčasťou prototypu sú taktiež vybrané služby na strane servera, slúžiace ako realizácia herného sveta, hier na strane servera a služby poskytujúce doménové znalosti. Výsledky vytvárania prototypu servera zhŕňa samostatná podkapitola.(viď 7.1.1) Vo fáze tvorby prototypu sme sa sústredili vytvorenie evolučným prístupom rozšíriteľného prototypu, z ktorého budú vychádzať ďalšie iterácie aplikácie.

Súčasťou prototypu na strane klienta je engine, postavený na báze rámca Microsoft XNA. Účelom prototypu bolo overiť použiteľnosť novej technológie pre účely tvorby systému. Prototypu engine je rozširovateľný a plánuje sa v jeho vývoji pokračovať až do finálnej verzie produktu. V prototypu sme engine integrovali s komunikačným rámcom servera a vytvorili sme jednoduchú aplikáciu, ktorá využíva podmnožinu funkcionality servera. Prototypu engine a klientskej aplikácie je popísaný v kapitole herný klient.. (viď 7.1.2)

Ďalšou časťou prototypu je editor máp. Jeho vytvorenie bolo logickým vyústením požiadavky tvorby mapy za účely prezentácie prototypu. Tiež ide o evolučný prototyp. Editor máp je popísaný v kapitole Editor máp. (viď 7.1.3)

Poslednou implementovanou časťou je aplikácia pre mobilné zariadenie. Účelom prototypovania bolo overiť technológiu a možnosti mobilných zariadení. Popis časti systému obsahuje kapitola PDA klient. (viď 7.1.4)

7.1 Implementácia častí prototypu

Podkapitola popisuje implementované súčasti prototypu. Pre každú implementovanú časť zhŕňa dôležité implementačné rozhodnutia a účel samotnej tvorby prototypu.

7.1.1 Server

Na strane servera sme sa rozhodli vytvoriť prototyp modelu komunikácie, aby sme overili vhodnosť vybranej technológie WCF. Bolo potrebné overiť, či bude možné s pomocou technológie WCF prevádzkovať na serveri naraz viacero duplexných služieb naraz v jednom procese a prepínať medzi nimi na strane klienta na príkaz servera.

Komunikačný rámec

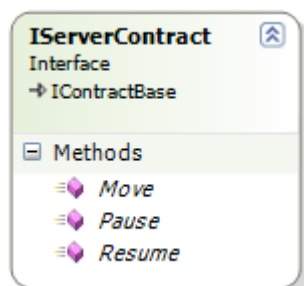
Prototyp komunikačného rámca predstavuje vymedzenie rozhraní pre komunikáciu jednotlivých súčastí systému.

Navrhnutý model komunikácie predpokladá duplexnú stavovú komunikáciu vzdialeným volaním metód. V prototypu komunikačného rámca bol realizovaný na vysokej úrovni abstrakcie modelovaním priameho volania vzdialených metód, poskytovanej technológiou WCF. technológia poskytuje možnosť definovania komunikačných rozhraní na princípe tzv. kontraktov. Prepojenia sú špecifikované tzv. *Bindingom*. Pre zabezpečenie duplexného

modelu komunikácie sme využili v komunikačnom rámci *WsDualHttpBinding*, ktorý je poskytovaný v knižnici WCF.

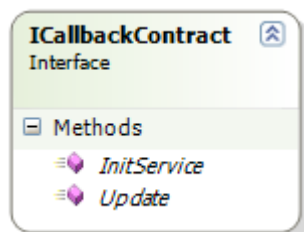
Služba World

Základná služba v systéme je služba World, ktorá predstavuje model virtuálneho priestoru sveta. Zároveň je hlavnou službou, ktorá riadi prístup klientov k ostatným službám. Kontrakt služby World predstavuje rozhranie *IServerContract* (viď obrázok č. 7-1) s metódami umožňujúcimi pohyb vo virtuálnom priestore.



Obrázok 7-1. Rozhranie *IServerContract*

Tzv. callback contract je realizovaný rozhraním *ICallbackContract*, ktoré poskytuje metódu pre notifikovanie klienta o zmene stavu sveta (*Update*) a príkaz pre klienta, aby oslovil inú službu (*InitService*).



Obrázok 7-2 Rozhranie *ICallbackContract*

Služba Communication

Služba Communication slúži na sprostredkovanie textovej komunikácie medzi pripojenými klientami. Pripojenie sa iniciuje zavolaním metódy *CreateToken*, ktorá vráti *Token*, za pomoci ktorého sa klient preukazuje pri posielaní správ. Môže si vyžiadať zoznam pripojených klientov zavolaním metódy *ListTokens*. Posielanie správ prebieha volaním metódy *SendMessage*. Kontrakt smerom ku klientovi (tzv. *callback contract*) reprezentuje rozhranie definované jedinou metódou *ReceiveMessage*, ktorú zavolá server v prípade, že niekto pošle správu s daným klientom uvedeným ako adresátom.

Rámec aktivít

Rámec aktivít definuje rozhrania a základné triedy pre vytváranie aktivít na serveri. Základom je trieda *GameBase* a *GameInstance*. Triedy odvodené z týchto tried predstavujú herné aktivity implementované na serveri.

Trieda *GameBase* riadi vytváranie inštancií hier a pridelovanie hráčov k jednotlivým inštanciám. Trieda *GameInstance* zahŕňa logiku prihlasovania hráčov k hre a spúšťania a ukončovania hry. Na toto slúžia metódy definované v triede *GameInstance*.

7.1.1.1.1 Implementované aktivity

Na overenie navrhnutého rámca aktivít sme implementovali niekoľko aktivít. Išlo o jednu interaktívnu aktivitu vo forme hry „Obesenec“ a jednu neinteraktívnu aktivitu vo forme hry, ktorá skúša hráča z matematických znalostí. Využíva pritom službu doménových znalostí z matematiky.

Doménové znalosti

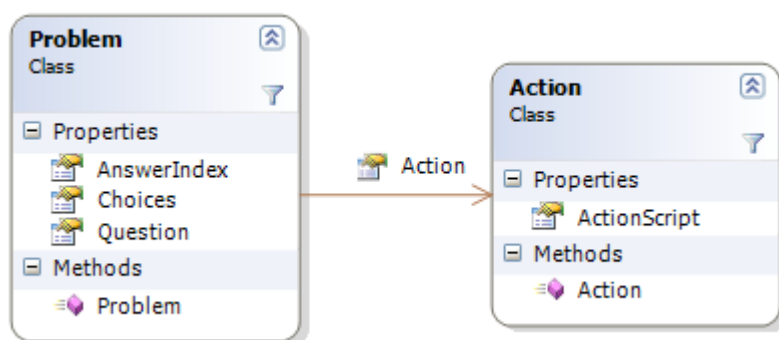
Súčasťou prototypu je implementácia služieb doménových znalostí. Navrhli a implementovali sme rámec, slúžiaci ako základ pre služby doménových znalostí a službu poskytujúcu otázky z domény matematiky.

7.1.1.1.2 Rámec doménových znalostí

Rámec doménových znalostí predstavuje základ pre tvorbu služieb doménových znalostí. Skladá sa z dvoch vrstiev. Prvá modeluje fyzické uloženie dát (dátová vrstva) a druhá problémovú oblasť (business vrstva).

7.1.1.1.2.1 Business vrstva

Business vrstva modeluje doménové problémy. Trieda problém reprezentuje otázku. Je definovaný textom otázky, možnosťami, a správnou odpoveďou. Zatiaľ neberieme do úvahy otázky s viacerými správnymi odpoveďami. Problém môže mať asociovanú akciu, ktorá sa vykoná, v prípade, že hráč správne odpovie. Akcia je vo forme skriptu, ktorý vykoná implementovaný skriptovací engine.



Obrázok 7-3 Triedy business vrstvy rámca aktivít

7.1.1.1.2.2 Dátová vrstva

Dátová vrstva abstrahuje fyzické ukladanie dát. Rozhranie ukladania dát predstavuje abstraktná trieda DAC (Data Access Component) s metódami na pridanie problému, výber všetkých problémov a výber náhodného problému.

V prototyp sme implementovali jednoduchý DAC, ktorý ukladá dáta vo forme XML súboru. Jeho realizáciou je trieda XmlDAC.

7.1.1.1.3 Matematické doménové znalosti

V prototyp sme implementovali službu matematických doménových znalostí, ktorá poskytuje jednoduché matematické príklady. Overili sme tým rámec doménových znalostí.

Pridávanie znalostí bolo vyriešené veľmi jednoduchou ASP.NET stránkou, obsahujúcou formulár pre pridanie nového problému.

Závery

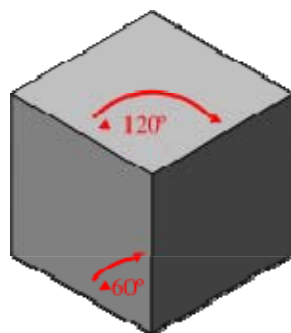
Implementáciou serverových častí prototypu sme overili vhodnosť použitia rámca Windows Communication Foundation pre potreby implementácie navrhovaného typu aplikácie. Tiež sme overili navrhnutú architektúru orientovanú na služby implementáciou niekoľkých služieb a protokolu pre prepínanie ich využívania klientom.

7.1.2 Herný klient

Na strane klienta sme sa rozhodli implementovať prototyp grafického prezentačného rámca za účelom overenia vhodnosti technológie Microsoft XNA pre potreby realizácie grafickej prezentačnej vrstvy. Vzhľadom na väčšie množstvo možných riešení bolo nutné nájsť najvhodnejšie z hľadiska realizácie projektu s dôrazom na jednoduchosť a požadovanú funkcionálnosť.

Spôsob zobrazenia

Grafické zobrazenie je založené na izometrickej projekcii (viď obrázok č. 7-4), ktorá slúži na reprezentáciu trojrozmerných objektov v dvoch rozmeroch. Pre túto grafickú projekciu sme sa rozhodli z dôvodu znížených nárokov na realizáciu, jednoduchosť pri tvorbe prezentovaných grafických prvkov ako sú virtuálne herné prostredie, jednotlivé entity, animácie a iné.



Obrázok 7-4. Izometrická projekcia

Grafické rozhranie klienta

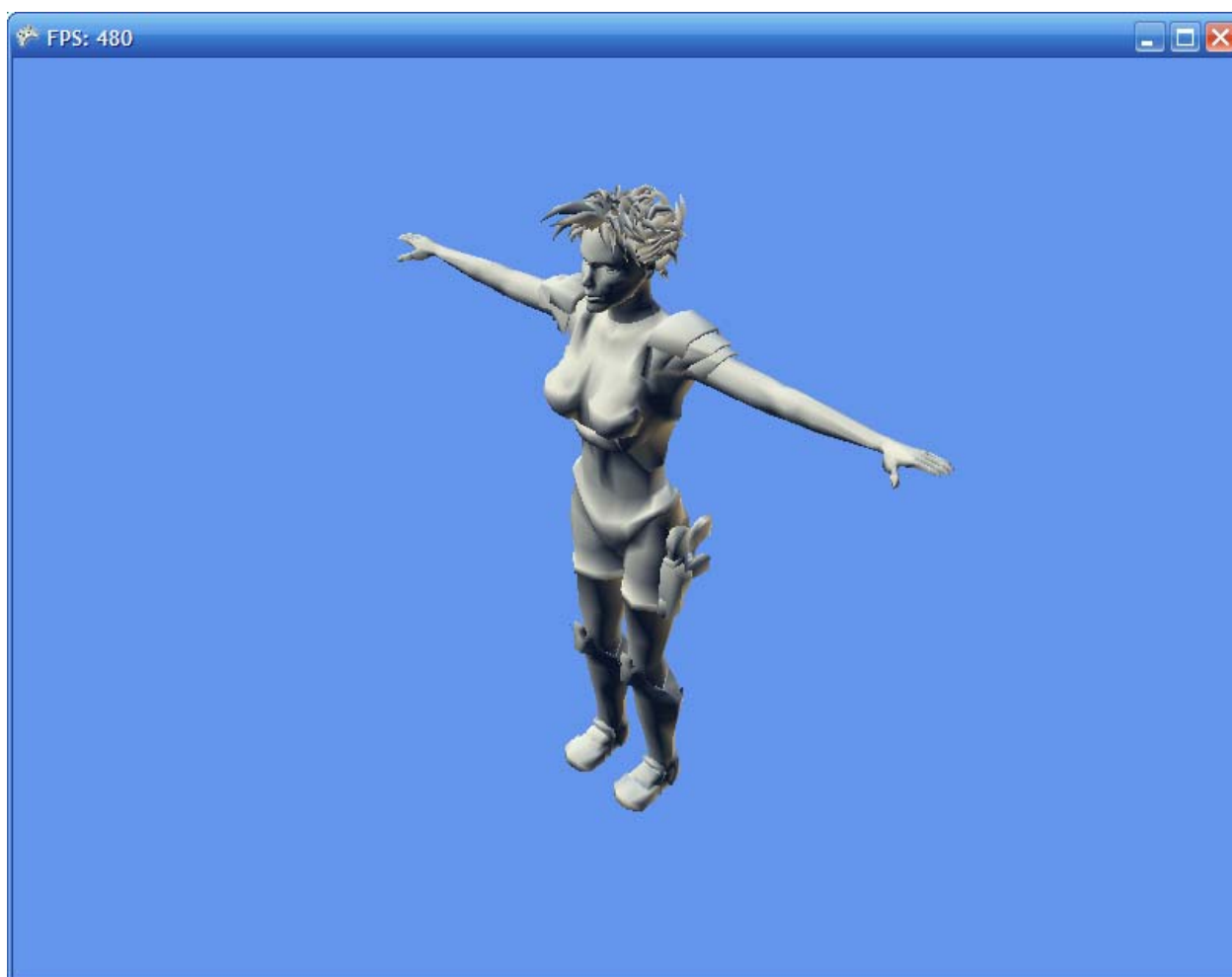
Virtuálne herné prostredie je reprezentované množinou políček usporiadaných do obdĺžnikových alebo štvorcových máp. Tieto je možné vytvárať pomocou príslušného editora implementovaného v rámci realizácie prototypu a použitím spoločnej knižnice je možné následne mapy importovať a prezentovať ako virtuálne herné prostredie (viď obrázok č. 7-5). Grafické rozhranie poskytuje okrem izometrického zobrazenia aj plynulé posúvanie obrazovky pri pohybe hráča.



Obrázok 7-5. Grafické rozhranie klienta

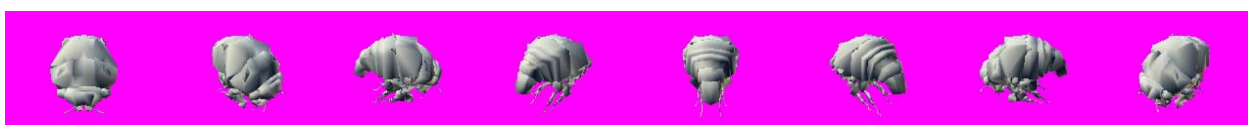
Nástroj pre export modelov

Postavy hráčov ale aj iné herné modely môžu byť vytvorené externými aplikáciami ako 3D modely a pomocou nástroja implementovaného v rámci prototypu (viď obrázok č. 7-6) je ich možné načítať a otočiť tak, aby bola ich herná grafická reprezentácia konzistentná s izometrickým zobrazením.



Obrázok 7-6 Nástroj na export 3D modelov pre izometrické zobrazenie

Výstupom nástroja je 2D reprezentácia modelu vo všetkých izometrických otočeniach požadovaných klientom (viď obrázok č. 7-7). Týmto spôsobom je možné zachytiť aj animácie modelu, prípadne iné neštandardné pózy. Nástroj dokáže pracovať s 3D modelmi formátu Microsoft Direct3D (.x) a AutoDesk FBX (.fbx).



Obrázok 7-7 Exportovaný model s izometrickými otočeniami

Integrácia komunikačného rámca

Klientska časť prototypu integruje rozhranie pre komunikáciu so serverom. Umožňuje pripojenie a odpojenie sa zo serveru, interpretáciu zmien herného sveta a posielanie požiadaviek na pohyb hráča. Na základe poznatkov o zmene herného sveta poskytnutých serverom dokáže graficky reprezentovať okolité prostredie vrátane iných pripojených hráčov.

Integrácia funkcionality hľadania cesty

Na strane klienta je tiež integrovaná funkcionality hľadania cesty v rámci virtuálneho herného prostredia. Hľadanie je realizované pomocou externej knižnice navrhutej v rámci

implementácie prototypu, ktorá umožňuje nájsť cestu z bodu A do bodu B na základe internej reprezentácie mapy herného sveta obsahujúcej informácie o priechodnosti jednotlivých políček. Samotné hľadanie je založené na algoritme A*.

Závery

Implementáciou klientských častí prototypu sme overili vhodnosť grafického rámca Microsoft XNA pre potreby realizácie projektu. Prototyp demonštroval jednoduchosť použitia technológie, schopnosť komunikácie so serverom a integrácie s editorom máp a knižnicou pre hľadanie cesty v hernom svete.

7.1.3 Editor máp

Súčasťou prototypu je aj prototyp editora máp, ktorého účelom je poskytnúť možnosť vytvárať a editovať políčkové mapy herného sveta a exportovať mapy v špecifikovanom formáte desktopovému klientovi a službe herného sveta. Je evolučný a bol vytvorený kvôli potrebe pochopenia možností a potrieb kladených na vytváranie hernej mapy ako aj jej exportu. Používa sa pre vytváranie máp pre prototyp klienta a údajov o priechodnosti políček na mape prototypu služby herného sveta.

V prvej časti je opísaná aplikácia z hľadiska funkcionality a v druhej časti spôsob exportovania údajov ostatným častiam systému.

Opis funkcionality

Mapa sa skladá z pozadia a z objektov. Editor umožňuje vytvorenie mapy so špecifikovanou výškou a šírkou.

Obrázky je možné načítavať jednotlivo alebo hromadne z adresára. Pri odobratí obrázku sú z mapy odobraté všetky políčka, pre ktoré bol zdrojovým obrázkom. Pre každé políčko je možné určiť

- *Zdrojový obrázok* : Atribút určuje, aký obrázok sa použije pri vykresľovaní políček. Dve rôzne políčka môžu mať ten istý zdrojový obrázok, čo je výhodné, ak je potrebné, aby na mape existovali dve opticky rovnaké políčka s rôznymi inými vlastnosťami.
- *Meno (Name)* : Pomenovanie pre účely editora
- *Priechodnosť (Accessibility)* : Určuje, či je možné, aby na políčko vstúpil hráč (napríklad tráva obyčajne býva prístupná, kým voda a steny nie)
- *Typ (IsObject)* : Určuje, či sa jedná o pozadie alebo objekt. Objekty môžu zaberat' väčšiu plochu, ako rozmer políčka a môžu mať určenú svoju vlastnú z-súradnicu.
- *Veľkosť (Size)* : Dvojzložkový vektor opisujúci šírku a výšku políčka. Políčka pozadia môžu zaberat' len jednotkovú veľkosť.
- *Z-súradnica (ZOrder)* : Určuje, ktoré políčko je nad ktorým. Políčka s vyššou z-súradnicou prekrývajú políčka s nižšou. Políčka pozadia majú túto hodnotu rovnú nule.

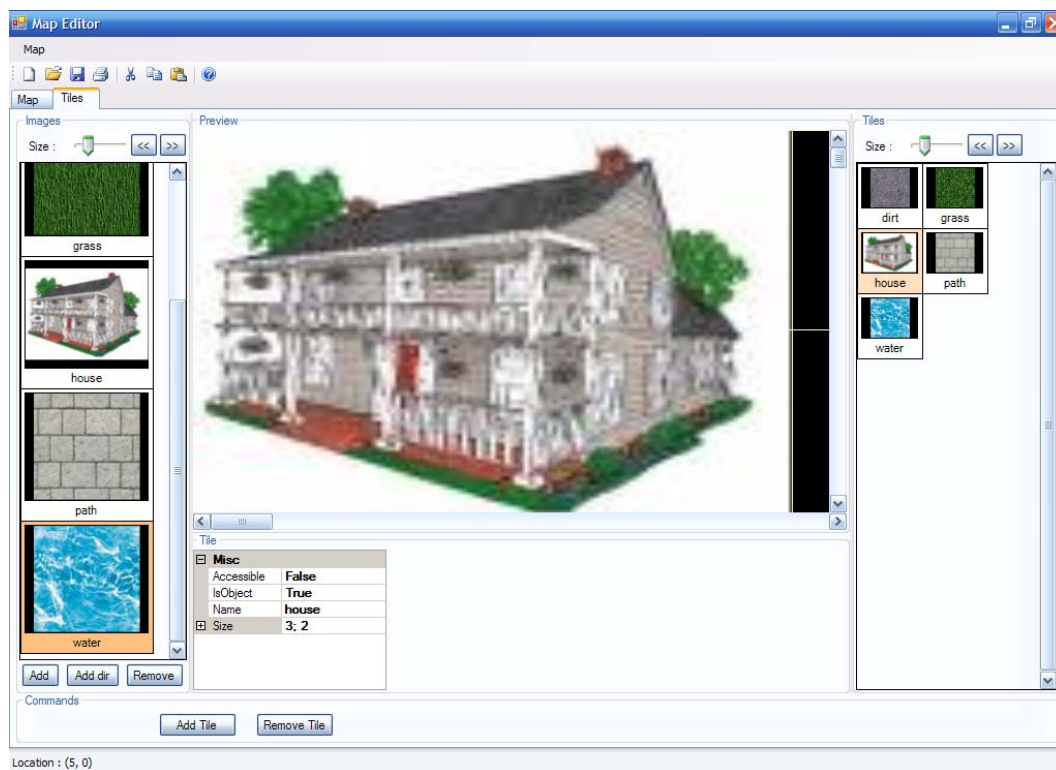
Vytvorenie mapy z dlaždíc sa vykonáva ich postupným pridávaním do editovacej oblasti (obrázok č. 2) . S mapou je možné pracovať v ľubovoľnom priblížení z vopred stanoveného intervalu.

Okrem týchto činností umožňuje editor ukladanie a načítavanie vytvorených máp.. Údaje o mape sú uložené v dvoch súboroch:

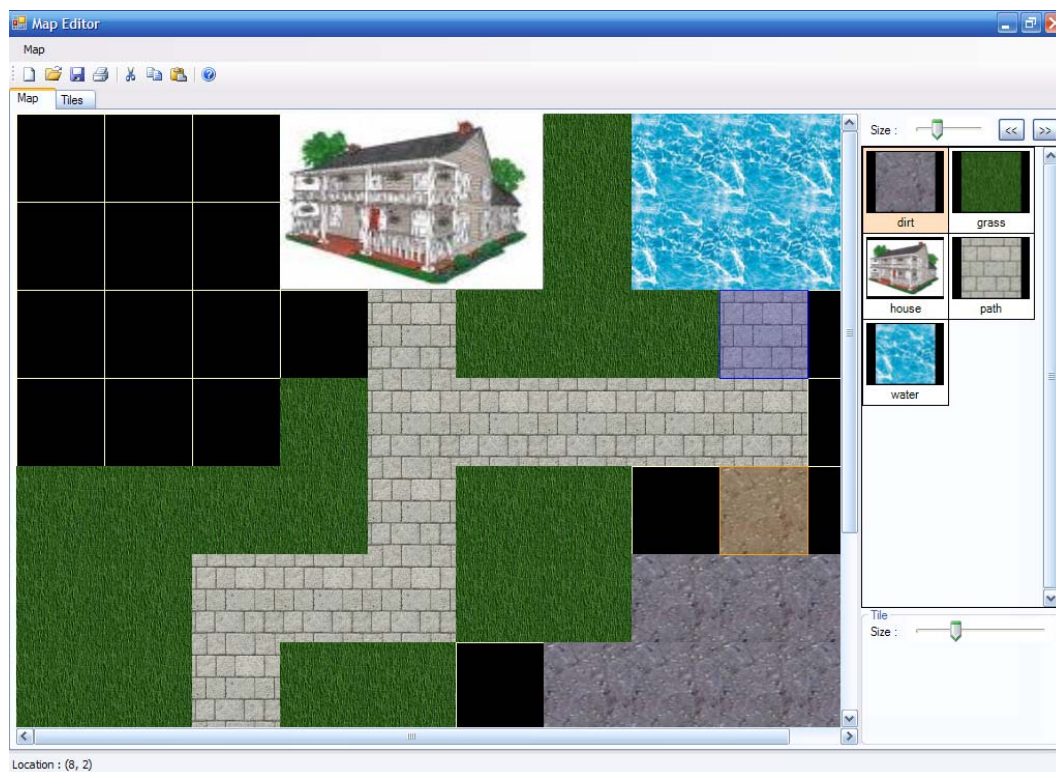
- **.imp* : Obsahuje všetky zdrojové obrázky dlaždíc nachádzajúcich sa v mape
- **.map* : Obsahuje údaje o samotnej mape

Pri načítavaní do editora je potrebné najprv načítať súbor s obrázkami a až potom súbor s mapou.

Rozdelenie dát mapy na niekoľko súborov sa prejaví v zníženej priestorovej náročnosti a v možnosti vytvorenia dlaždicových setov pre rôzne typy oblastí.



Obrázok 7-8 Definovanie políčok (používané obrázky políčok sú len ilustračné)

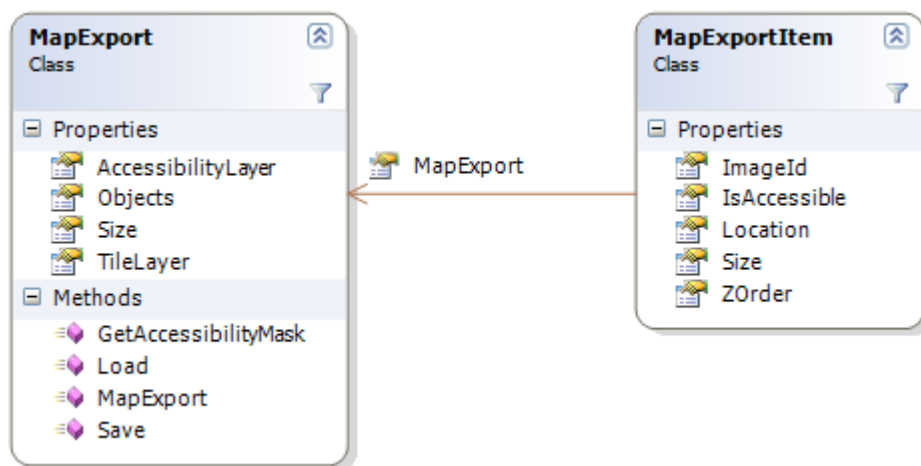


Obrázok 7-9 Vytváranie mapy (používané obrázky políčok sú len ilustračné)

Exportný formát

V tejto časti sú bližšie popísané exportné súbory.

- *.imp : V súbore sa nachádza binárne serializované pole obrázkov
- *.map : Obsah súboru tvoria serializované triedy zobrazené na obrázku č. 3.



Obrázok 7-10 Triedy použité na serializáciu pri exporte mapy

Hlavnou triedou, reprezentujúcou celú mapu je trieda *MapExport*. Obsahuje v sebe nasledujúce atribúty:

- *AccessibilityLayer* : Matica, ktorá určuje prístupnosť jednotlivých políčok pozadia
- *Objects* : Každý objekt na mape (teda nie pozadie) je reprezentovaný jednou inštanciou triedy *MapExportItem*. Tento atribút reprezentuje pole takýchto objektov.
- *Size* : Veľkosť mapy
- *TileLayer* : Identifikačné číslo zdrojového obrázku políčka pozadia

Trieda *MapExportItem* slúži na reprezentáciu objektov na mape. Obsahuje atribúty:

- *ImageId* : Identifikačné číslo zdrojového obrázka objektu
- *IsAccessible* : Určuje, či je objekt prístupný
- *Location* : Poloha objektu na mape
- *Size* : Veľkosť objektu (koľko políčok zaberá v oboch rozmeroch)
- *ZOrder* : Z-ová súradnica objektu

Výsledok

Navrhli a implementovali sme prototyp editora máp, ktorý je evolučným prototypom a plánujeme ho ďalej rozširovať hlavne podľa požiadaviek ostatných častí systému.

7.1.4 PDA klient

PDA klient je aplikácia, ktorá zatiaľ nebola integrovaná do zostávajúcich častí systému. Jej cieľom bolo analyzovať možnosti vytvorenia rozšíreného herného prostredia pre hráča (učiace sa dieťa) na platforme PDA.

Aplikácia PDA klient pozostáva z logickej a vizuálnej časti. Implementovaná bola najmä logická časť.

Logika aplikácie

Fungovanie aplikácie je založené na simulácii. Súčasťou hernej simulácie je samotný PET, ako aj prostredie, v ktorom sa nachádza. Na hernú simuláciu môžu vplývať dva druhy vplyvov:

- vplyvy zvnútra simulácie
- vplyvy spôsobené intervenciou používateľa

Vplyvy zvnútra aplikácie vyvoláva plánovací podsystem, ktorý funguje na základe plánovacích pravidiel, ktoré na základe času alebo iných veličín (ktoré presnejšie špecifikujeme v ďalšom prototypu) vyvolávajú patričné Akcie.

Vplyvy zvonka vyvoláva používateľ interakciou s používateľským rozhraním, čím priamo vyvoláva Akcie.

Subsystem funguje na princípe generovania a spracovávania akcií.

7.1.4.1.1 Generovanie Akcií

Akcie môže generovať inštancia každej triedy.

7.1.4.1.2 Spracovávanie Akcií

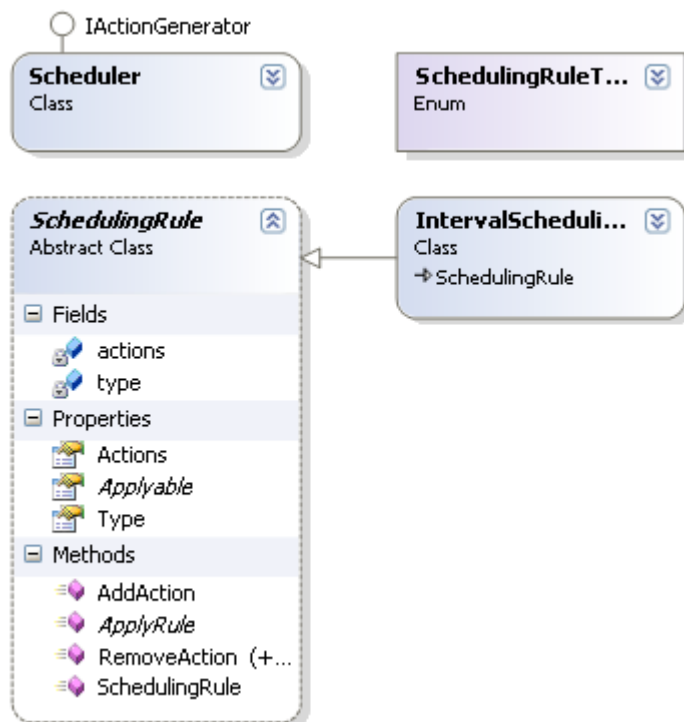
Spracovať akciu znamená zabezpečiť jej vykonanie. Prijat' a následne vykonať akciu môže len objekt, ktorý implementuje rozhranie `IActionHandler`. Každá akcia má danú množinu prijímateľov (Implementujúcich `IActionHandler`), na ktorých sa dokáže spustiť.



Obrázok 7-11 Rozhranie `IActionHandler`

7.1.4.1.3 Plánovanie Akcií

Plánovanie akcií je proces, v ktorom sa na základe časových dát a ďalších informácií spúšťajú Akcie. Informáciu, kedy ktorú akciu spustiť, určuje množina plánovacích pravidiel. Vždy po uplynutí zvoleného časového intervalu sa prehľadáva množina pravidiel, a vyberajú sa aplikovateľné inštanície pravidiel. Tie sa vyhodnotia a vygeneruje sa množina akcií, ktoré sa vykonajú. Trieda Scheduler implementuje rozhranie IActionGenerator. Plánovač je len jeden z možných generátorov akcií (Ďalšími sú napr. Prvky používateľského rozhrania)

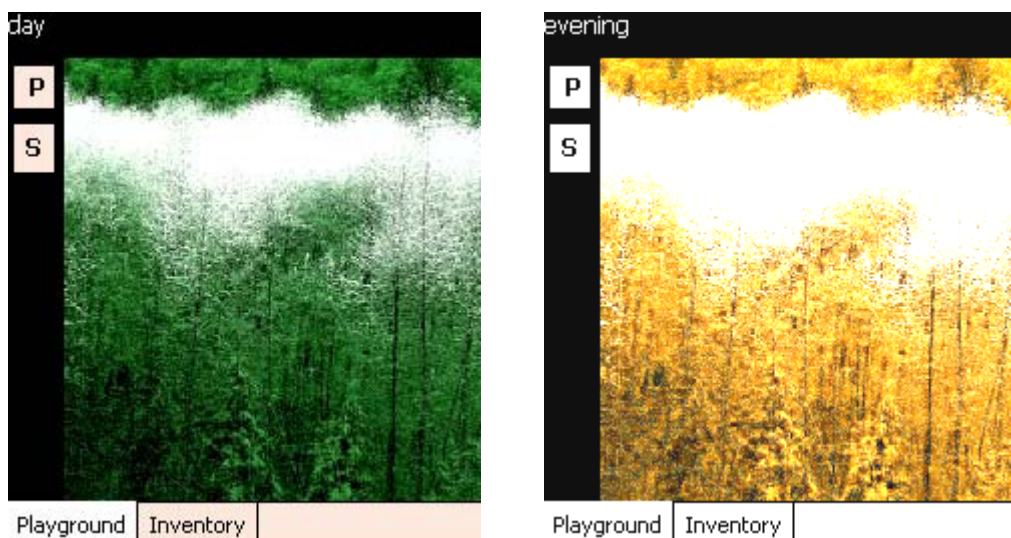


Obrázok 7-12 Plánovací podsystem (jeden z generátorov akcií)

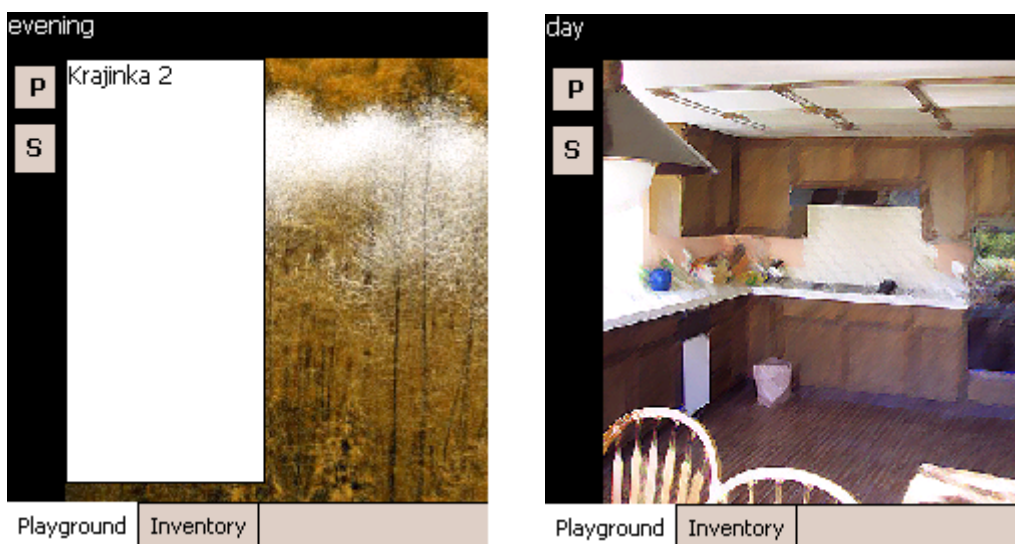
Vizuálna stránka

Z vizuálnej stránky je v prvom prototype naprogramovaný inventár a kontrolka PlaygroundControl, ktorá slúži ako hlavný manipulačný a zobrazovací prostriedok pre prostredie a PET. S jej pomocou je možné meniť vnútorný stav prostredia, v ktorom sa pet nachádza.

Kontrolka umožňuje pohodlnú navigáciu v prostredí vždy len medzi navzájom susediacimi oblasťami. Každá oblasť sa môže nachádzať v rôznych stavoch. Medzi stavmi je možné preklikávať ručne, alebo nastaviť pravidlo, ktoré túto činnosť zautomatizuje (striedanie dňa a noci).



Obrázok 7-13 Oblasť v rôznych stavoch



Obrázok 7-14 Prechod medzi oblasťami

7.1.5 Zhrnutie

Implementovaním prototypu PDA klienta sme zistili možnosti uplatnenia konceptu dynamicky sa meniaceho stavu herného sveta v prenosnom zariadení. Plánujeme rozšíriť navigačné možnosti používateľa implementovaním navigačnej mapky. V ďalšej iterácii plánujeme pridať samotné zvieratko do prostredia, čo si bude vyžadovať prídanie dodatočnej grafiky a dodatočných pravidiel. V ďalšom kroku plánujeme prepojiť PDA klienta s ostatnými aplikáciami balíku.

7.2 Závery prototypovania

Výsledkom prototypovania je funkčný systém, na ktorom sme overili jednotlivé technológie a ich spojenie. Technológie sa ukázali ako správne vybrané a podarilo sa nám ich spojiť bez problémov. Prototyp považujeme za evolučný a plánujeme pokračovať v jeho vývoji ďalšími iteráciami.

8 Spresnenie a doplnenie špecifikácie

V tejto kapitole sa nachádza spresnenie a doplnenie špecifikácie vytvorenej v prvom semestri. Prvá časť obsahuje zhodnotenie prototypovania, ďalšia špecifikuje herné aktivity a v záverečnej časti je opísaná služba *Zvieratko*.

8.1 Zhodnotenie prototypovania

Prototyp vytvorený v minulom semestri bol evolučným, a podľa plánu sme pokračovali v jeho ďalšom rozširovaní. Prototyp zachytil a overil navrhnutú architektúru systému založenú na webových službách a jej rozširovaním pokračujeme vo vývoji ďalej.

Z prototypu evolučným princípom vznikli ďalšie iterácie rozšírenia špecifikácie. Ďalšie kapitoly obsahujú špecifikáciu nových herných aktivít a dodatočnej služby, reprezentovanej v konečnom dôsledku ako podporná aplikácia pre herné aktivity.

8.2 Herné aktivity

V tejto časti sa nachádza špecifikácia herných aktivít, ktoré sú navrhnuté tak, aby pokryli výučbové oblasti matematika, prvouka, čítanie a písanie a hudba. Aktivity zároveň spĺňajú nasledujúce vlastnosti:

- Jednotný design používateľského rozhrania
- Jednoduché ovládanie doplnené rozoznávaním hlasových povelov v prípadoch, keď je to vhodné
- Schopnosť parametrizovaného spúšťania z herného sveta ako aj samostatne
- Úprava používateľského modelu na základe vstupov
- Komunikácia so službou *Zvieratko*, ktorá je opísaná nižšie

8.2.1 Strokes

Hra *Strokes* slúži na rozvoj motorických schopností detí a v neskorších fázach aj na tréning písania jednotlivých písmen.

Úlohou hráča je obtiahnuť predlohu špecifikovanú ako postupnosť ťahov (pozri obr. 8-1). Hra priebežne hodnotí presnosť ťahu a poskytuje hráčovi spätnú väzbu prostredníctvom zvieratka, ktoré ho chváli, ak sa mu darí, prípadne povzbudzuje, v prípade že sa mu darí menej. Pri hodnotení sa berie do úvahy aj rýchlosť hráča pri obtahovaní.

Hra okrem režimu pre jedného hráča, v ktorom môže fungovať aj ako nezávislá aplikácia, ponúka aj možnosť hráčov navzájom súťažiť v obtahovaní predlohy. Spojení sú cez sieťové rozhranie, teda každý obtahuje na tablete pri svojom počítači a vidí ťahy druhého hráča inou farbou. Vyhráva hráč, ktorý obtiahne predlohu presnejšie a zároveň je v tom rýchlejší.

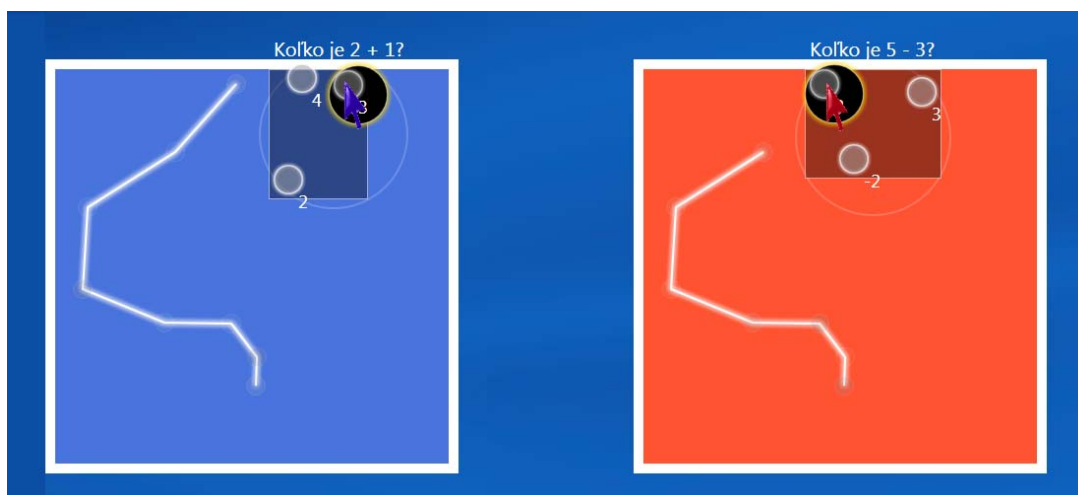


Obrázok 8-1. Návrh používateľského rozhrania hry Strokes.

8.2.2 SmartLines

Hra *Smart Lines* slúži na precvičovanie vedomostí z ľubovoľnej domény, ktoré poskytuje úložisko doménových znalostí s odpoveďami s možnosťou výberu. Príkladom takejto domény môže byť matematika. V hre dieťa postupuje tak, že spája myšou body, ktoré tvoria obrys obrázka. Pri spájaní bodov vyberá správnu odpoveď na zadanú otázku, pomocou ktorej určí nasledujúci bod vo vykresľovanom obrázku. Sledujú sa nielen správne odpovede, resp. počet pokusov správnych odpovedí, ale aj čas odpovede.

Hru môže hrať jeden hráč v hernom svete alebo si ju spustí z hlavného menu. Zároveň ju môžu hrať niekoľkí hráči pri jednom počítači s využitím viacerých myší, kedy je obrazovka rozdelená na niekoľko častí (pozri obr. 8-2).



Obrázok 8-2. Návrh používateľského rozhrania hry SmartLines.

8.2.3 Matematické balóny

V hre *Matematické balóny* hráči trénujú matematické príklady. Hra im zadáva po jednom príklad s viacerými možnosťami výberu odpovede a tieto možnosti zobrazí vo forme balónov (pozri obr. 8-3). Dieťa kliknutím na správny balón vyberie odpoveď.

Hra sa prispôsobuje vedomostiam dieťaťa odhadnutým na základe jeho odpovedí. Počas trénovania riešenia úloh sa hráčovi prispôsobuje na základe jeho úspešnosti. Príklady sú rozdelené do niekoľkých typov a tým sa trénujú postupne sčítanie malých čísel, precvičuje násobilka a nakoniec operácie s väčšími číslami.

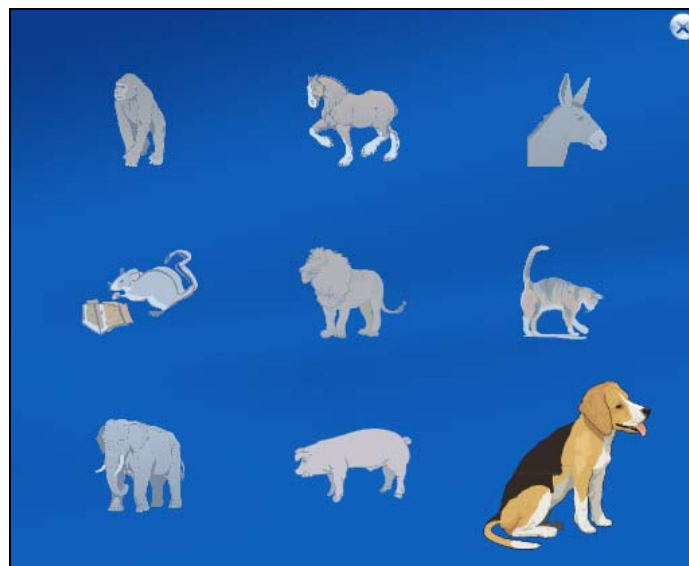


Obrázok 8-3. Návrh používateľského rozhrania hry *Matematické balóny*.

8.2.4 Zvuky

Úlohou detí v hre *Rozpoznávanie zvukov* je priradovať zvuky k zodpovedajúcim obrázkom. Hru je možné využiť na výučbu viacerých domén, na obr. 8-4 je zobrazené rozhranie pri precvičovaní zvukov zvierat. Ďalším príkladom použitia hry rozpoznávanie zvukov je použitie pre hudobnú doménu pri rozpoznávaní zvukov nástrojov.

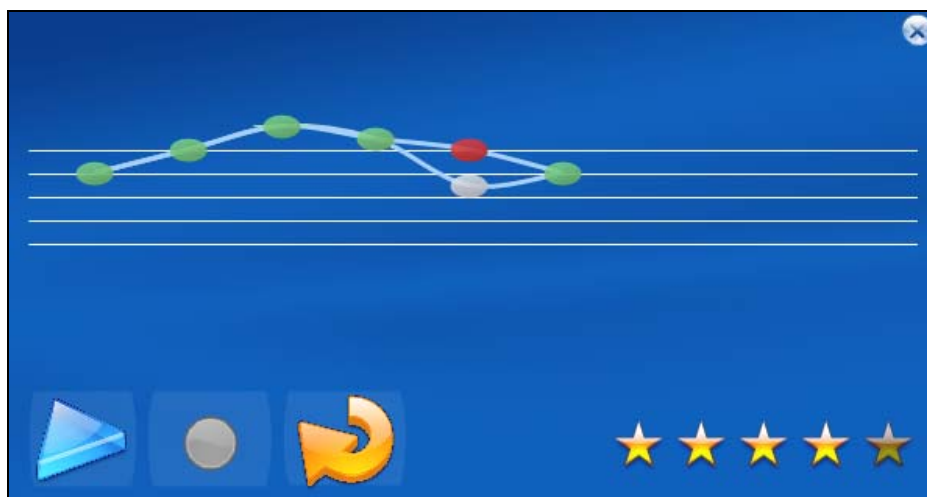
Hra obsahuje dva režimy, prehliadací a skúšací. V prvom môže dieťa počúvať zvuky kliknutím na príslušný obrázok a takto sa učiť a trénovať v rozpoznávaní zvukov, v druhom mu je najprv zahratá ukážka a jeho úlohou je vybrať, ktorému zobrazenému predmetu patrí.



Obrázok 8-4. Návrh používateľského rozhrania hry Zvuky.

8.2.5 Opakovanie melódie

Cieľom hry *Opakovanie melódie* je rozvíjať hudobnú predstavu a sluch. Úlohou deti je zopakovať krátku melodickú ukážku spevom, písaním alebo na hudobnom nástroji napríklad na flaute. Na obrazovke sa pritom zobrazujú tóny úlohy ako aj tóny, ktoré dieťa postupne zadáva (pozri obr. 8-5). Rozpoznávanie výšky tónu je realizované s použitím spektrálnej analýzy. Úspešnosť dieťaťa sa vyhodnotí pomocou metriky, ktorá dokáže určiť podobnosť dvoch melódii a je mu poskytnutá spätná väzba vysvietením určitého počtu hviezdíček a prostredníctvom jeho zvieratka.

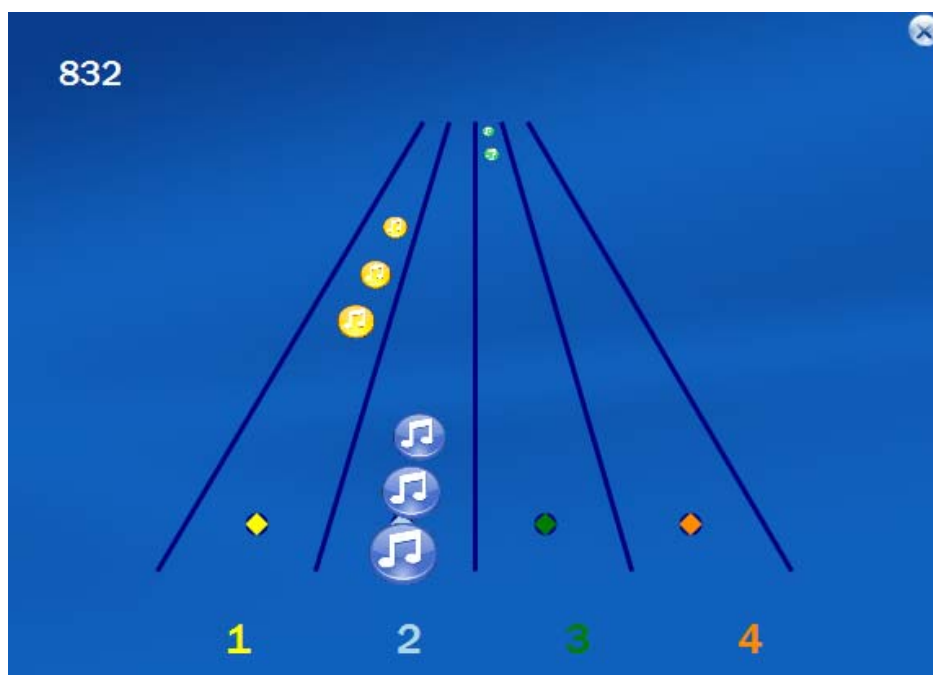


Obrázok 8-5. Návrh používateľského rozhrania hry Opakovanie melódie.

8.2.6 Malý hudobník

V hre má dieťa za úlohu za sprievodu pesničky trafiť noty pohybujúce sa z vrchného okraja obrazovky dole tak, aby boli čo najbližšie ku značke (pozri obr. 8-6). Pri tom musí držať číslo alebo čísla tých radov, v ktorých sa značky nachádzajú. Pohybujúce sa noty prechádzajú

značkou synchronizovane s hudobným sprievodom, takže dieťa môže využiť a rozvíjať rytmické cítenie. Počet radov ako aj obtiažnosť úloh sú rôzne podľa veku dieťaťa.



Obrázok 8-6. Návrh používateľského rozhrania hry Malý hudobník.

8.3 Oddychové aktivity

V tejto časti sa nachádza popis oddychových aktivít, ktoré dieťaťu hra poskytuje.

8.3.1 Čítanie rozprávok

Aplikácia zobrazuje text rozprávky spolu s obrázkom, pričom prehráva jej nahovorený text. (obrázok 8-7). Taktiež umožňuje rodičom nahovoriť rozprávku a tieto dáta uložiť na prehranie pri nasledujúcom použití dieťaťom.



Muzikálna myš



Muzikálna myš vždy chcela dirigovať orchester. Ale doteraz jej to nikto nedovolil. Hovorili že je príliš malá na to aby dirigovala veľký orchester. "Nevadí", premýšľala muzikálna myš. "Spravím si svoj vlastný malý orchester, a budeme hrať nejaké skvelé skladby. Budeme potrebovať bubon aby robil bub bub bub. A budeme potrebovať trúbku aby robila tramtarará. A husle aby robili...akokoľvek robia husle. A ja budem dirigent. Muzikálna myš teda

ďala levovi bubon, husle tigrovi a medveďovi dala trúbku. "Skvelé!" povedala muzikálna myš, "teraz je čas niečo zahrať!" "Skvelé!" povedal lev, tancujúc okolo a hrajúc na bubon. "Skvelé!" povedal medveď, tancujúc a trúbiac na trúbku.

Obrázok 8-7. Návrh aktivity Čítanie rozprávok.

8.3.2 Dlhé slová

Táto aktivita náhodne zobrazuje rôzne dlhé slová. Jej návrh je zobrazený na obrázku 8-8.



Obrázok 8-8. Návrh aktivity Dlhé slová.

8.4 Služba Zvieratko

Služba zvieratko predstavuje aplikáciu, ktorá slúži na poskytovanie interakcie hráčovi. Jednotlivé hry ju môžu využívať na syntézu hlasových upozornení a prezentovanie emočných stavov zvieratka.

Služba beží v procese odseparovanom od serverových služieb a na strane služby poskytuje grafické animované znázornenie zvieratka. Rozhranie umožňuje nasledovné operácie:

- Vyslov zadanú vetu
- Prejav zadanú emóciu

Prípady použitia služby Zvieratko sú uvedené v špecifikácii hier Strokes a SmartLines.

9 Podrobný návrh

Táto kapitola obsahuje popis návrhu systému. Je rozčlenená nasledovne: Prvá podkapitola opisuje návrh jednotlivých služieb. Ďalšia podkapitola opisuje jednotlivé metódy a prístupy aplikované s cieľom zabezpečiť adaptívnosť výučby a obsahuje popis metódy navrhutej pre ohodnocovanie úspešnosti pri hre Strokes. Posledná podkapitola popisuje návrh jednotlivých aplikácií

9.1 Služby

9.1.1 Služba svet

Služba svet modeluje objektívnu realitu. Poskytuje súradnicovú sústavu, v ktorej sa hráči pohybujú. Zabezpečuje interakciu hráčov s entitami a navzájom a umožňuje im vytvárať tímy.

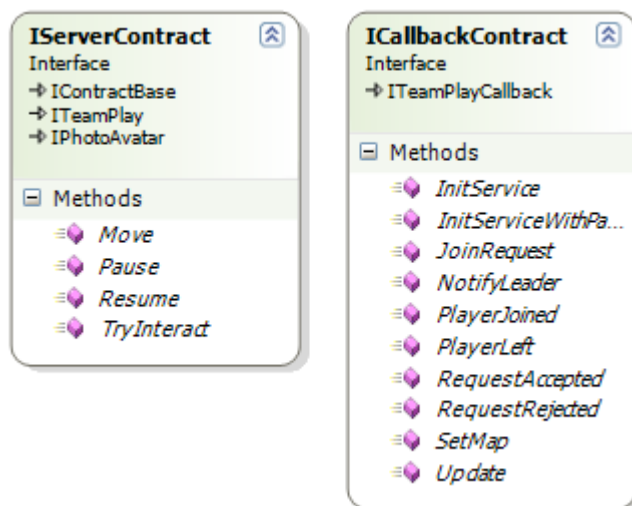
Rozhranie služby svet

Rozhranie služby svet predstavuje rozhranie *IServerContract*. Umožňuje pohyb hráčov a zastavenie prípadne znovuprihlásenie sa k odberu notifikačných správ zo služby.

Rozhranie *IServerContract* implementuje rozhranie *ITeamPlay*, slúžiace na manažment tímov. Rozhranie *IPhotoAvatar* slúži na manažment avatarov hráčov. Poskytuje metódy *SetPhotoAvatar* a *GetPhotoAvatar*.

Pre prijímanie správ na strane klienta slúži rozhranie *ICallbackContract*. Ako je zrejmé z diagramu na obrázku č. 9-1 **Error! Reference source not found.**, rozhranie *ICallbackContract* obsahuje metódy *Update*, *InitService*, *InitServiceWithParameters*, *SetMap* a callback metódy rozhrania *ITeamPlayCallback*.

Metóda *Update* slúži na notifikáciu klientov o zmene stavu sveta. Odovzdáva na klienta dátovú štruktúru *GameState* (viď ďalej). Metóda *InitService* slúži pre zmenu služby, s ktorou používateľ na strane klienta interaguje (viď popis protokolu prihlasovania s k službám v kapitole 9.1.2)

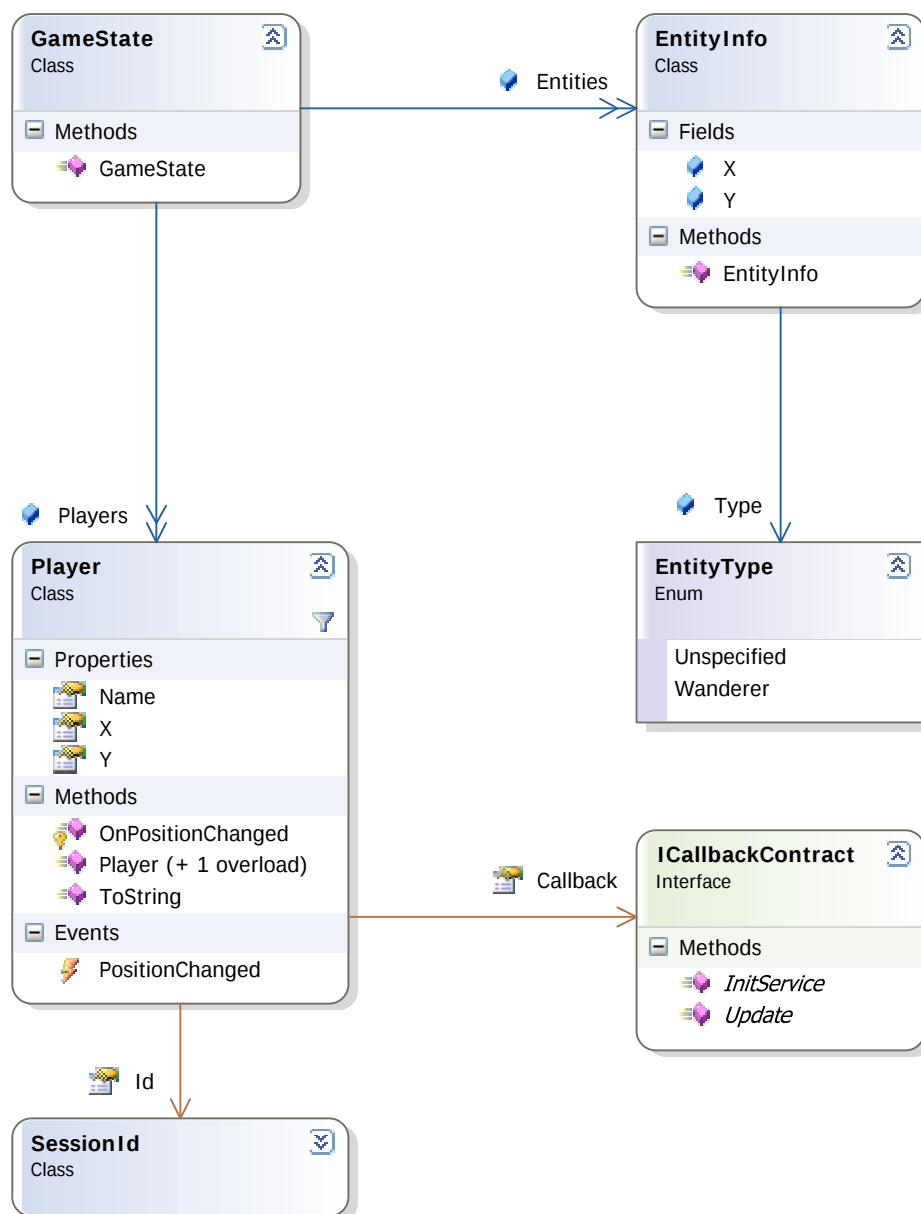


Obrázok 9-1. Rozhrania služby World.

Dáta služby svet

Hlavnú dátovú štruktúru služby svet predstavuje trieda *GameState* (vid' obrázok 9-2). Táto obsahuje zoznam hráčov a zoznam ostatných entít sveta. Hráč je reprezentovaný ako inštancia triedy *Player*. Hráč na strane servera vystavuje udalosť *PositionChanged*, ktorou notifikuje svet pri zmene polohy. Toto využíva trieda *World* (vid' ďalej). Na triedu *Player* sa viaže inštancia triedy implementujúcej rozhranie *ICallbackContract*, cez ktorú je možné zo strany servera posilať správy na pripojeného klienta reprezentujúceho daného hráča. Ďalej obsahuje trieda *Player* referenciu na inštanciu *SessionId*, ktorá slúži ako jednoznačný identifikátor pripojeného hráča.

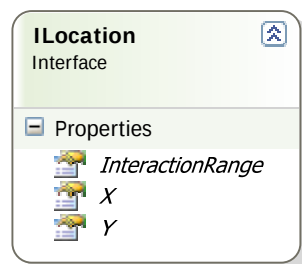
Trieda *EntityInfo* zahŕňa informáciu o súradniciach danej entity v svete vo forme koordinátov *x* a *y* a typ entity reprezentovaný ako hodnota enumerácie *EntityType* (vid' obrázok 9-2).



Obrázok 9-2. Trieda GameState a asociované triedy.

Reprezentácia herných entít

Základná trieda reprezentujúca entity v hre je abstraktná trieda *Entity*. Všetky objekty v hre sú odvodené z tejto triedy. Trieda *Entity* implementuje rozhranie *ILocation*, ktorého metódy sú zrejmé z diagramu na obrázku č. 9-3.

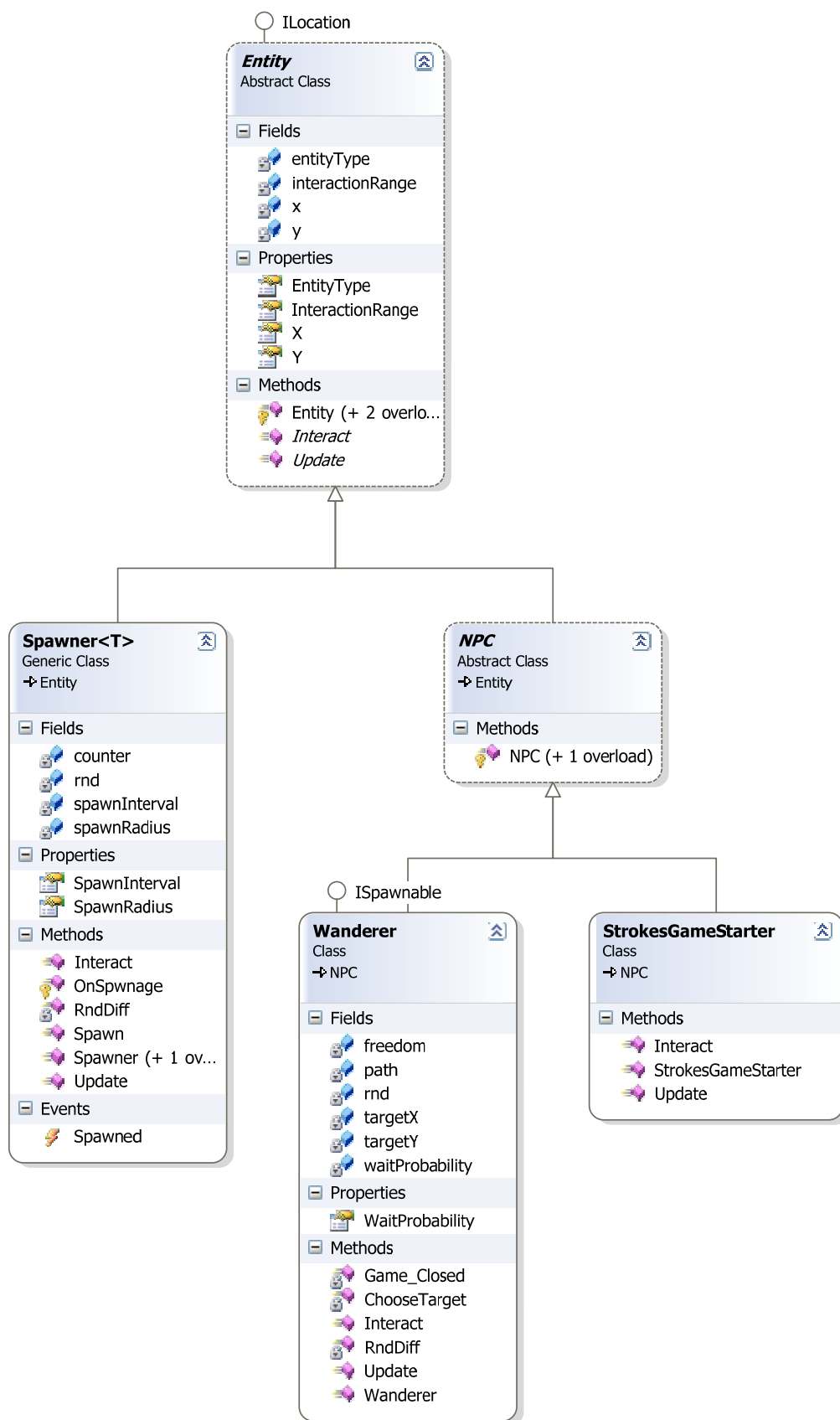


Obrázok 9-3. Rozhranie *ILocation*.

Hierarchia základných entít je vyobrazená na obrázku č. **Error! Reference source not found.** Trieda *Entity* obsahuje abstraktné metódy *Update* a *Interact*. Metóda *Update* umožňuje zmenu stavu entity v každom kroku hry (viď ďalej – popis triedy *World*) a volaním metódy *Interact* v hernej slučke je zabezpečená možnosť interakcie medzi hernými entitami a hráčmi vo vzdialenosti špecifikovanej metódami rozhrania *ILocation* danej entity.

Generická trieda *Spawner<T>* slúži pre vytváranie nových inštancií entít typu špecifikovaného typovým parametrom *T* triedy *Spawner<T>*. Je možné definovať interval vytvárania entít, miesto, kde budú vytvorené aj rádius ich náhodného rozmiestnenia pri vytváraní.

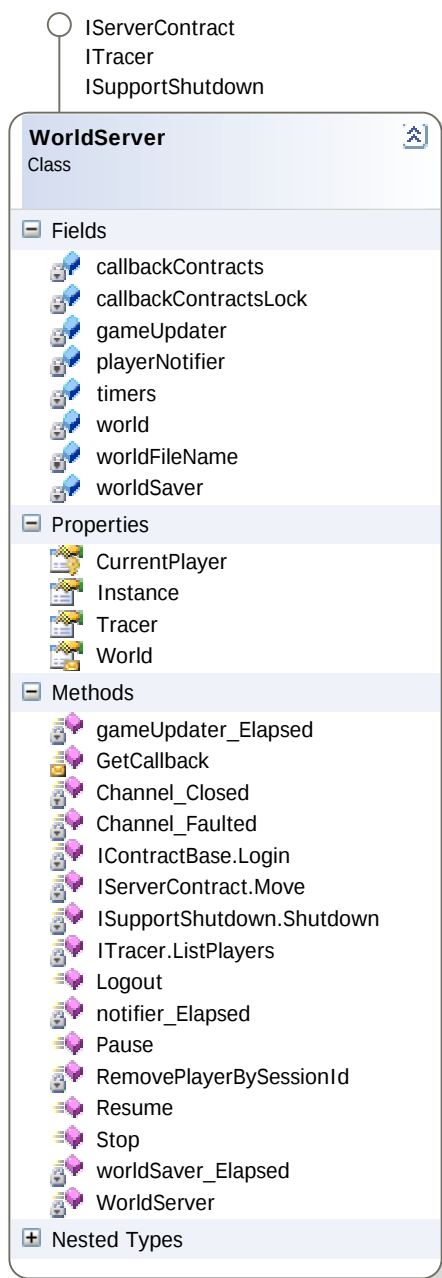
Trieda *NPC* (*non-player character*) reprezentuje postavy v hre, ktoré nie sú ovládané hráčmi. Typicky ide o postavy, ktoré zadávajú hráčovi questy prípadne iné úlohy. Na diagrame na obrázku č. 9-4 sú zobrazené triedy *Wanderer* a *StrokesGameStarter* odvodené od triedy *NPC*.



Obrázok 9-4. Trieda Entity a základné asociované triedy.

Herná slučka

Beh hernej slučky zabezpečuje trieda *WorldServer*, ktorá zároveň implementuje rozhranie *IServerContract*. Ďalej implementuje rozhranie *ITracer* slúžiace na trasovanie výstupov pre testovacie účely a *ISupportShutdown* podporované aplikáciou hostujúcou službu a volané pri vypínaní služby. V triede *WorldServer* zabezpečí uloženie stavu sveta pri vypínaní servera. Všetky dátové členy a metódy triedy *WorldServer* sú zobrazené v diagrame na obrázku č. 9-5.



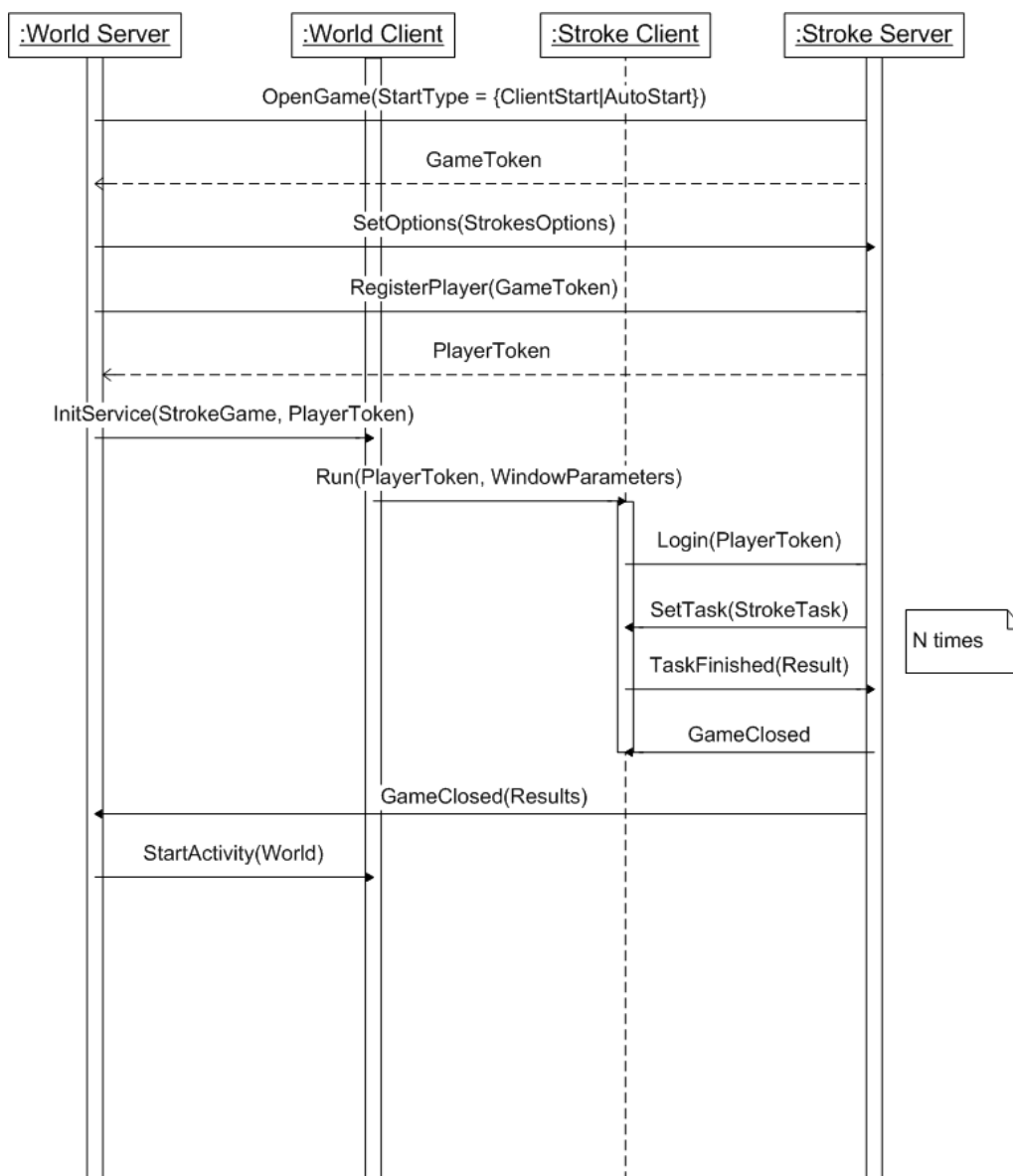
Obrázok 9-5. Trieda *WorldServer*.

Herná slučka beží na časovači, ktorý je inštančnou premennou služby *WorldServer*.

9.1.2 Rámec pre vytváranie služieb aktivít

Rámec pre vytváranie služieb aktivít slúži ako základ pri programovaní komponentov aktivít bežiacich na serveri. Základnou triedou je trieda *GameBase<TGameInstace, TCallbackContract>*, ktorá zastrešuje základnú logiku vytvárania hier, registrovania a prihlasovania hráčov do konkrétnych hier a riadenia životného cyklu inštancií jednotlivých hier. Trieda *GameInstance<TCallbackContract>* predstavuje konkrétnu implementáciu inštancie hry. Implementuje rozhranie definované typovým parametrom *TCallbackContract* a zastrešuje konkrétnu hernú logiku, protokol a poradie výmeny správ a spustenie a ukončenie konkrétnej inštancie hry.

Hráči sa ku herným aktivitám prihlasujú prostredníctvom tokenov (inštancií triedy *PlayerToken*). Hry sa vytvárajú volaním metódy *OpenGame* na rozhraní vystavenom inštanciou *GameBase* a sú reprezentované inštanciou *GameTokenu*. Diagram zachytávajúci vytvorenie hry, prihlasovanie hráčov ku hernej aktivite, priebeh a ukončenie hry v prípade, že hra je spúšťaná zo servera mechanizmom *InitService* (viď. kapitola 9.1.1) je na obrázku 9-6.



Obrázok 9-6. Mechanizmus *InitService*.

Aktivita môže byť spúšťaná tiež z klienta, a to v prípade že hráč hrá konkrétnu hru v *standalone* móde. Vtedy volá metódy *OpenGame* a *RegisterPlayer* namiesto služby *World* priamo herný klient, s ktorým interaguje hráč.

9.1.3 Rámec pre vytváranie úložísk doménových znalostí

Služby doménových znalostí slúžia na ukladanie znalostí v jednotlivých doménach, ku ktorým majú prístup služby aktivít a kladením otázok uložených v úložiskách služieb doménových znalostí overujú vedomosti používateľov v daných doménach.

Úlohy majú dvojitú povahu – prvou sú faktické úlohy, overujúce znalosti používateľov, a druhou úlohy overujúce ich schopnosti, viazané na konkrétne aplikácie.

Webové služby, na ktorých je systém postavený, podmieňujú, že predmetom výmeny dát medzi klientom a serverom sú dáta, pre ktoré existuje XML reprezentácia. Toto sa dá využiť na strane servera na jednoduché ukladanie jednotlivých úloh vo forme XML súborov.

Rámec pre vytváranie úložísk doménových znalostí poskytuje všeobecný mechanizmus ukladania a sprístupňovania XML reprezentácií inštancií znalostí. Generická trieda *StorageManager<T>* (viď obr. č. ...) poskytuje možnosť riadiť ukladanie inštancií typu T v súborovom systéme na serveri a sprístupňovať ich.

Metódy triedy *StorageManager<T>*:

```
bool Store(T t, string identifier)
```

- uloží inštanciu pre ďalšie sprístupnenie pod identifikátorom *identifier*.

```
T GetRandom()
```

- vyberie náhodnú inštanciu z úložiska a vráti ju

```
T Get(string identifier)
```

- vráti uloženú inštanciu s daným identifikátorom alebo *null*, ak neexistuje

```
List<T> List()
```

- vráti všetky uložené inštancie

```
List<string> ListIDs()
```

- vráti všetky identifikátory uložených inštancií

9.1.4 Služba StoryTeller

Služba *StoryTeller* slúži na ukladanie a sprístupňovanie rozprávok. Jej rozhranie umožňuje pridať rozprávku so zadaným identifikátorom a vybrať konkrétnu alebo náhodnú vloženú rozprávku.

Uložené rozprávky sú reprezentované ako HTML dokumenty. Klient služby môže po spracovaní riadiacich značiek v rámci HTML obsahu rozprávky použiť TTS rozhranie pre prečítanie rozprávky syntetizovaným hlasom.

9.1.5 Služba Zvieratko

Služba zvieratko slúži ako platforma pre interakciu používateľa so zvieratkom reprezentovaným vo forme animovaného obrázku a schopného zvukovej interakcie cez TTS rozhranie.

Zvieratko je schopné vysloviť zadanú vetu a vyjadriť emócie animovaním obrázku, ktorý ho reprezentuje.

Rozhranie služby

Rozhranie služby zvieratko poskytuje nasledovné metódy:

```
Say(string sentence)
```

- vysloví zadaný reťazec

```
Animate(PetAnimation animation)
```

- spustí animáciu definovanú parametrom *animation*

```
SetHideTime(double time)
```

- umožňuje nastaviť, za akú dlhú dobu sa okno hostujúcej aplikácie skryje

```
Move(Point point)
```

- informuje aplikáciu, kam má presunúť okno, ktoré zobrazuje zvieratko

Ďalej ponúka metódy *Show* a *Hide*, riadiace aplikáciu hostujúcu zvieratko, ktoré zobrazia, respektíve skryjú okno hostujúcej aplikácie.

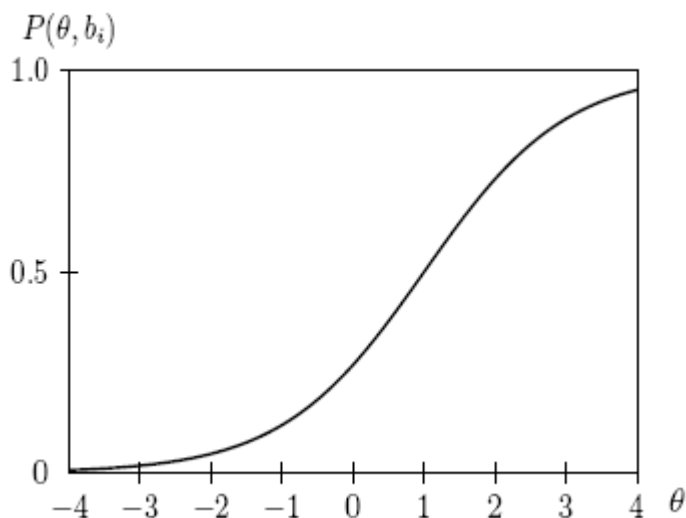
Na strane servera je rozhranie implementované v triede *PetService*. Táto vystavuje udalosti *Say*, *Animate*, *Show*, *SetHideTime*, *Move* a *Hide*, ktorých obslužením reaguje konkrétna aplikácia slúžiaca ako reprezentácia zvieratka na volania z klientskej strany.

9.2 Metódy a prístupy

9.2.1 Item Response Theory

Metóda adaptívneho testovania (CAT) a item response theory (IRT) bola použitá primárne v doméne matematiky, ale návrh umožňuje jej použitie vo všetkých prípadoch, kde sa dá určiť obtiažnosť jednotlivých úloh.

Podstata IRT spočíva v tom, že každá úloha sa dá opísať tzv. charakteristickou krivkou (*Item Characteristic Curve*), ktorá je zobrazená na obrázku 9-7.



Obrázok 9-7. Charakteristická krivka úlohy.

Táto krivka udáva závislosť medzi hráčovými znalosťami θ a pravdepodobnosťou P , že na otázku odpovie správne. Existuje viacero modelov, na určenie tejto krivky pre konkrétnu úlohu, v tomto projekte bol použitý trojparametrový, ktorý berie do úvahy náročnosť úlohy b , diskriminačný faktor a a pravdepodobnosť, že hráč uhádne správnu odpoveď náhodným výberom z možností (vzorec 9-1).

$$P(\theta) = c + \frac{1-c}{1 + e^{-a(\theta-b)}} \quad (9-1)$$

Pravdepodobnosť úspešného vyriešenia úlohy je použitá pri výbere vhodnej otázky, kde sa dá parametrizovať pravdepodobnosť, ktorú bude systém používať ako najvhodnejšiu pre kladenie otázok, pričom štandardne je použitá hodnota 50 percent. Ďalej sa pravdepodobnosť P využíva pre úpravu hráčových znalostí θ pri vrátení odpovede. Tento výpočet sa vykonáva iteratívne podľa vzorca 9-2.

$$\theta_{i+1} = \theta_i + \frac{\sum_{i=1}^N S_i(\theta_N)}{\sum_{i=1}^N I_i(\theta_N)} \quad (9-2)$$

Hodnoty S_i a I_i sa dajú vypočítať podľa vzorcov 9-3 a 9-4.

$$S_i(\theta) = [r_i - P_i(\theta)] \frac{P'_i(\theta)}{P_i(\theta)[1 - P_i(\theta)]} \quad (9-3)$$

Premenná r_i má hodnotu 1, ak bola otázka zodpovedaná správne, v druhom prípade má hodnotu 0.

$$I_i(\theta) = \frac{P'_i(\theta)^2}{P_i(\theta)[1 - P_i(\theta)]} \quad (9-4)$$

Tento model úpravy hráčových znalostí má tú vlastnosť, že v prípade hráčovej správnej odpovede na ťažšiu otázku sa mu hodnota znalostí zvýši viac ako keby odpovedal na ťažšiu. Podobne to platí aj pri nesprávnej odpovedi, kde sa mu táto hodnota naopak znižuje.

Pomocou týchto mechanizmov sa systém dokáže adaptovať na aktuálne schopnosti používateľa ako aj na ich zmeny v procese učenia a vyberať vždy otázky primeranej obtiažnosti.

9.2.2 Metóda častejšieho opakovania nesprávne zodpovedaných otázok

Tento postup je vhodný pre úlohy, kde si dieťa musí správne odpovede namemorovať. Ako príklad bola použitá malá násobilka ale túto metódu je možné aplikovať vo všetkých oblastiach, kde existuje konečný počet otázok.

O každej otázke sa evidujú v modeli používateľa nasledujúce údaje:

- Celkový počet odpovedí

- Počet správnych odpovedí s
- Počet nesprávnych odpovedí f
- Počet správnych odpovedí za sebou, ktorý je nulový, ak bola posledná odpoveď nesprávna n
- Počet nesprávnych odpovedí za sebou, ktorý je nulový, ak bola posledná odpoveď správna m

Pravdepodobnosť výberu sa potom vypočíta podľa vzorca 9-5.

$$P = s - f + 3(m - n) \quad (9-5)$$

Táto metóda zabezpečuje častejší výber tých otázok, ktoré používateľ zodpovedal nesprávne, najmä ak na ne odpovedal nesprávne niekoľko krát za sebou.

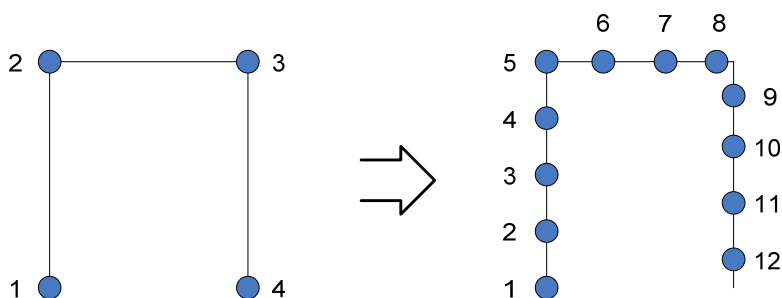
9.2.3 Stroke metrika

Táto metóda je použitá na určenie podobnosti vzorových a hráčových ťahov v hre *Strokes*. Tieto ťahy sú zapísané ako postupnosť úsečiek, ešte presnejšie ako postupnosť ich počiatkových a koncových bodov. Počet bodov označme ako N , potom počet úsečiek bude $N-1$. Pre ohodnotenie podobnosti hráčových ťahov so vzorovými ťahmi sa vykonávajú postupne nasledujúce kroky:

1. Pre každý ťah sa vypočíta dĺžka kroku d (rovnica 9-6). Tá je určená ako podiel dĺžky ťahu a počtu vyhodnocovacích bodov na ťah. Počet vyhodnocovacích bodov k je konštanta, pričom čím je vyššia jej hodnota, tým presnejšie ale aj časovo náročnejšie je vyhodnocovanie. Symboly x^i a y^i predstavujú zložky bodu s príslušným indexom i .

$$d = \frac{\sum_{i=1}^{N-1} \sqrt{(x^{i+1} - x^i)^2 + (y^{i+1} - y^i)^2}}{k} \quad (9-6)$$

2. Nasleduje výpočet vyhodnocovacích bodov na základe dĺžky kroku. Tento krok sa vykonáva kvôli tomu, že dva veľmi podobné ťahy sa môžu skladať z rôzneho počtu rôzne dlhých úsečiek, takže je potrebné vybrať iné body, pomocou ktorých sa bude podobnosť počítať. Zjednodušená ukážka tohto procesu sa nachádza na obrázku 9-8.



Obrázok 9-8. Proces výpočtu vyhodnocovacích bodov.

3. Vzdialenosť vzorového ťahu od hráčovho l sa vypočíta podľa vzorca 9-7, kde M je počet kontrolných bodov, symboly x_v^i a y_v^i predstavujú zložky kontrolných bodov vzorového ťahu a symboly x_h^i a y_h^i zložky bodov hráčovho ťahu s príslušným indexom i .

$$l = \sum_{i=1}^M \sqrt{(x_v^i - x_h^i)^2 + (y_v^i - y_h^i)^2} \quad (9-7)$$

Pretože hráč mohol obťahovať vzorový ťah odzadu, vypočítali sme aj rozdiel ťahov odzadu l' pomocou vzorca 9-8. Ak bola táto hodnota nižšia ako rozdiel l , tak bola použitá namiesto nej.

$$l' = \sum_{i=1}^M \sqrt{(x_v^i - x_h^{M-i+1})^2 + (y_v^i - y_h^{M-i+1})^2} \quad (9-8)$$

4. V poslednom kroku sa vzdialenosť ťahov normalizuje vydelením dĺžkou hráčovho ťahu. Tento krok zabezpečuje vlastnosť, že čím je ťah dlhší, tým je metrika menej prísnejšia a naopak.

Pre túto metriku sa dajú vypracovať viaceré zlepšenia, kedy sa pri cyklických ťahoch, napríklad kruhoch nepočítajú vzdialenosti ťahov len od začiatku a od konca ale pre všetky posunutia a následne sa vyberie najlepší.

9.2.4 Rozoznávanie výšky tónov a opakovanie melódie

V rámci hry *Malý hudobník* je na určenie výšky tónov použitá knižnica FMOD, ktorá realizuje tento proces pomocou spektrálnej analýzy.

Následne sa podobnosť vzorovej a rozpoznanej melódie určuje podľa nasledujúcej metriky:

- za nerovnaký počet tónov sa pre každý pridá k rozdielu penalizácia
- pre všetky príslušne dva tóny zistí poltónová vzdialenosť; jedinou výnimkou sú intervaly e-f a h-c, pretože napriek tomu, že tvoria malé sekundy sa im priraduje cena dvoch poltónov.

Výsledkom metriky je súčet vzdialeností tónov zvýšená o penalizáciu.

9.2.5 Adaptívnosť pri SmartLines

Ak je herná aktivita *SmartLines* spustená z herného klienta, prispôsobuje sa kladenie otázok aktuálnym schopnostiam hráča. Na výber znalostí z problémovej domény sa využíva adaptívny selektor (trieda *AdaptiveSelector*), ktorý v spolupráci s modelom používateľa s využitím Item Response Theory vyberá vhodné príklady na precvičovanie pre dieťa.

Hra *SmartLines* nehodnotí každý pokus o odpoveď zvlášť, ale uzatvára vyhodnocovanie na bloky. Blok je zahájený, keď je hráčovi položená otázka a uzavretý, keď na ňu správne odpovie. Po uzavretí bloku sa vyhodnocuje počet nesprávnych odpovedí predtým, ako bola vybraná správna odpoveď. Metrika berie do úvahy aj celkový počet odpovedí, ktoré otázka obsahuje. Dôležitý je podiel nesprávnych odpovedí na celkový počet, ktorý sa berie ako výsledok metriky.

9.3 Aplikácie

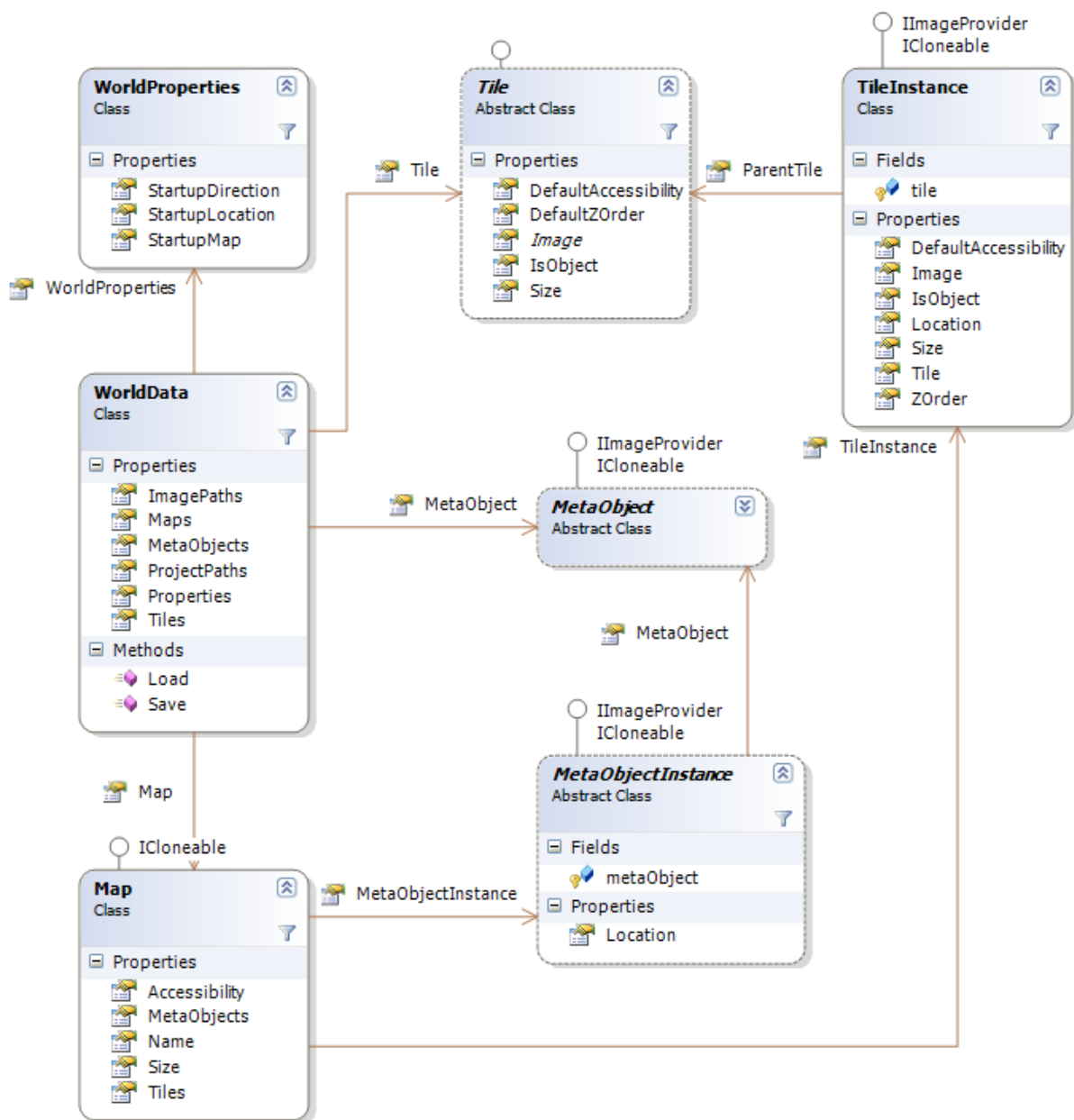
9.3.1 Aplikácia hostujúca služby

Všetky základné služby, okrem úložísk doménových znalostí a doplnkových služieb (ako napríklad služba *StoryTeller*), bežia v jednom hostujúcom procese. Tento poskytuje jednoduché textové používateľské rozhranie a beží ako konzolová aplikácia.

Hlavnou triedou, ktorá riadi spúšťanie a zastavovanie služieb, je trieda *HostManager*. Služby sa spustia zavolaním metódy *StartServices* a vypnú zavolaním metódy *StopServices*. Službám, ktoré implementujú rozhranie *ISupportShutdown*, je pred zastavením zavolaná metóda *Shutdown* a majú možnosť uložiť tak pred vypnutím servera svoj stav, poodpájať pripojených klientov alebo vykonať inú akciu.

9.3.2 Editor máp

Architektúru editora máp možno rozdeliť na dve časti a to na triedy reprezentujúce herný svet a triedy zabezpečujúcu funkcionálnosť editora.

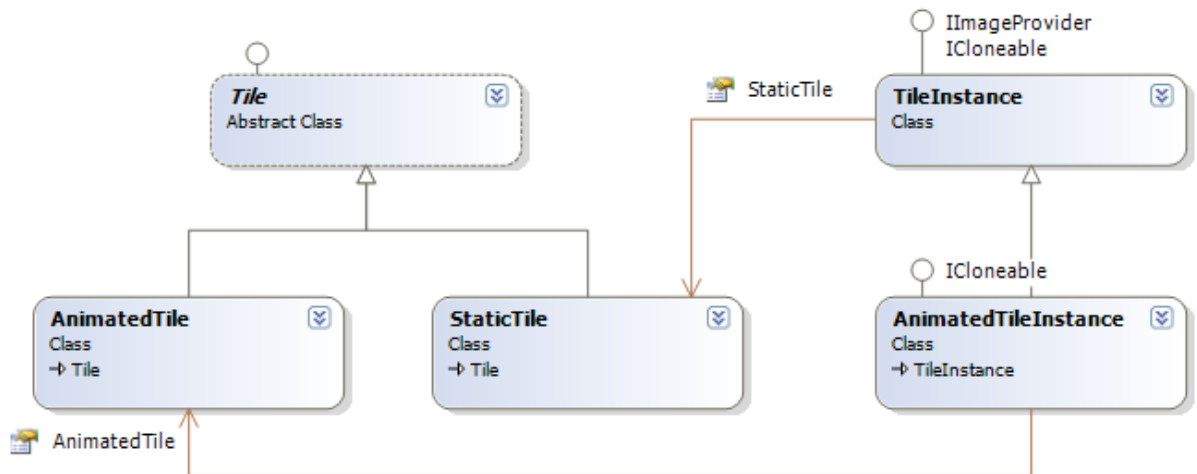


Obrázok 9-9. Triedy reprezentujúce herný svet.

Na obrázku 9-9 sa nachádzajú hlavné triedy reprezentujúce herný svet. Sú to:

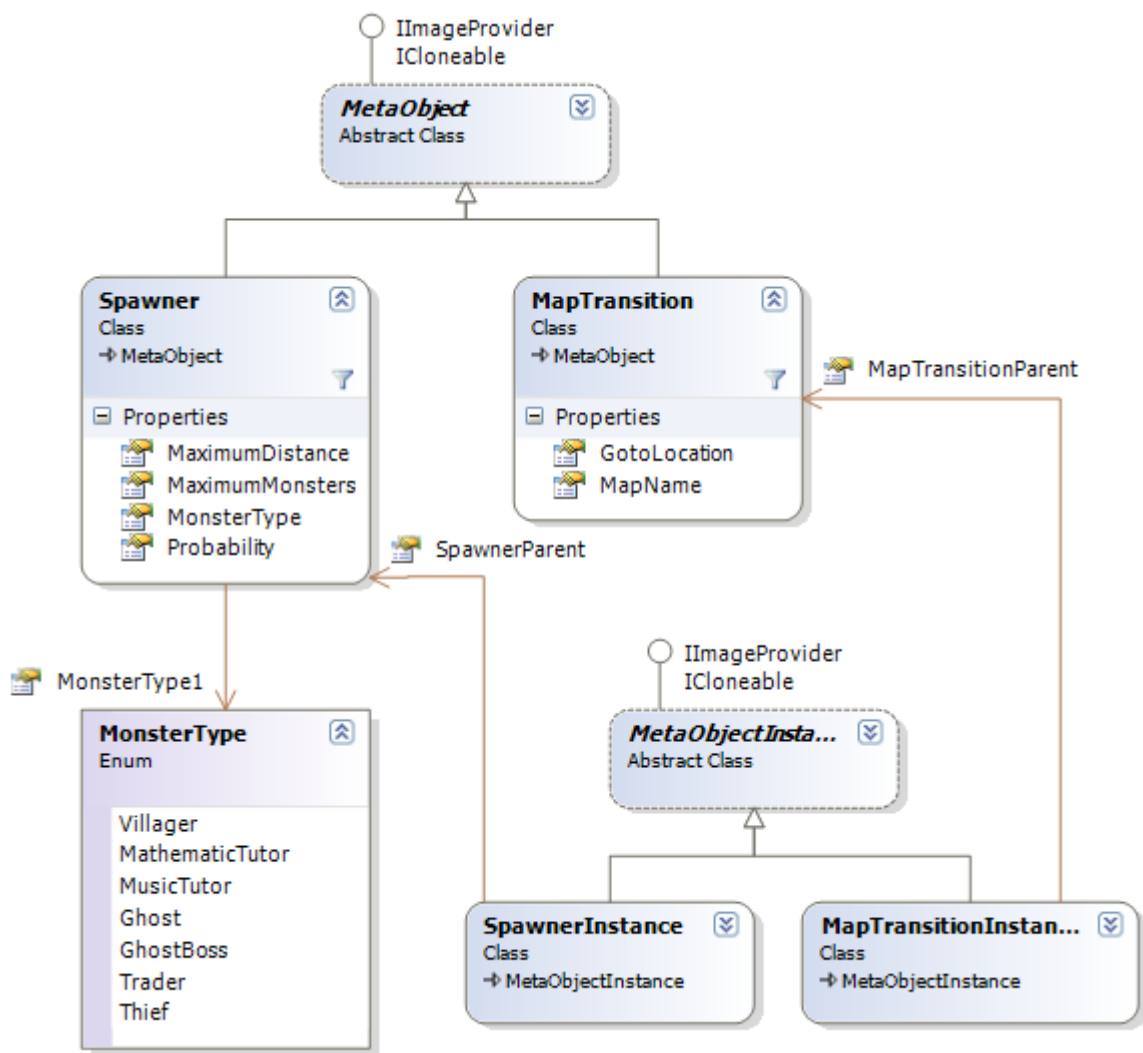
- **WorldData**: Trieda zapúzdrujúca celý herný svet. Obsahuje metódy na načítanie a uloženie do súboru.
- **WorldProperties**: Špecifikuje počiatočnú mapu, kde sa hráč objaví, jeho počiatočnú pozíciu a otočenie.
- **Map**: Reprezentuje hernú oblasť, pre ktorú sú definované tily, priechodnosť a meta objekty.
- **Tile**: Trieda reprezentujúca jedno políčko mapy, pre ktoré určuje jeho obrázok alebo obrázky, poradie vykresľovania, veľkosť, predvolenú prechodnosť a či je to objekt alebo pozadie. Políčka môžu byť statické a animačné a ich konkrétne umiestnenia sú reprezentované triedami s príponou *Instance*. Vzťahy medzi triedami reprezentujúce políčka sú zobrazené na obrázku 9-10.

- *TileInstance*: Predstavuje konkrétne umiestnenie políčka na mape, jeho inštanciu (obr. 9-10).



Obrázok 9-10. Vzťahy medzi triedami reprezentujúcimi políčko.

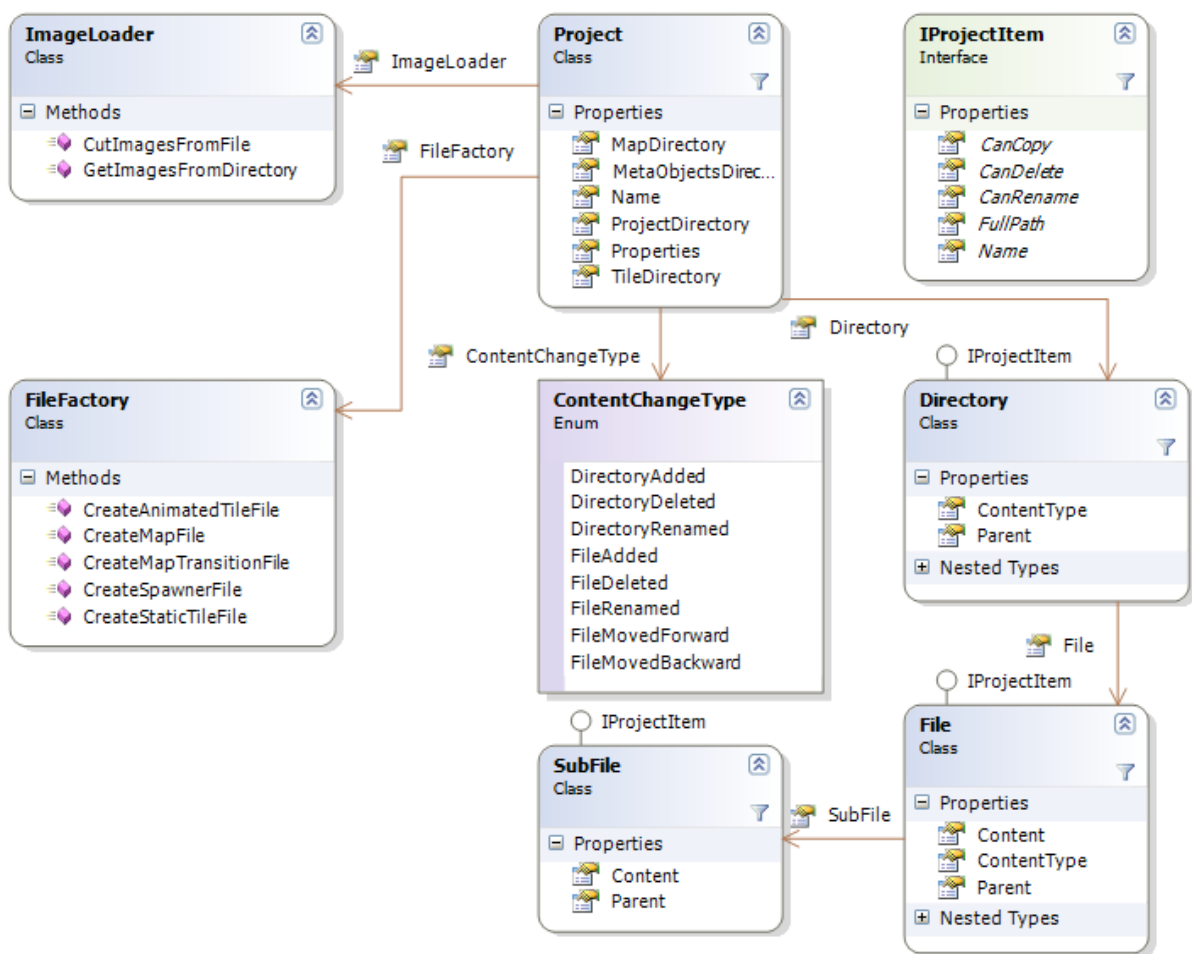
- *MetaObject*: Trieda poskytujúca meta informácie o hernej mape. V hernom svete sa nachádzajú dva druhy takýchto objektov a to *MapTransition*, ktorý popisuje prechod medzi dvoma mapami a *Spawner*, ktorý obsahuje informácie o vytváraní počítačom riadených postáv. Ich konkrétne umiestnenia na mape sú reprezentované triedami s príponou *Instance*. Vzťahy medzi týmito triedami sa nachádzajú na obrázku 9-11.
- *MetaObjectInstance*: Reprezentuje konkrétne umiestnenie meta objektu na mape (obr. 9-11).



Obrázok 9-11. Vzťahy medzi triedami reprezentujúcimi meta objekty.

Druhou časťou architektúry sú triedy zabezpečujúce funkcionality samotného editora. Sú zobrazené na obrázku 9-12. Nasleduje ich krátky popis:

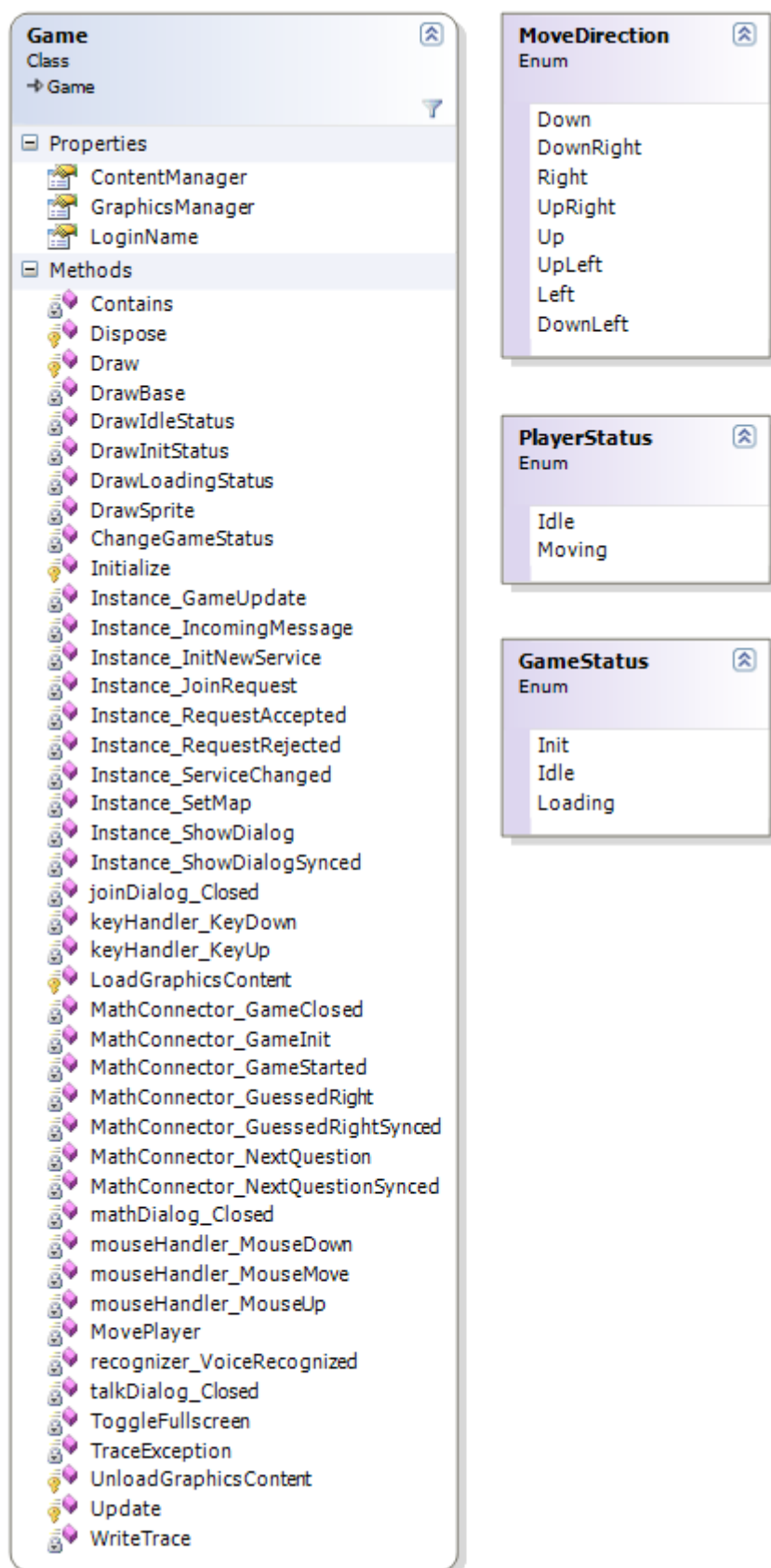
- *Project*: Reprezentuje celú štruktúru projektu, ktorá je reprezentovaná pomocou adresárov a súborov. Pri zmene obsahu vyvoláva udalosti, ktorých charakter je špecifikovaný enumeráciou *ContentChangeType*
- *IProjectItem*: Rozhranie, ktoré implementuje každá trieda nachádzajúca sa v adresárovej štruktúre projektu
- *Directory*: Trieda reprezentujúca adresár, obsahuje ďalšie adresáre alebo súbory, pričom je pre ne možné špecifikovať zoradovanie
- *File*: Reprezentuje súbor projektu. Môže obsahovať rôzne typy údajov napríklad hernú mapu alebo animačné políčko
- *SubFile*: Umožňuje rozčleniť súbor na menšie jednotky s obsahom rôzneho typu
- *ImageLoader*: Obsahuje metódy na načítavanie obrázkov z adresárov a nasekanie obrázku na menšie časti
- *FileFactory*: Umožňuje vytvárať súbory s rôznym obsahom



Obrázok 9-12. Triedy editora máp.

9.3.3 Herný klient

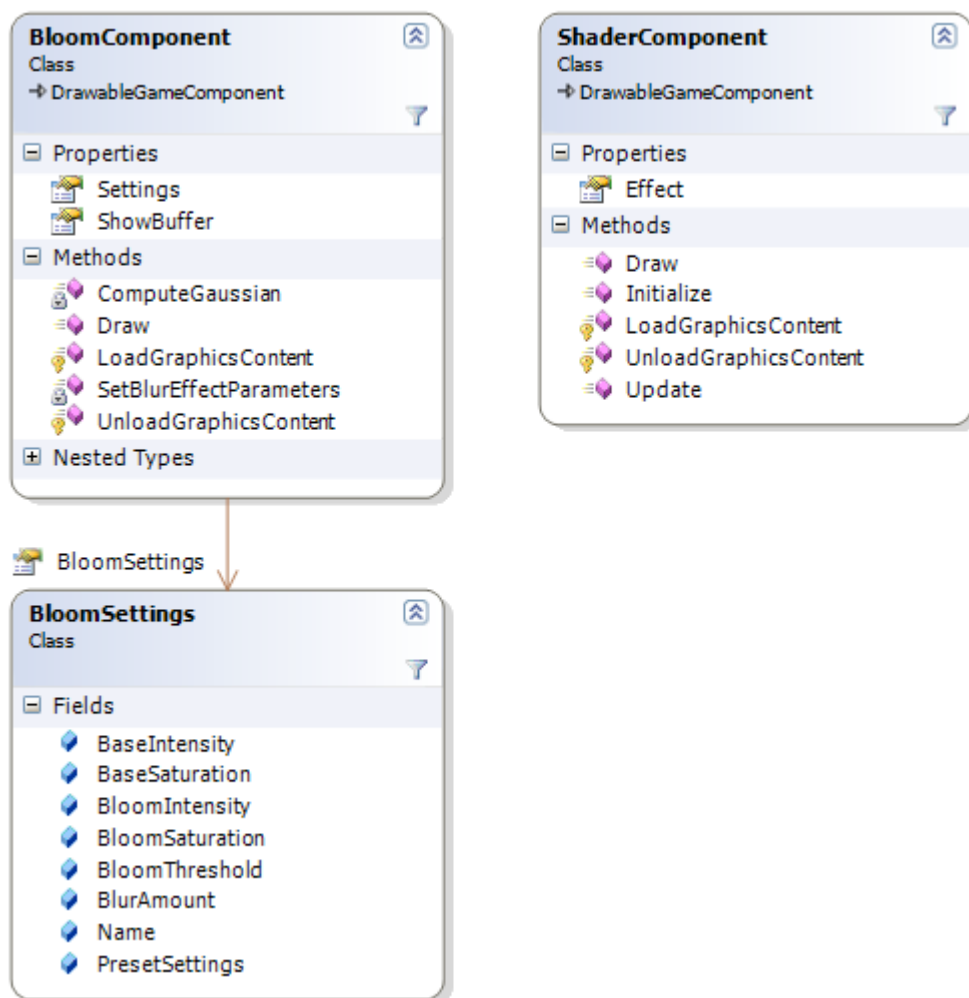
Architektúra herného klienta zahŕňa niekoľko logických častí – triedy súvisiace s hernou logikou a herné komponenty, triedy grafického používateľského rozhrania, triedy pre spracovanie vstupu používateľa a triedy týkajúce sa reprezentácie mapy herného sveta.



Obrázok 9-13. Trieda reprezentujúca logiku herného klienta.

Na obrázku 9-13 sa nachádza diagram triedy reprezentujúcej logiku herného klienta a súvisiace enumerácie. Nasleduje ich zoznam a stručný opis:

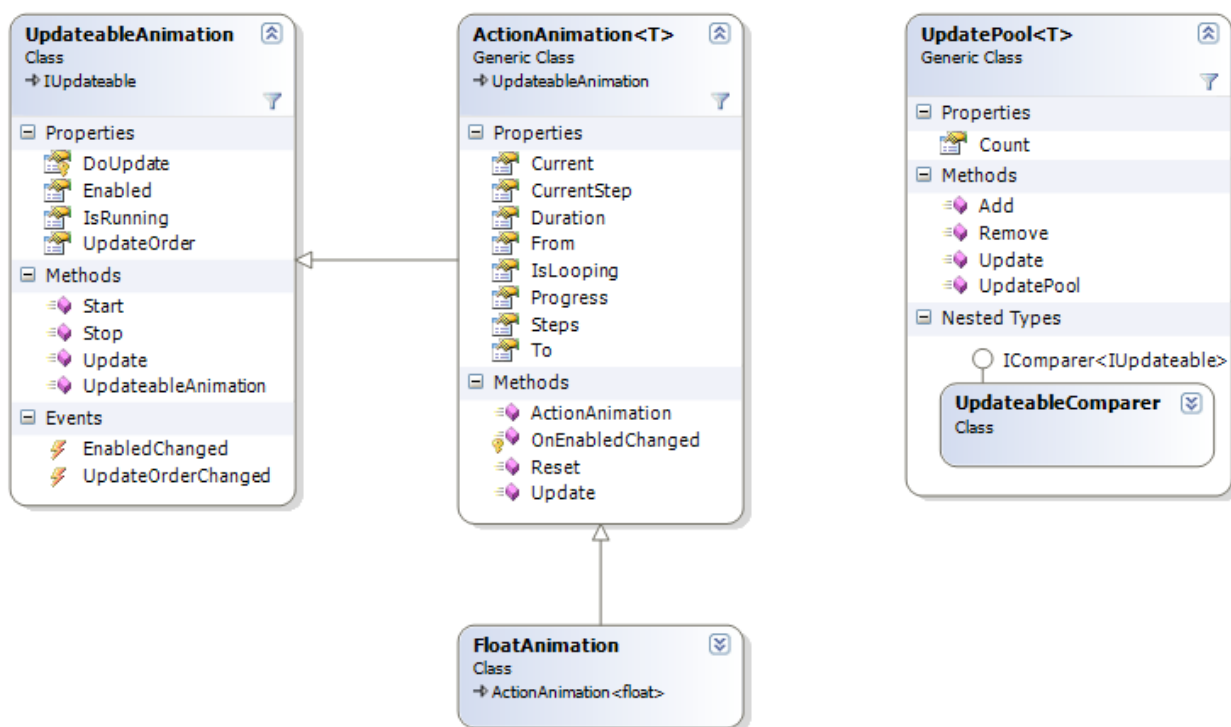
- *Game* – trieda reprezentujúca samotného grafického klienta herného sveta, integruje infraštruktúrne a dátové služby, umožňuje hlasové ovládanie, zodpovedá za kreslenie herného sveta v závislosti od aktuálneho herného stavu a spracúva ovládacie vstupy pre pohyb a interakciu postavy hráča s herným svetom
- *GameStatus* – reprezentuje stav herného klienta
- *PlayerStatus* – reprezentuje stav hráča
- *MoveDirection* – predstavuje smer pohybu hráča



Obrázok 9-14. Prehľad grafických komponentov.

Na obrázku 9-14 sa nachádza diagram tried grafických komponentov. Nasleduje ich zoznam a stručný opis:

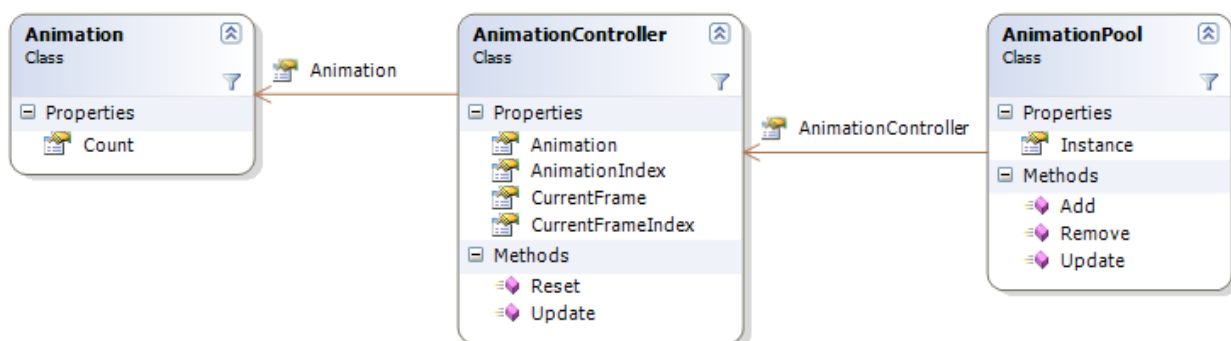
- *BloomComponent* – grafický komponent slúžiaci na spracovanie obrazu medzi nakreslením herného sveta a grafického používateľského rozhrania, ako efekt pre dynamické zvýraznenie kontrastu medzi najtmavším a najsvetlejším bodom v hernej scéne
- *BloomSettings* – nastavenie parametrov efektu zvýraznenia kontrastu
- *ShaderComponent* – komponent umožňujúci kreslenie ľubovoľného grafického efektu



Obrázok 9-15. Triedy akčných animovaných sekvencií.

Na obrázku 9-15 sa nachádza diagram tried akčných animovaných sekvencií. Nasleduje ich zoznam a stručný opis:

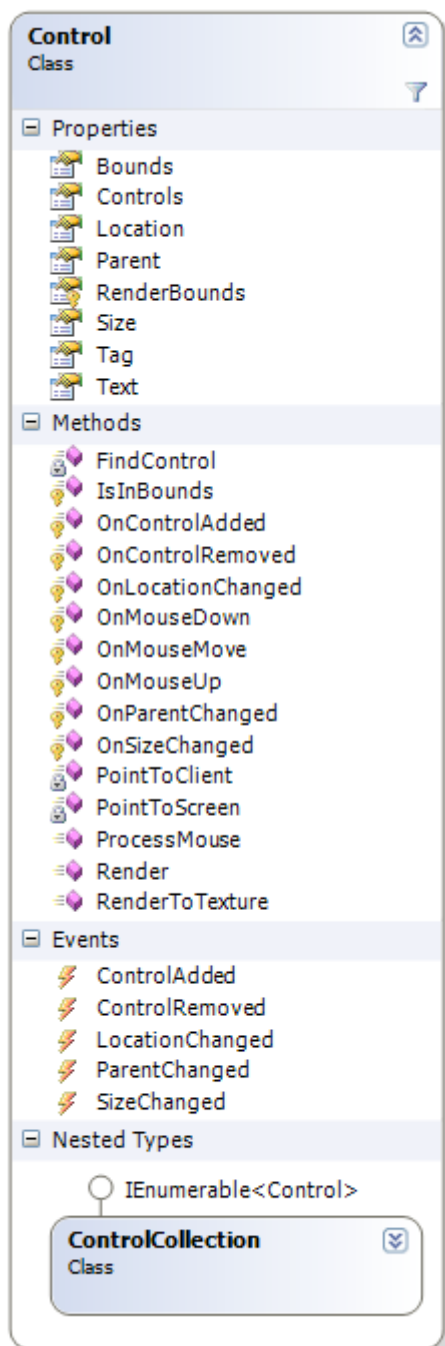
- *UpdateableAnimation* – predstavuje základ pre negrafické animačné sekvencie s možnosťou pozastavenia prehrávania a určenia priority ďalšej aktualizácie vo vzťahu k ostatným animovaným sekvenciám
- *ActionAnimation* – generická trieda reprezentujúca všeobecnú animačnú sekvenciu prechodu z jedného stavu do druhého za požadovaný čas a počet krokov
- *FloatAnimation* – konkrétna akčná sekvenčná animácia pre transformáciu z jedného desatinného čísla na iné
- *UpdatePool* – generická kolekcia spravujúca množinu aktualizovateľných animácií podľa požadovanej priority



Obrázok 9-16. Triedy grafických animovaných sekvencií.

Na obrázku 9-16 sa nachádza diagram tried grafických animovaných sekvencií. Nasleduje ich zoznam a stručný opis:

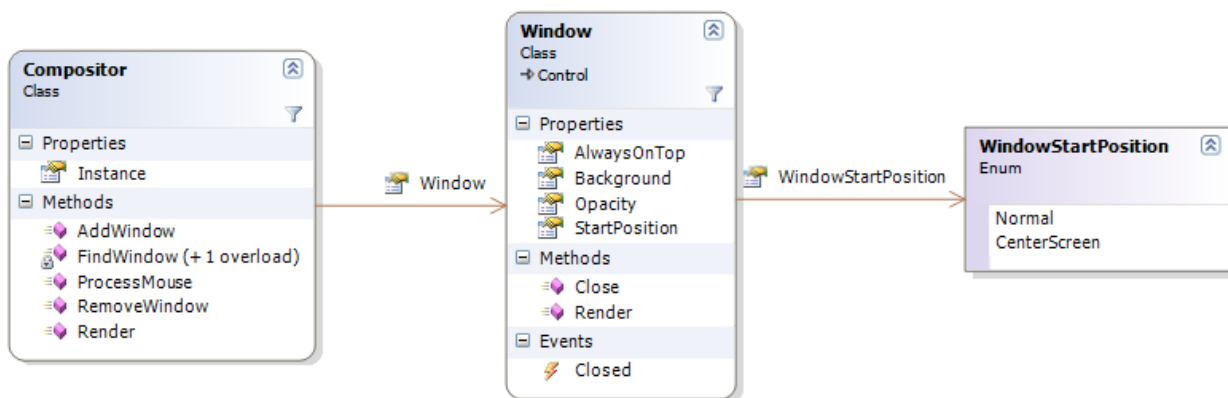
- *Animation* – základná trieda pre grafickú animáciu (sekvenciu textúr)
- *AnimationController* – ovládač konkrétnej grafickej animovanej sekvencie
- *AnimationPool* – kolekcia určená pre synchronizovanú aktualizáciu viacerých animačných ovládačov



Obrázok 9-17. Základná trieda pre všetky ovládacie prvky grafického používateľského rozhrania.

Na obrázku 9-17 sa nachádza diagram základnej triedy pre všetky ovládacie prvky grafického používateľského rozhrania. Nasleduje jej stručný opis:

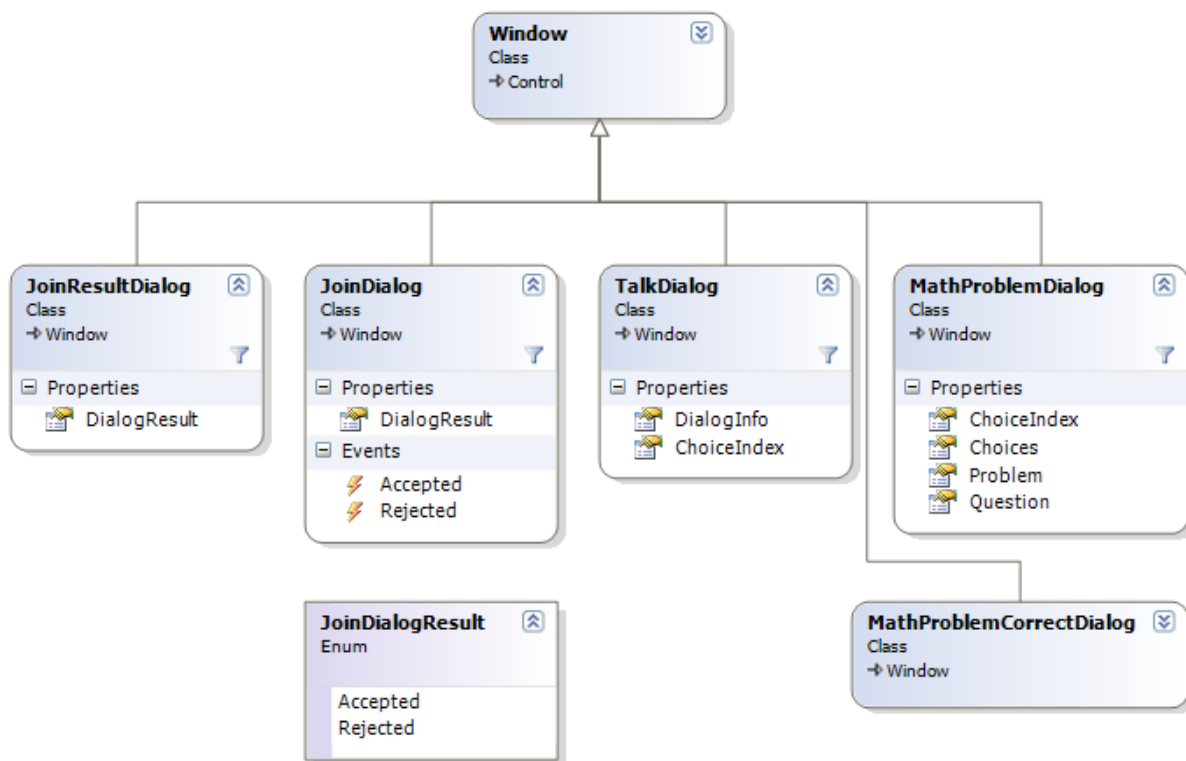
- *Control* – základná trieda pre všetky ovládacie prvky grafického používateľského rozhrania, ktorá nesie informácie o svojej aktuálnej polohe a veľkosti, rodičovi a svojich deťoch a reaguje na udalosti na základe vstupu a deleguje vstup na vnorených potomkov; vie ako sa má kresliť na obrazovku alebo do textúry



Obrázok 9-18. Triedy súvisiace s kompozíciou grafického používateľského rozhrania.

Na obrázku **Error! Reference source not found.** sa nachádza diagram tried súvisiacich s kompozíciou grafického používateľského rozhrania. Nasleduje ich zoznam a stručný opis:

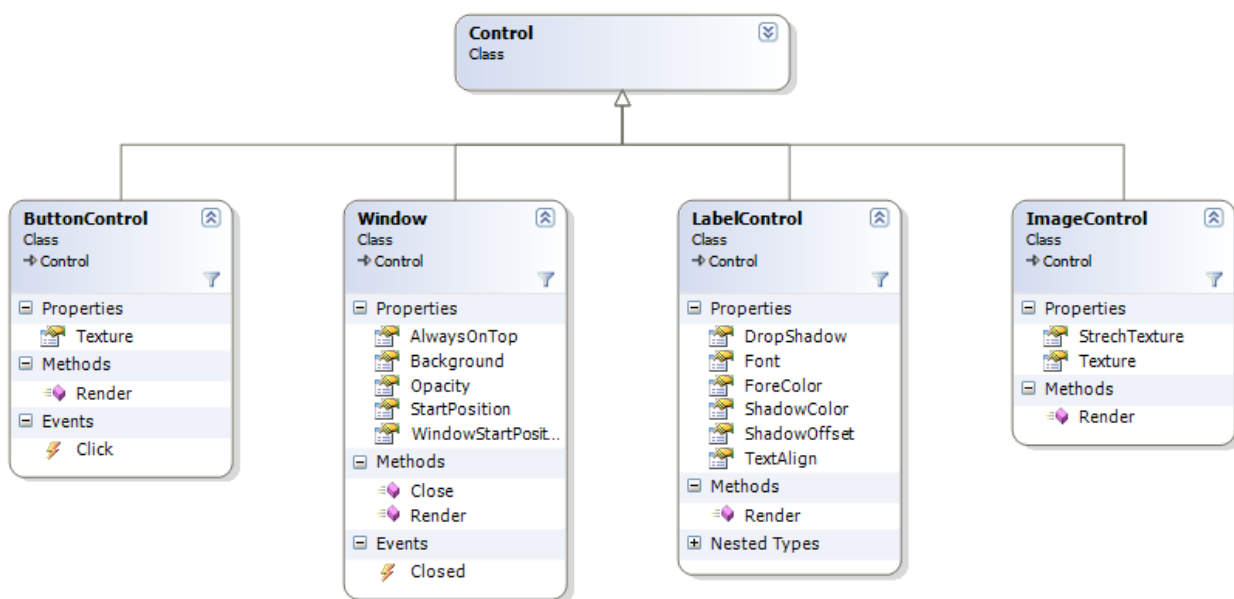
- *Compositor* – trieda slúžiaca na kompozíciu okien (*Window*) grafického používateľského rozhrania a jeho kreslenie; spracúva všetok vstup potrebný pre interakciu s používateľským rozhraním a deleguje udalosti na aktívne spravované okná
- *Window* – predstavuje okno grafického používateľského rozhrania s podporou pre transparentiu a možnosťou precedencie grafického zobrazenia pred ostatnými oknami
- *WindowStartPosition* – enumerácia určujúca počiatočné umiestnenie okna na obrazovke



Obrázok 9-19. Herné dialógové okná.

Na obrázku 9-19 sa nachádza diagram tried herných dialógových okien. Nasleduje ich zoznam a stručný opis:

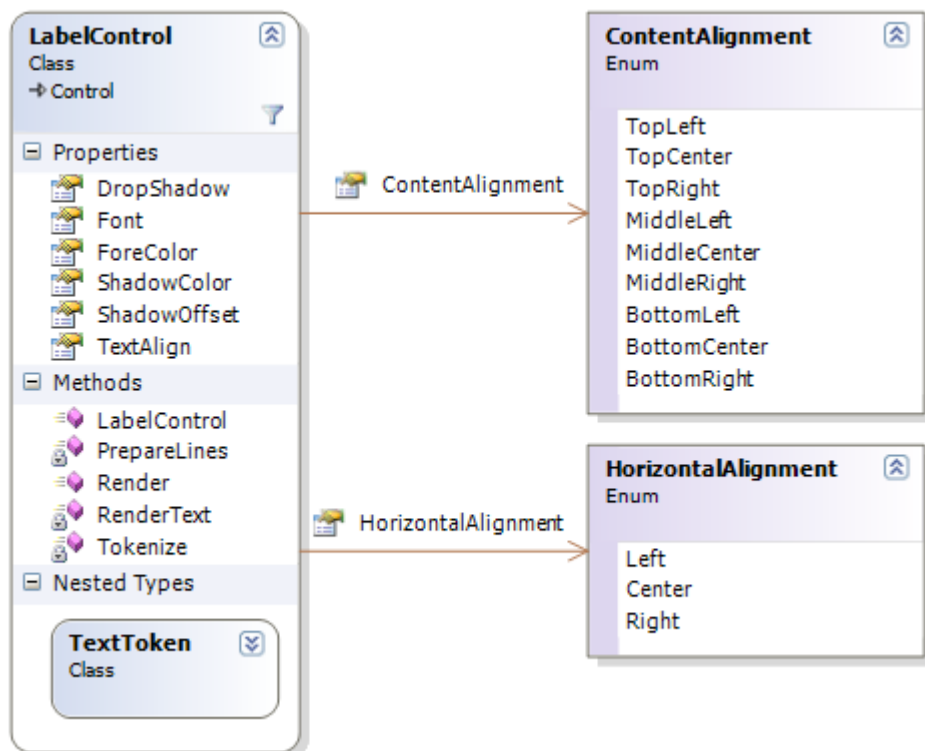
- *JoinResultDialog* – dialógové okno zobrazujúce odpoveď hráča na požiadavku pre vstup do tímu
- *JoinDialog* – dialógové okno s požiadavkou hráča pre vstup do tímu
- *MathProblemDialog* – dialógové okno zobrazujúce matematický problém v rámci hry matematika v tíme
- *MathProblemCorrectDialog* – dialógové okno indikujúce správnosť zodpovedaného matematického príkladu
- *TalkDialog* – dialog slúžiaci na komunikáciu a interakciu s postavami v hernom svete, umožňujúci čítanie vypovedanej repliky a na ňu ponúknutých odpovedí



Obrázok 9-20. Prehľad tried ovládacích prvkov grafického používateľského rozhrania.

Na obrázku 9-20 sa nachádza diagram tried ovládacích prvkov grafického používateľského rozhrania. Nasleduje ich zoznam a stručný opis:

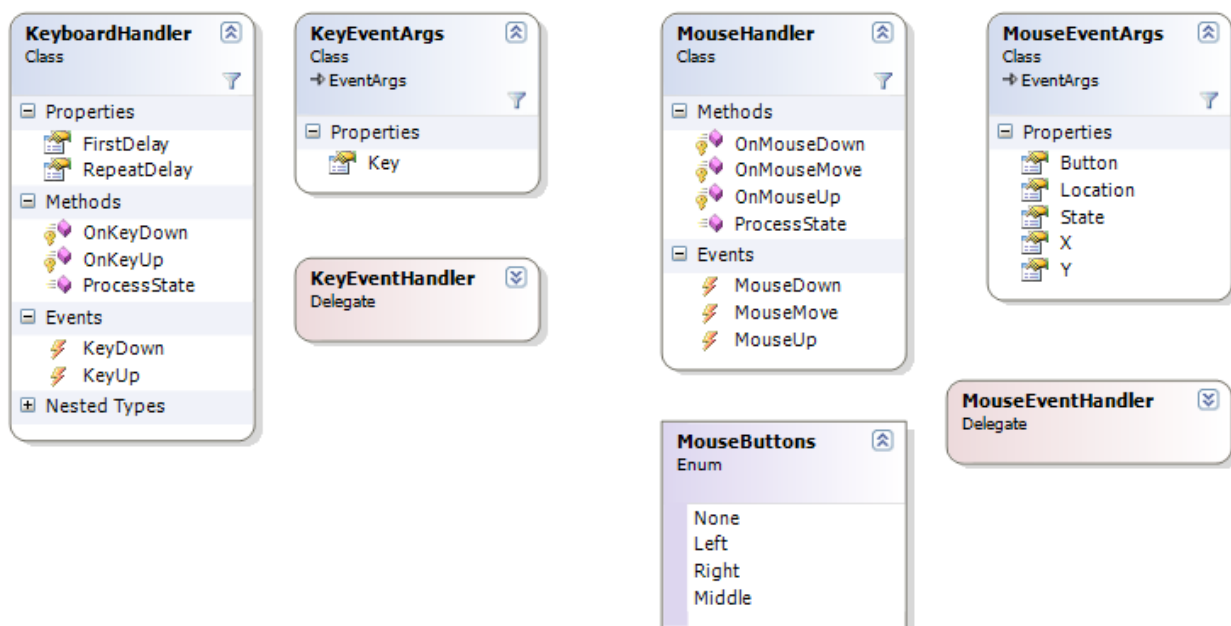
- *ButtonControl* – ovládací prvok grafického používateľského rozhrania typu tlačidlo
- *Window* – kontajnerový ovládací prvok určený pre výslednú kompozíciu na obrazovke
- *LabelControl* – statický textový prvok
- *ImageControl* – prvok slúžiaci na zobrazenie obrázku



Obrázok 9-21. Statický textový prvok.

Na obrázku 9-21 sa nachádza diagram triedy statického textového prvku a súvisiacich enumerácií. Nasleduje ich zoznam a stručný opis:

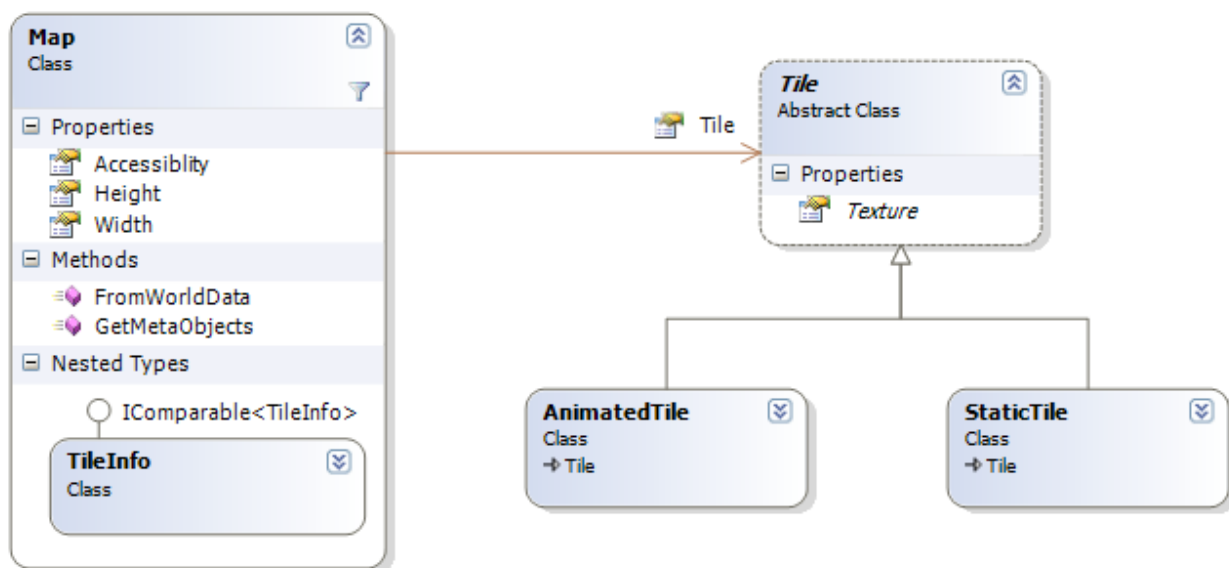
- *LabelControl* – statický textový prvok s možnosťou určenia farby, fontu a zalomenia textu a schopnosťou zobrazenia tieňu pod textom s určenou farbou a polohou voči textu
- *ContentAlignment* – reprezentuje polohu obsahu vzhľadom na vymedzenú oblasť
- *HorizontalAlignment* – reprezentuje horizontálnu polohu obsahu



Obrázok 9-22. Prehľad tried na spracovanie vstupu používateľa.

Na obrázku 9-22 sa nachádza diagram tried na spracovanie vstupu používateľa. Nasleduje ich zoznam a stručný opis:

- *KeyboardHandler* – slúži na spracovanie vstupu z klávesnice a oznamuje vo forme udalostí stlačenie a pustenie klávesy
- *KeyEventArgs* – nesie informáciu o udalosti z klávesnice
- *KeyEventHandler* – delegát slúžiaci na spracovanie udalosti z klávesnice
- *MouseHandler* – slúži na spracovanie vstupu z myši, ktoré oznamuje vo forme udalosti stlačenia alebo pustenia tlačidla, prípadne pohybu myši po obrazovke
- *MouseEventArgs* – nesie informáciu o udalosti z myši, obsahuje stlačené tlačidlá a aktuálnu polohu kurzoru myši na obrazovke
- *MouseEventHandler* – delegát slúžiaci na spracovanie udalosti z myši
- *MouseButton* – predstavuje jedno alebo kombináciu tlačidiel na myši



Obrázok 9-23. Triedy reprezentujúce herný svet.

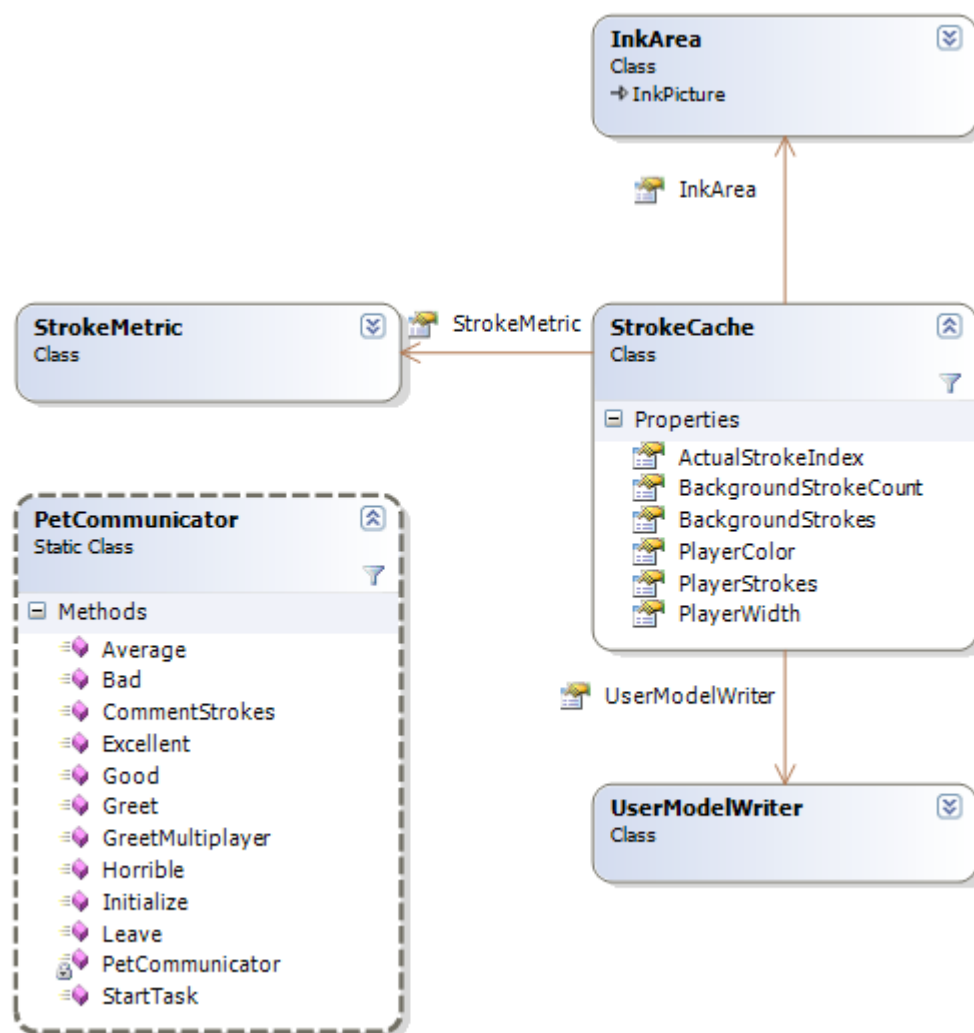
Na obrázku 9-23 sa nachádza diagram tried reprezentujúcich herný svet. Nasleduje ich zoznam a stručný opis:

- *Map* – reprezentuje mapu herného sveta s informáciou o jej výške, šírke a priechodnosti, načítaných z dát exportovaných editorom máp herného sveta
- *TileInfo* – nesie informáciu o polohe a topografickej výške políčka herného sveta, pre potreby spracovania načítanej mapy
- *Tile* – slúži ako základ pre grafickú reprezentáciu políčka herného sveta
- *AnimatedTile* – predstavuje animované políčko herného sveta
- *StaticTile* – predstavuje statické políčko herného sveta

9.3.4 Hra Strokes

Na obrázku 9-24 sa nachádza diagram najdôležitejších tried hry *Strokes*. Nasleduje ich stručný popis:

- *StrokeCache*: Slúži na uchovávanie aktuálnej úlohy a ťahov, ktoré hráč zatiaľ nakreslil. Ďalej uchováva farbu a hrúbku hráčovho pera. Pri hre viacerých hráčov sú v nej uložené aj ťahy súperov
- *InkArea*: Prvok používateľského rozhrania zobrazujúci obrázok, vzorové ťahy a ťahy hráčov. Umožňuje kreslenie na svoj povrch perom zvolenej hrúbky a farby pomocou myši alebo tabletu
- *StrokeMetric*: Obsahuje metódy pre výpočet podobnosti dvoch ťahov (viď kapitolu 9.2.3)
- *UserModelWriter*: Zapisuje nové hodnoty zistených používateľových schopností a výkonov do modelu používateľa
- *PetCommunicator*: Komunikuje so službou *Zvieratko* a zabezpečuje jeho vhodnú reakciu na hráčove vstupy ako napríklad vyslovenie pochvaly za presne obkreslený ťah

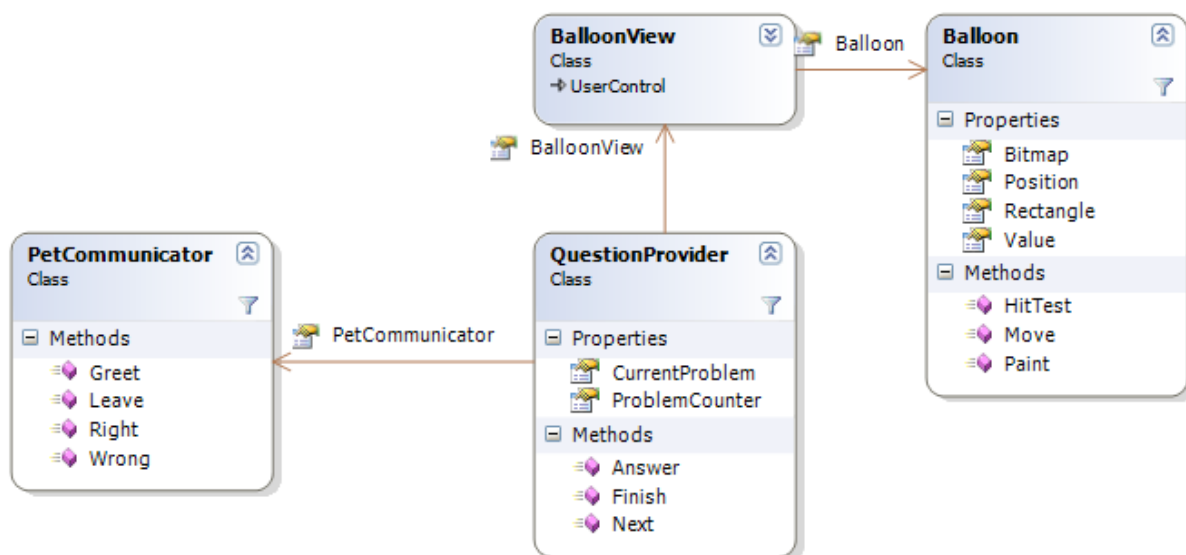


Obrázok 9-24. Podrobný návrh hry Strokes.

9.3.5 Hra Matematické balóny

Na obrázku 9-25 sú zobrazené triedy použité v hre Matematické balóny. Nasleduje ich zoznam a krátky popis:

- *QuestionProvider*: Umožňuje výber a zodpovedanie položenej otázky, eviduje aktuálnu otázku a počet doteraz zodpovedaných otázok
- *BalloonView*: Prvok používateľského rozhrania zobrazujúci balóny so správnymi a nesprávnymi odpoveďami k aktuálnej otázke
- *Balloon*: Trieda zodpovedaná za vykreslenie konkrétneho balóna. Obsahuje metódu na detekciu, či naň používateľ klikol
- *PetCommunicator*: Posiela služby *Pet* notifikácie pri rôznych herných udalostiach



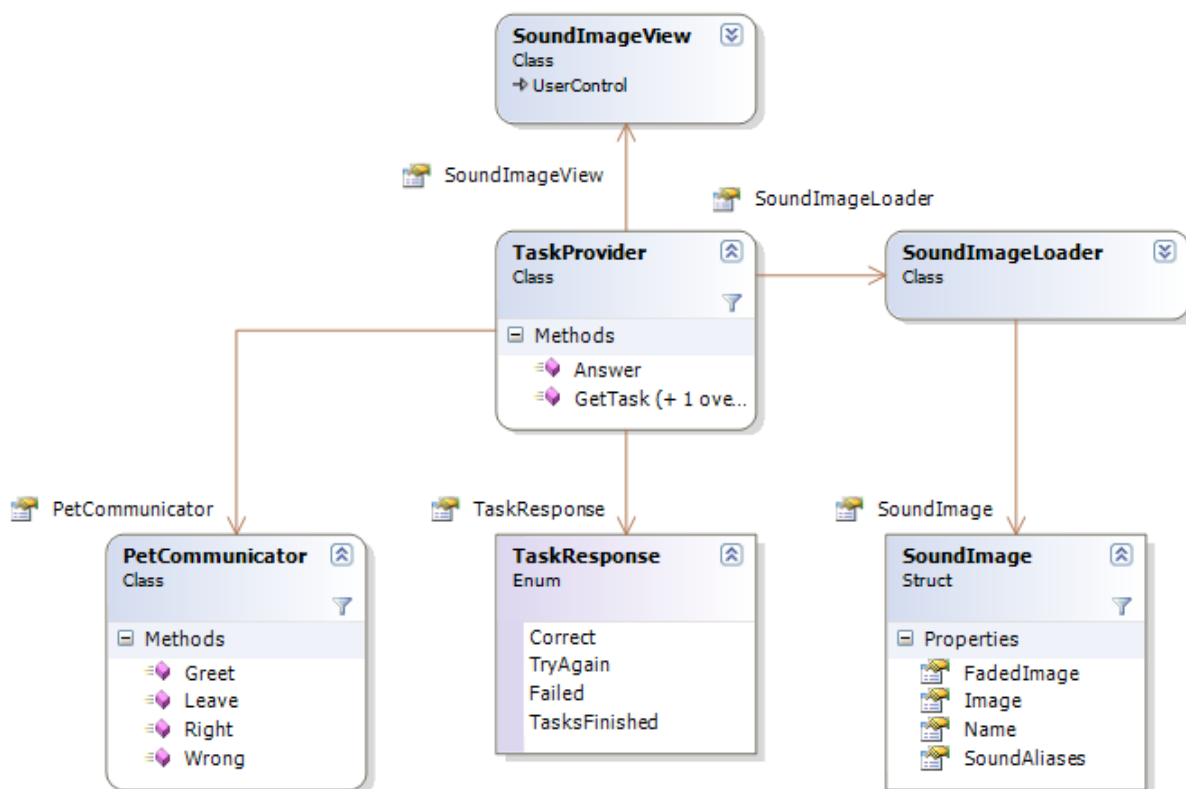
Obrázok 9-25. Podrobný návrh hry Matematické bolóny.

9.3.6 Hra Zvuky

Hru je možno spustiť v dvoch módoch. Prvý je prehliadací, kde sa používateľovi po kliknutí zahrá príslušný zvuk. Ďalším je režim skúšania, pri ktorom sa dá parametrami určiť počet otázok a pokusov správne odpovedať.

Na obrázku 9-26 sa nachádza diagram tried hry Zvuky. Nasleduje ich stručný opis:

- *TaskProvider*: Zodpovedá za výber a zodpovedanie otázky. Na zodpovedanie je možné použiť viacero pokusov, čo je zohľadnené použitím enumerácie *TaskResponse*
- *SoundImageLoader*: Načítava obrázky a zvuky zo zadaných adresárov
- *SoundImage*: Obsahuje obrázok a jemu priradený zvuk
- *SoundImageView*: Prvok používateľského rozhrania, ktorý zobrazuje obrázky a po kliknutí prehrá príslušný zvuk.
- *PetCommunicator*: Komunikuje so službou *Pet* a zabezpečuje vhodnú reakciu na hráčove vstupy

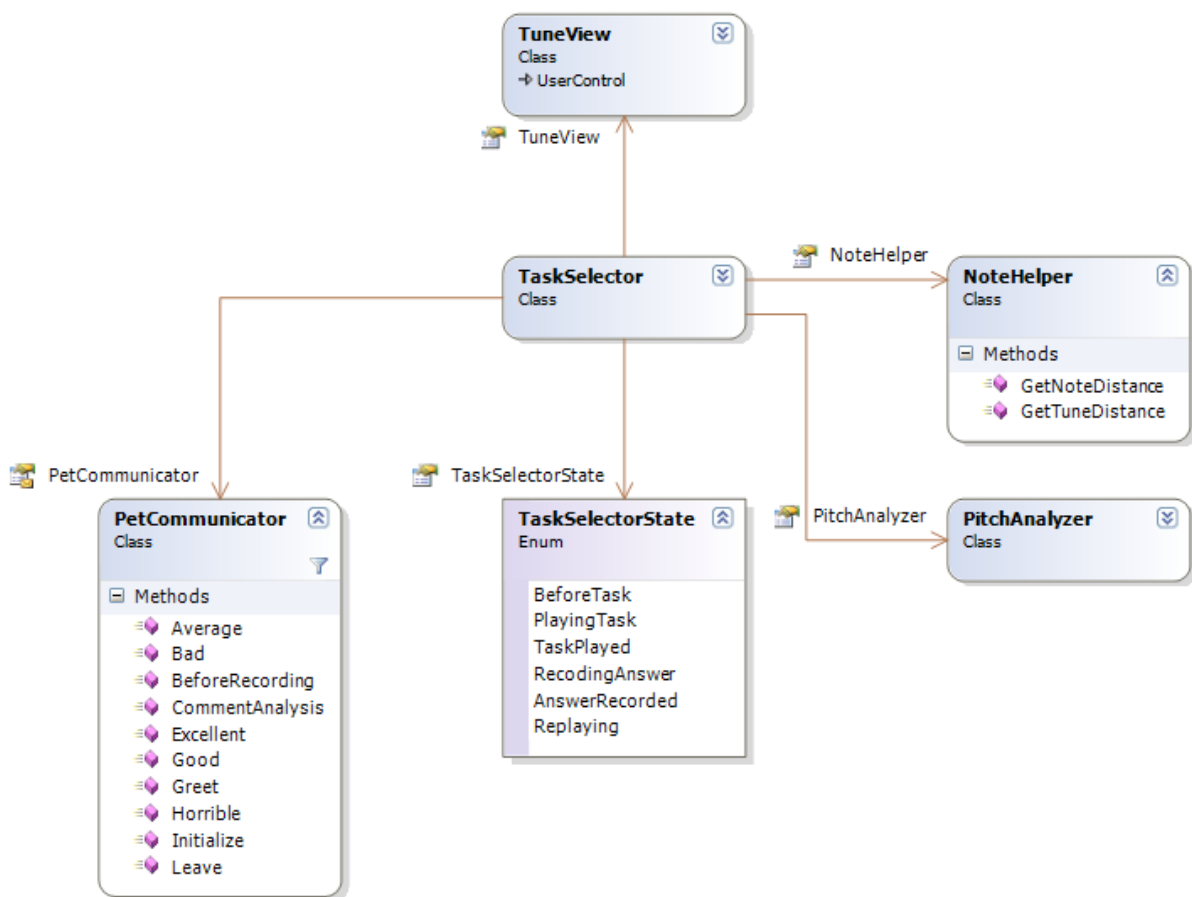


Obrázok 9-26. Diagram tried jadra hry Zvuky.

9.3.7 Hra Opakovanie melódie

Obrázok 9-27 zobrazuje podrobný návrh hry Opakovanie melódie. Nasleduje opis najdôležitejších tried:

- *TaskSelector*: Zabezpečuje načítanie a výber úloh ako aj vyhodnotenie odpovede. Môže sa nachádzať v stavoch definovaných enumeráciou *TaskSelectorState*
- *NoteHelper*: Nachádzajú sa v nej metódy na určenie vzdialenosti tónov a melódii
- *PitchAnalyser*: Umožňuje analyzovať prehrávaný alebo práve nahrávaný zvuk použitím knižnice FMOD
- *TuneView*: Prvok používateľského rozhrania, ktorý zobrazuje notovú osnovu a noty vzorovej a hráčom zopakovanej melódie, pričom farebne odlišuje zhodné a rôzne noty.
- *PetCommunicator*. Zabezpečuje komunikáciu so službou *Pet*

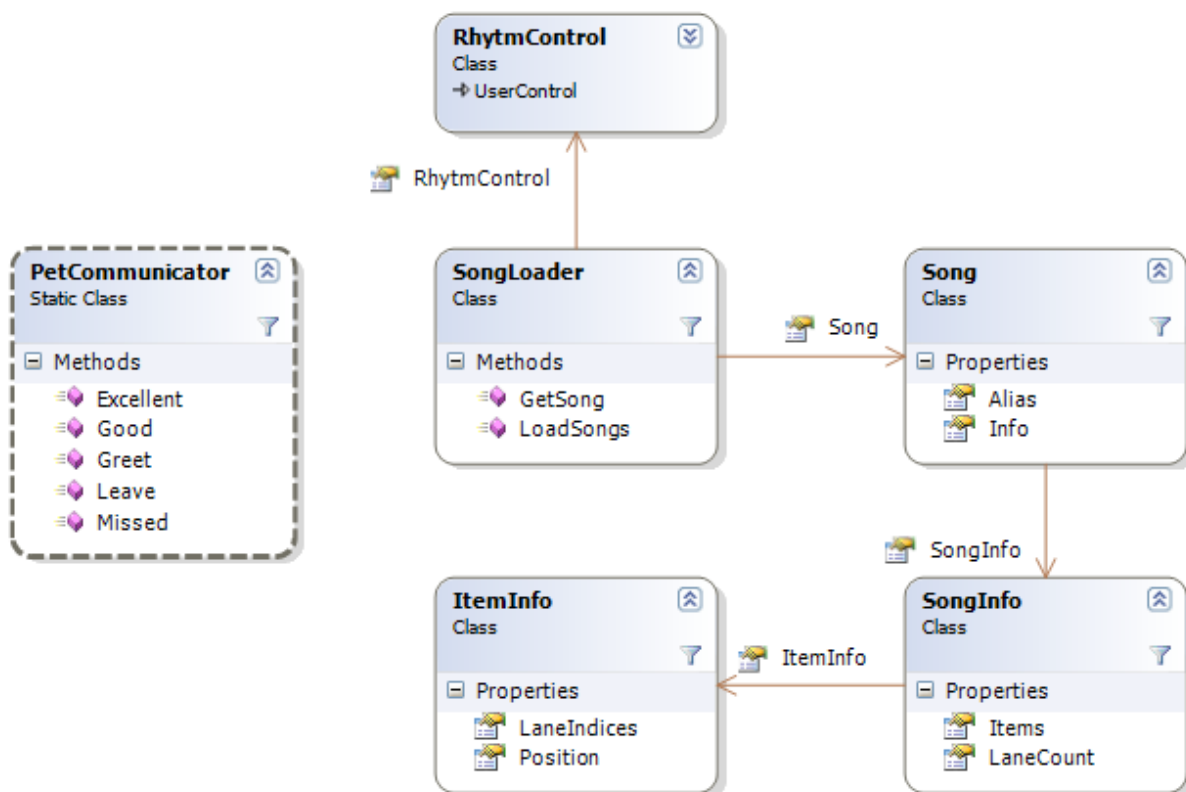


Obrázok 9-27. Diagram tried jadra hry Opakovanie melódie.

9.3.8 Hra Malý hudobník

Na obrázku 9-28 sa nachádzajú najdôležitejšie triedy, z ktorých je hra zložená. Sú to:

- *SongLoader*: Zabezpečuje načítanie a výber pesničiek spolu s ich anotáciami
- *Song*, *SongInfo* a *ItemInfo*: Triedy popisujúce pozície a časové okamihy, pri ktorých sa majú noty v hre objavovať
- *PetCommunicator*: Trieda zabezpečujúca komunikáciu so službou *Pet*



Obrázok 9-28. Podrobný návrh hry Malý hudobník

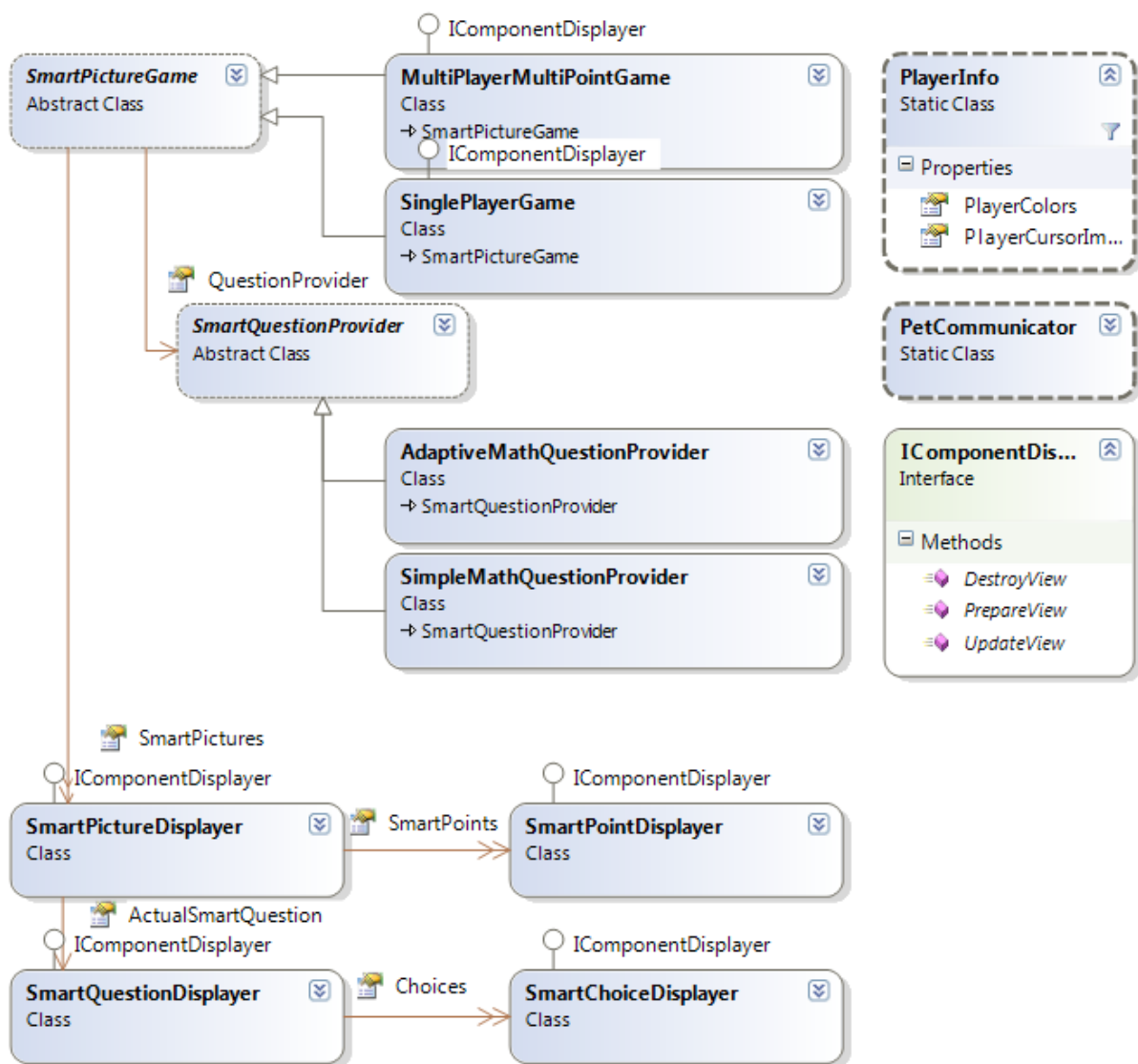
9.3.9 Hra SmartLines

Hra môže bežať v dvoch režimoch: Režim pre jedného hráča, ktorý sa ďalej prispôbuje v závislosti od miesta volania hry (z herného sveta, zo sidebaru). V tomto režime hra umožňuje ovládať používateľské rozhranie hlasom. Druhý mód je hra pre viacerých hráčov, kedy sa využívajú služby knižnice Microsoft Multipoint, využitie ktorej umožňuje ovládanie viacerými myšami.

Na obrázku 9-29 sú zobrazené hlavné triedy aplikácie SmartLines. Najdôležitejšie triedy sú:

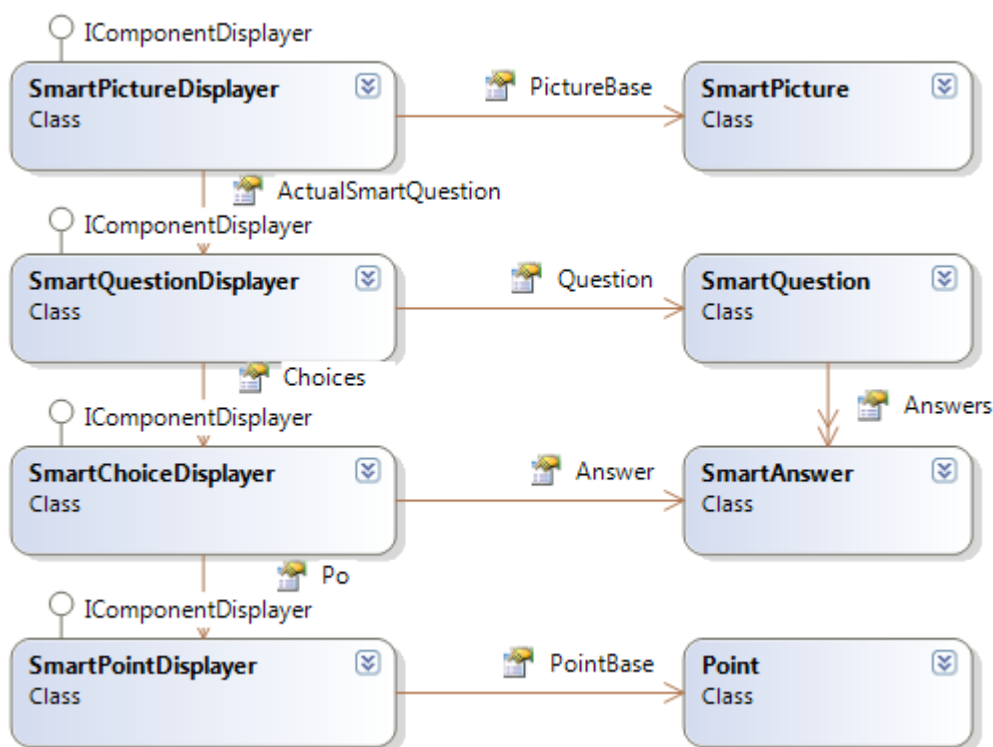
- *SmartPictureGame*: Základná abstraktná trieda aplikácie, ktorá v sebe zahŕňa všetky dáta súvisiace so zobrazením a dátami pre zobrazenie, ktoré nie sú závislé od režimu hry. Obsahuje abstraktné metódy na spustenie, a ukončenie hernej inštancie, ktoré sú ďalej implementované v dcérskych triedach
- *SinglePlayerGame*: Inštancia hry, ktorá implementuje abstraktné metódy bázevej triedy pre jedného hráča a rozširuje správanie o ovládanie hlasom
- *MultiPlayerMultiPointGame*: Inštancia hry, ktorá je prispôbená pre hru viacerých hráčov. Obsahuje triedu MultiMouseClassWrapper, ktorá obaluje a rozširuje triedu MultiMouseClass, poskytujúcu rozhranie pre prácu s niekoľkými myšami
- *SmartPictureDisplayer*: Trieda, ktorá zabezpečuje komplexnú interakciu medzi jediným hráčom a hrou. Nie je závislá od režimu hry. Obsahuje v sebe logiku hry, generovanie a vykresľovanie čiary, vytváranie rozhrania pre kladenie otázok hráčovi. Deleguje spracovanie na nižšie vrstvy. Údaje pre zobrazovanie jej poskytuje trieda SmartPicture
- *SmartPointDisplayer*: Trieda, ktorá zabezpečuje interakciu medzi hráčom a jediným bodom v čiare, resp. jediným falošným bodom (Bod nesprávnej odpovede vygenerovaný v rámci SmartQuestionInstance)

- *SmartQuestionDisplayer*: Trieda zabezpečujúca, generovanie a zobrazovanie otázky používateľovi, vytváranie falošných bodov a interakciu s nimi
- *SmartQuestionProvider*: Trieda zabezpečujúca základné rozhranie pre poskytovanie otázok pre ostatné triedy aplikácie. Funguje na princípe Pull modelu. A poskytuje možnosti cacheovania otázok, za účelom nižšieho zaťaženia siete a zvýšenia výkonu
- *SimpleMathQuestionProvider*: Poskytovateľ matematických otázok, ktorý nezohľadňuje informácie z profilu hráča
- *AdaptiveMathQuestionProvider*: Poskytovateľ otázok, ktorý zohľadňuje informácie z profilu hráča, a na základe týchto informácií vyberá otázky vo vhodnej náročnosti
- *SmartChoiceDisplayer*: Trieda, ktorá zabezpečuje zobrazovanie jednej z odpovedí otázku. Jej implementácia umožňuje na základe typu odpovede zobrazit' ako textové, tak aj multimediálne informácie
- *PlayerInfo*: Statická trieda obsahujúca informácie o hráčoch, ktoré sa používajú najmä pre účely jednotného zobrazovania
- *PetCommunicator*: Statická trieda určená na sprostredkovanie komunikácie so Zvieratkom. Sú v nej centralizované reakcie zvieratka na jednotlivé herné situácie.
- *Rozhranie IComponentDisplayer*: Rozhranie, ktoré poskytuje základné metódy slúžiace na zjednotenie zobrazovania jednotlivých inštancií v kompozitnom zobrazovacom modeli



Obrázok 9-29. Podrobný návrh hry SmartLines.

Architektúra aplikácie je navrhnutá tak, aby ju bolo možné neskôr rozšíriť pridaním nových herných módov. Dáta aplikácie sú preto oddelené od ich zobrazovania. Každá Trieda s postfixom „Displayer“ má priradenú zodpovedajúcu triedu bez postfixu, ktorá predstavuje údaje získané prostredníctvom dátových služieb (*SmartQuestionProvider*, *SmartPictureStorage*). Údaje môžu byť spoločné pre viacerých hráčov (Duel), alebo každý hráč hrajúci tú istú hru na tom istom počítači môže mať vlastné údaje (Tréningový režim) závisiace od jeho aktuálnych schopností.

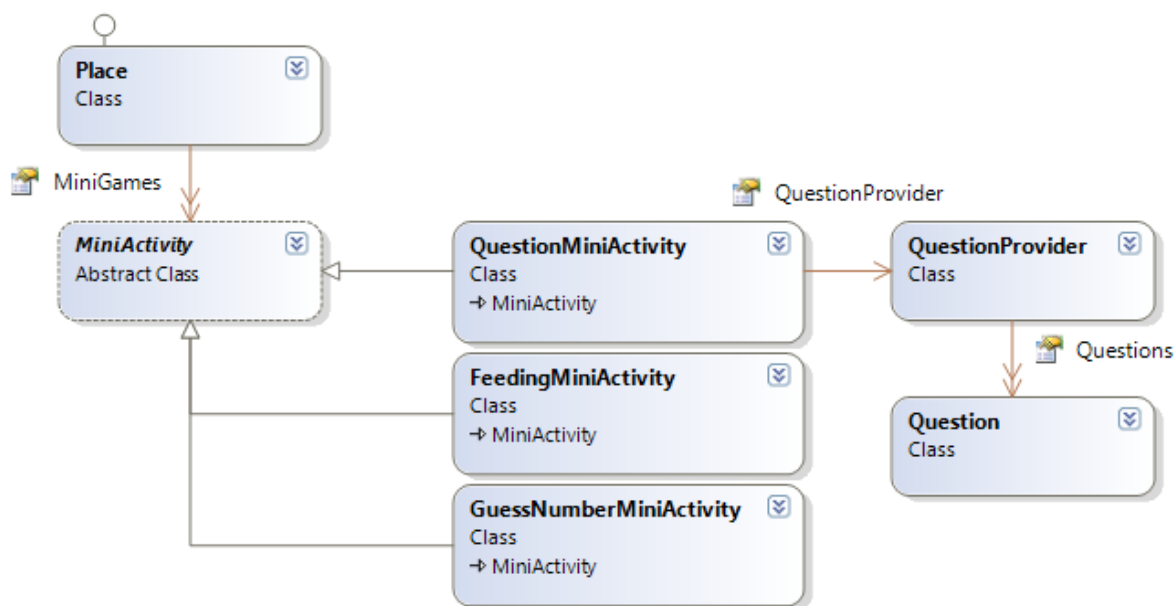


Obrázok 9-30. Prepojenie hernej logiky s dátovou vrstvou.

9.3.10 Zvieratko v PDA

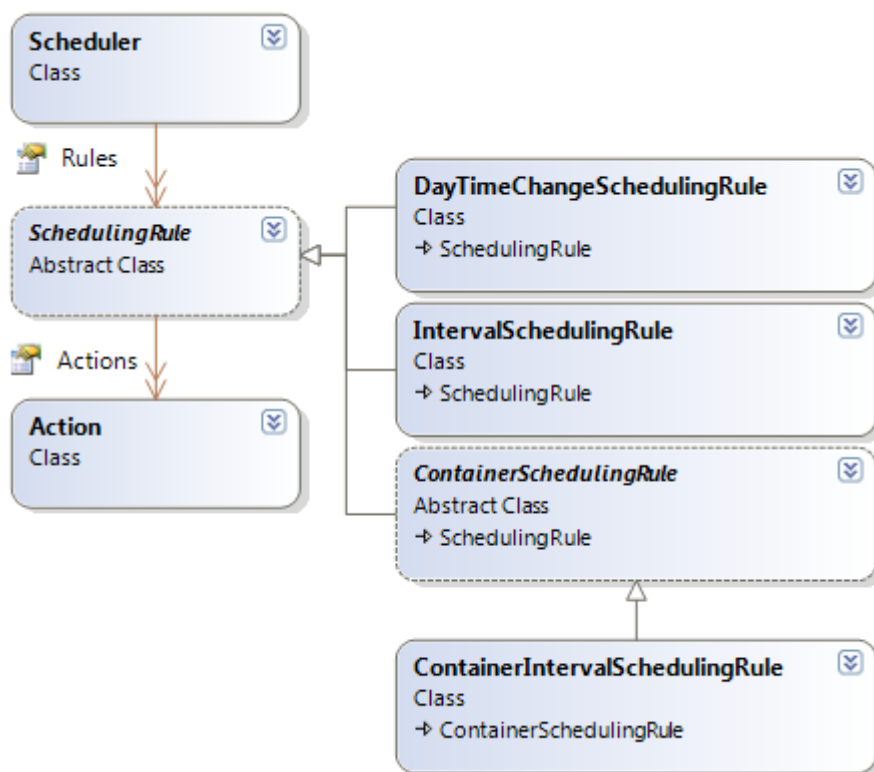
Zvieratko v PDA je aplikácia, ktorá beží v PDA, obsahuje jednoduché grafické používateľské rozhranie, je pre ňu charakteristické oddelenie logiky aplikácie od zobrazovania. Aplikácia je navrhnutá s ohľadom na budúce rozširovanie. Podrobný návrh je zobrazený na obrázku 9-33. Najdôležitejšie triedy aplikácie:

- *Simulation*: Základná kontajnerová trieda, ktorá obsahuje plánovač (*Scheduler*) a ihrisko (*Playground*)
- *Playground*: Trieda, ktorá predstavuje herný svet v PDA. Obsahuje zoznam miest a inštanciu triedy *PetAvatar*
- *PetAvatar*: Trieda, reprezentujúca vnútorný stav zvieratka. Zahŕňa inventár a štatistiky zvieratka
- *Inventory*: Inventár obsahuje zoznam predmetov, ktoré zvieratko vlastní
- *Statistics*: Trieda obsahuje zoznam štatistických záznamov
- *StatisticEntry*: Štatistický záznam, ktorý obsahuje informáciu o svojej aktuálnej hodnote, a zoznam hraníc intervalov, ktoré vymedzujú stav záznamu v textovej reprezentácii. (Low, Good, Maximum,...)
- *PetState*: Opisuje vnútorný stav zvieratka, a zabezpečuje striedanie obrázkov v animácii, ktorú riadi *PetAvatar*
- *Place*: Každé miesto má svoj vzhľad, zoznam mini-aktivít, ktoré sa na ňom dajú spustiť, a svoj vnútorný stav, ktorý je závislý od doby dňa. Miesto si eviduje aj zoznam svojich susedov. Herný svet je preto reprezentovaný formou jednoduchého grafu bez slučiek.



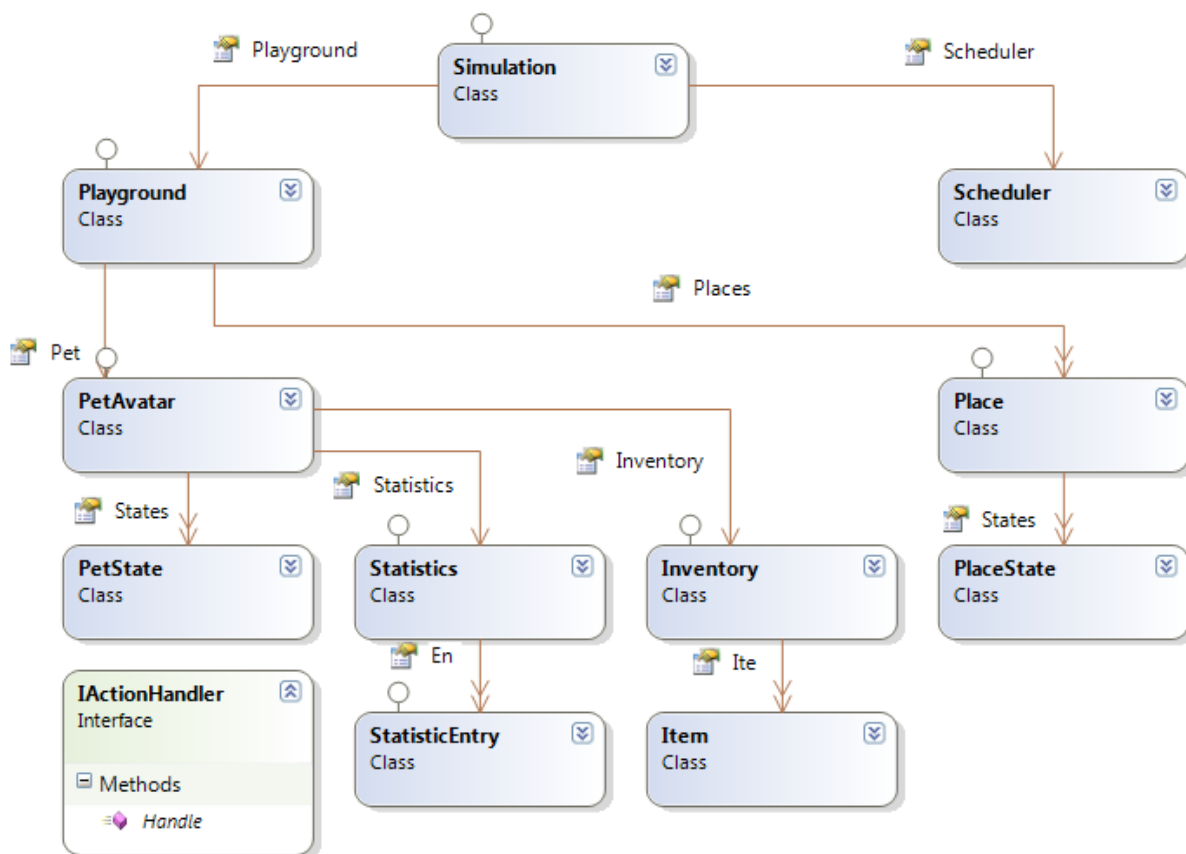
Obrázok 9-31. Mini-aktivity v hernom svete PDA klienta.

- *MiniActivity*: Abstraktná trieda reprezentujúca každú mini-aktivitu, ktorá je dostupná na tom ktorom mieste v hernom svete. Vytvára rozhranie pre začiatok, koniec hry a nasledujúci ťah. Každá mini-aktivita obsahuje zoznam štatistík zvieratka, ktoré ovplyvňuje (zvyčajne na konci hry)
- *Scheduler*: Plánovač, zabezpečujúci opakovanie dejov, ktoré môžu ovplyvniť každý aspekt herného sveta



Obrázok 9-32. Plánovanie a plánovacie pravidlá.

- *SchedulingRule*: Každé plánovacie pravidlo obsahuje zoznam akcií, ktoré sa majú vykonať v prípade, že je inštancia pravidla aplikovateľná v čase, keď ju plánovač spúšťa. Abstraktná trieda umožňuje ďalej implementovať špecifické jednoduché a kompozitné plánovacie pravidlá. Podrobný návrh plánovania je zobrazený na obrázku 9-32
- *Action*: Akcia v sebe zahŕňa textovú správu, zoznam parametrov, a zoznam jej prijímateľov, ktorí musia implementovať rozhranie *IActionHandler* (všetky objekty herného sveta)

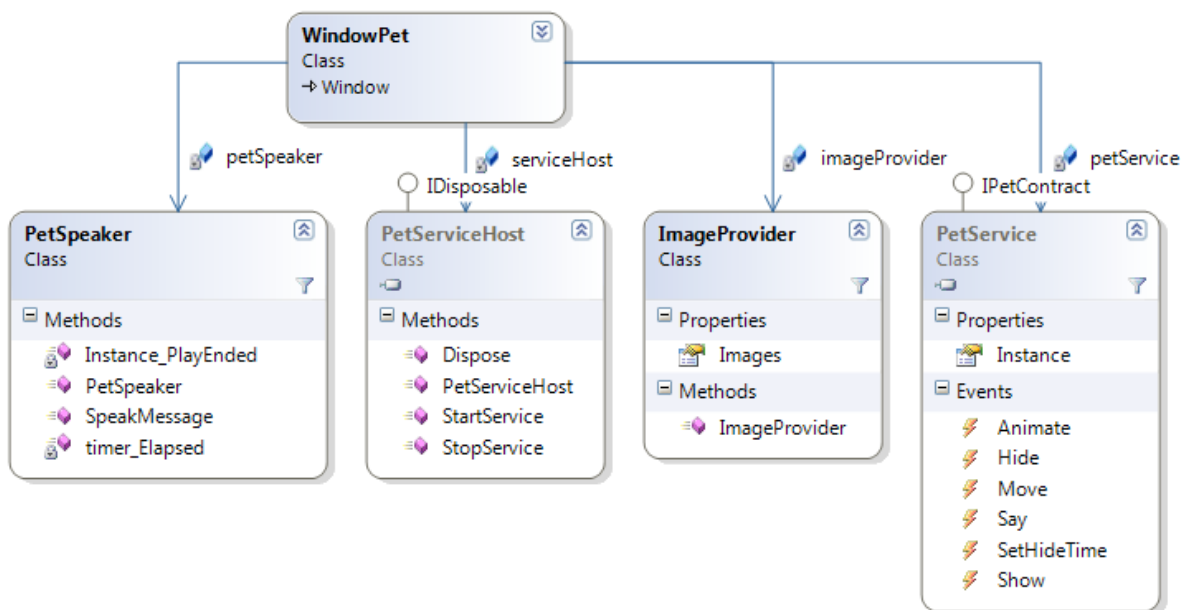


Obrázok 9-33. Podrobný návrh aplikácie Zvieratko v PDA.

9.3.11 Zvieratko

Aplikácia slúži ako sprievodca hráča počas herných aktivít. Obsahuje unifikované rozhranie pre zobrazovanie, posúvanie, prehrávanie zvuku a animovanie zvieratka. Podrobný návrh aplikácie sa nachádza na obrázku č. 9-34. Aplikácia obsahuje nasledovné dôležité triedy:

- *WindowPet*: Trieda slúžiaca na zobrazovanie okna zvieratka, a obsluhujúca reakcie na externé volania pochádzajúce zo služby PetService
- *PetServiceHost*: Trieda zapúzdrujúca funkčnosť hostujúcej servisnej aplikácie. Obsahuje metódy na zapnutie a vypnutie služby
- *PetService*: Služba sprostredkujúca komunikačné rozhranie medzi aplikáciou a vonkajším svetom (v našom prípade herné aktivity, ktoré službu volajú prostredníctvom triedy PetCommunicator)
- *PetSpeaker*: Obalová trieda, ktorá poskytuje funkčnosť knižnice FMOD pre asynchrónne prehrávanie zvukov
- *ImageProvider*: Trieda, poskytujúca obrázky na základe vnútorného stavu zvieratka



Obrázok 9-34. Podrobný návrh aplikácie zvieratko.

9.3.12 Zvieratko v Sidebar Gadgete

V systéme vystupuje aplikácia vo forme Windows Vista Sidebar Gadget, cez ktorý si môžu deti spustiť buď samotnú hru (klienta herného sveta) alebo priamo obľúbené herné aktivity. Na zrelaxovanie si tiež môže dieťa nechať povedať rozprávku. Aplikácia v gadgete je napísaná ako HTML stránka s javascriptom používajúcim Gadget API. Využíva grafické animačné efekty pre zvýšenie atraktívnosti rozhrania. Grafické rozhranie gadgetu je zobrazené na obrázku 9-35.



Obrázok 9-35. Vista Sidebar Gadget.

10 Implementácia

10.1 Výber implementačného jazyka a prostredia

Pre realizáciu projektu sme vybrali integrované vývojové prostredie Microsoft Visual Studio 2005 Professional, pretože je použitie jedného z jazykov postavených na platforme .NET Framework podmienkou pre účasť v súťaži Imagine Cup, v rámci ktorej bol systém navrhnutý a implementovaný. Celý systém je implementovaný v jazyku Visual C#. Ďalšou podmienkou pre účasť v súťaži je použitie ASP.NET webovej služby, preto je časť systému nasadená na webovom serveri z balíka služieb Microsoft Internet Information Services.

10.2 Komponenty systému

Kapitola obsahuje prehľad všetkých komponentov systému, ich stručný popis a vymenovanie vzájomných závislostí.

Tabuľka 10-1. Zoznam komponentov v systéme.

Komponent	Popis
Client.Controls.WinForms	Knižnica obsahujúca prvky používateľského rozhrania používané v hrách
Client.Core	Knižnica pre komunikáciu so službou svet
Client.Hangman	Knižnica pre komunikáciu so službou Hangman (testovanie rámca herných aktivít)
Client.Hangman.UI	Používateľské rozhranie pre klienta služby Hanuman (testovanie rámca herných aktivít)
Client.WinForms.UI	Implementácia testovacieho klienta (testovanie služby Svet)
DomainKnowledge.Client.Math	Knižnica pre prístup k službe úložiska matematických znalostí
DomainKnowledge.Common	Základná knižnica služieb doménových znalostí
DomainKnowledge.Common.Tests	Unit testy knižnice DomainKnowledge.Common
DomainKnowledge.Data	Základná knižnica pre dátovú reprezentáciu v službách doménových znalostí
FMOD	Knižnica určená pre prehrávanie zvukov a hudby

LongWords	Aplikácia „povedz mi dlhé slovo“
MapEditor	Aplikácia Editor máp
MapEditor.Common	Knižnica, ktorá obsahuje model herného sveta
MapEditor.Controls	Knižnica obsahujúca prvky používateľského rozhrania použité v Editore máp
MapEditor.Core	Knižnica obsahujúca triedy pre prácu s projektom v Editore máp
MathClient	Hra Matematické balóny
MathKnowledgeService	Služba sprostredkujúca otázky z odboru matematika
ModelViewer**	Aplikácia slúžiaca na zobrazenie a automatizované ukladanie natočení modelov do animovaných sekvencií
MusicClient	Hra Opakovanie melódie
ParticleTest**	Aplikácia určená na modelovanie časticových efektov a ich renderovanie do animovaných sekvencií
PathFinding	Implementácia algoritmu hľadania cesty (A*)
Pelikan.Speech	Knižnica obsahujúca triedy na rozpoznávanie hlasu
PelikanClient**	Klient herného sveta
PelikanClient.Gui**	Knižnica s triedami reprezentujúcimi grafické používateľské rozhranie a jeho ovládacie prvky
PelikanClient.Gui.Test**	Aplikácia na testovanie grafického používateľského rozhrania, slúžiaca ako inkubátor pre navrhovanie nových herných dialógových okien
PelikanClient.Input**	Knižnica obsahujúca triedy na spracovanie vstupu od používateľa
PelikanClient.Login	Aplikácia pre vytváranie a správu účtov, spúšťanie klienta herného sveta

Pet.Application.WPF	Aplikácia slúžiaca na zobrazovanie animovaného zvieratka v prostredí herných aktivít.
Pet.Client	Knižnica pre prístup k službe Pet
Pet.Client.Test	Aplikácia pre interaktívne testovanie knižnice Pet.Client
Pet.MobileClient	Mobilný klient zvieratka v PDA, zabezpečuje zobrazovanie simulácie herného sveta.
Pet.MobileClient.Common	Knižnica, ktorá obsahuje všetky triedy potrebné pre simuláciu herného sveta.
Pet.ServiceLibrary	Knižnica zabezpečujúca komunikáciu aplikácie Pet.Application.WPF s hernými aktivitami.
RhythmEditor	Editor úloh pre hru Malý hudobník
RhythmGame	Hra Malý hudobník
Server	Aplikácia hostujúca služby
Server.Core	Knižnica združujúca komponenty infraštruktúry servera
Server.Ratings	Služba hodnotení
Server.World	Služba svet, rámec herných aktivít a služby herných aktivít
ShaderTest**	Knižnica združujúca komponenty na kreslenie grafických efektov
SmartPictureClient	Aplikácia, obsahujúca implementáciu hernej aktivity SmartPictures.
SmartPictureEditor	Aplikácia obsahujúca implementáciu editora pre hernú aktivitu SmartPictures
SmartPictureStorage*	Webová služba ukladajúca obrázky pre hernú aktivitu SmartPictures
SoundGame	Hra Zvuky
StorageServices.Common	Implementácia všeobecného úložiska XML reprezentovateľných aktivít
StoryReader	Aplikácia Čítanie rozprávok

StoryTeller*	Služba úložisko rozprávok
StrokeClient	Hra <i>Strokes</i>
StrokeClient.Connector	Knižnica pre pripojenie k službe <i>Strokes</i>
StrokeEditor	Editor úloh hry <i>Strokes</i>
StrokeStorage*	Úložisko úloh hry <i>Strokes</i>
UserModel.Client	Knižnica pre manipuláciu s modelom používateľa
UserModel.Common	Implementácia modelu používateľa

Tabuľka 10-2. Zoznam závislostí komponentov.

Komponent	Závislosti
Client.Controls.WinForms	-
Client.Core	-
Client.Hangman	-
Client.Hangman.UI	Client.Hangman
Client.WinForms.UI	Client.Core
DomainKnowledge.Client.Math	DomainKnowledge.Common MathKnowledgeService Server.Core UserModel.Client UserModel.Common
DomainKnowledge.Common	-
DomainKnowledge.Common.Tests	-
DomainKnowledge.Data	DomainKnowledge.Common
FMOD	-
LongWords	Client.Controls.WinForms
MapEditor	MapEditor.Common MapEditor.Controls MapEditor.Core
MapEditor.Common	-
MapEditor.Controls	MapEditor.Common

	MapEditor.Core
MapEditor.Core	MapEditor.Common
MathClient	Client.Controls.WinForms DomainKnowledge.Client.Math DomainKnowledge.Common Pelikan.Speech Pet.Client Server.Core UserModel.Common
MathKnowledgeService	-
ModelViewer**	-
MusicClient	Client.Controls.WinForms FMOD Pet.Client
ParticleTest**	PelikanClient.Input
PathFinding	-
Pelikan.Speech	-
PelikanClient**	Client.Core FMOD MapEditor.Common PathFinding Pelikan.Speech PelikanClient.Gui PelikanClient.Input
PelikanClient.Gui**	PelikanClient.Input
PelikanClient.Gui.Test**	PelikanClient.Gui
PelikanClient.Input**	-
PelikanClient.Login	Client.Controls.WinForms
Pet.Application.WPF	FMOD Pet.ServiceLibrary
Pet.Client	-
Pet.Client.Test	Pet.Client
Pet.MobileClient	Pet.MobileClient.Common

Pet.MobileClient.Common	-
Pet.ServiceLibrary	-
RhythmEditor	FMOD RhythmGame
RhythmGame	Client.Controls.WinForms FMOD Pet.Client
Server	Server.Core Server.Ratings Server.World
Server.Communication	-
Server.Core	-
Server.Ratings	-
Server.World	DomainKnowledge.Client.Math DomainKnowledge.Common DomainKnowledge.Math MapEditor.Common PathFinding Server.Core SmartPictureStorage StrokeStorage
ShaderTest**	-
SmartPictureClient	DomainKnowledge.Client.Math DomainKnowledge.Common DomainKnowledge.Data Pelikan.Speech Pet.Client SmartPictureStorage
SmartPictureEditor	SmartPictureStorage
SmartPictureStorage*	-
SoundGame	Client.Controls.WinForms Client.Core Client.WinForms.UI

	FMOD Pet.Client
StorageServices.Common	-
StoryReader	Client.Controls.WinForms
StoryTeller*	-
StrokeClient	Client.Controls.WinForms Pelikan.Speech Pet.Client Server.Core StrokeClient.Connector StrokeStorage UserModel.Client UserModel.Common
StrokeClient.Connector	Client.Controls.WinForms Client.Core
StrokeEditor	StrokeStorage
StrokeStorage*	-
UserModel.Client	UserModel.Common
UserModel.Common	Server.Core

11 Overenie riešenia

Kapitola sa zaoberá overením nášho systému. Je v nej opísané testovanie systému, konkrétne plán testovania a testovacie údaje.

11.1 Testovanie

V tejto časti sa nachádza opis testovania ako aj testovacích dát, ktoré sme počas testovania jednotlivých častí ale aj systému ako celku použili.

11.1.1 Plán testovania

Jednotlivé aktivity sa testovali počas vývoja pomocou testovacích dát opísaných v druhej kapitole.

Pre účely živej ukážky počas prezentácie v národnom kole súťaže Imagine Cup sme bol náš systém podrobený špeciálnemu testovaniu. Jeho hlavným cieľom bola eliminácia všetkých chybových stavov, ktoré by mohli v priebehu interakcie so systémom nastať. Prezentácia a teda aj testovanie boli založené na nasledujúcom scenári:

1. Otvorenie okna gadgetu a spustenie herného sveta kliknutím na príslušnú ikonu
2. Prechod z počiatočnej oblasti do domčeka
3. Rozhovor s učiteľom matematiky
4. Spustenie, priebeh a ukončenie hry Matematické balóny
5. Pokračovanie rozhovoru s učiteľom matematiky, zadanie úlohy
6. Prechod do oblasti Divočina
7. Vytvorenie tímu s druhým hráčom a hranie hry Tímová matematika interakciou s duchmi
8. Rozhovor s pánom duchov, spustenie hry Strokes
9. Návrat do dediny
10. Rozhovor s postavou Zlodej, hranie hry Malý hudobník
11. Rozhovor s učiteľom hudby, hranie hry Opakovanie melódie
12. Opustenie herného sveta
13. Spustenie hry SmartLines z gadgetu a jej hranie pomocou dvoch myší
14. Spustenie aktivity Čítanie rozprávok

11.1.2 Testovacie dáta

V tejto časti sa nachádza popis dát, ktoré boli použité v rámci testovania projektu. Sú rozčlenené podľa aktivít a služieb, ktoré ich využívali. Všetky dáta boli vytvorené v slovenskej a anglickej verzii kvôli overeniu lokalizačných mechanizmov.

Herný svet

Pre overenie klienta zobrazujúceho herný svet, služby Svet a ich vzájomnej komunikácie boli vytvorené nasledujúce dáta:

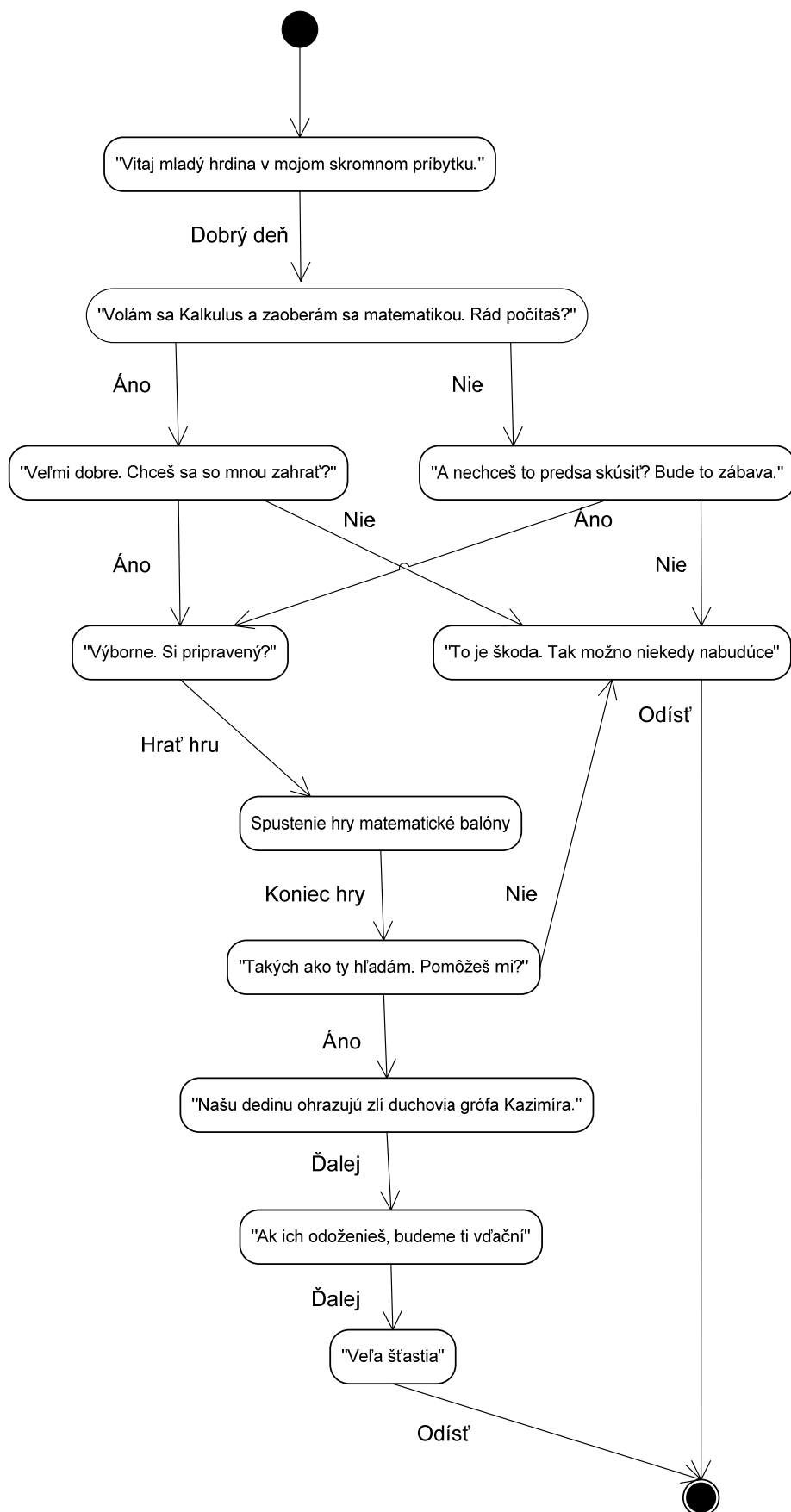
- Mapa: Obsahuje tri oblasti vytvorené v editore máp a to vnútro domu, dedinu a divočinu (obrázok 11-1). Pre každé políčko je definované jeho poradie vykresľovania *zOrder* a priechodnosť. Ďalej sú na mape definované meta objekty a to miesta prechodov medzi oblasťami a objekty, ktoré určujú pravdepodobnosť a oblasť vytvárania rôznych herných postáv (*Spawner*).
- Herné postavy: Vo svete sa nachádzajú nasledujúce herné postavy:
 - o Učiteľ matematiky
 - o Obchodník
 - o Zlodej
 - o Učiteľ hudby
 - o Duch
 - o Pán duchov
 - o Dedinčan

Postavy sa v oblasti náhodne pohybujú (duch, dedinčan) alebo stoja na mieste (ostatné). Každá postava okrem dedinčana má definované hlášky, pomocou ktorých interaguje s hráčom. Diagram dialógu postavy *Učiteľ matematiky* sa nachádza na obrázku 11-2.

- Nahovorené hlášky postáv: Všetky hlášky postáv sú nahovorené a uložené ako mp3 súbory na klientovi.



Obrázok 11-1. Oblasť herného sveta Divočina.



Obrázok 11-2. Diagram dialógu postavy Učiteľ matematiky.

Herné aktivity

Pre každú hernú aktivitu bolo vytvorených niekoľko úloh rôznej náročnosti tak, aby sa na nich dali overiť všetky herné možnosti. Pri ich tvorbe boli podľa druhu aktivity použité pre tento účel vytvorené editory.

Hry Malý muzikant a zvuky obsahujú hudobné ukážky a skladby, ktoré boli zložené a nahrané špeciálne pre účely tohto projektu. Niektoré z nich sú na motívy známych piesni pre deti.

Aktivita StoryTeller

Pre túto aktivitu bolo použitých niekoľko textov rozprávok. Všetky boli nahovorené a zvukovo upravené tak, aby bol docielený efekt, že rozprávku číta zvieratko.

Služba Zvieratko

Pre každú aktivitu je definovaná a nahratá množina hlášok, ktoré zvieratko pri určitej hernej situácii povie. Zároveň môže použiť niektorú z emotikon, ktoré lepšie znázorňujú jeho pocity a stav hry.

12 Záver

Po oboznámení sa s témou súťaže Imagine Cup 2007 a analýze úspešných projektov z predchádzajúcich rokov, vyplývajúcej z našej úspešnej účasti v tejto súťaži sme dospeli k myšlienke projektu, predstavenej v tomto dokumente. Jej obsah predstavuje viacpoužívateľskú výukovú hru pre deti v predškolskom a ranom školskom veku.

Analyzovali sme problémovú oblasť, osnovy výučby v prvých ročníkoch na základných školách a tiež sme využili skúsenosti niektorých členov tímu s pedagogikou a s metodikou výučby hudby a špecifikovali sme doménové oblasti, obsah z ktorých náš projekt ponúka.

Z analýzy vyplynulo, že neexistuje projekt, ktorý by ponúkal podobné možnosti kolaborácie a zároveň výchovy a vzdelávania detí predškolského veku. Navrhli sme inovatívne rozhrania formou syntézy a rozoznávania reči a obrazu. Navrhli sme architektúru systému s ohľadom na jednoduchosť, rozšíriteľnosť a elegantnosť návrhu.

Ako výstup pre druhý kontrolný bod sme implementovali niektoré súčasti zamýšľaného systému, aplikáciou ktorých sme odhalili niektoré skryté vlastnosti novovytvorených technológií. Tvorbou prototypu sme zároveň vytvorili rámec pre ďalšie rozširovanie v nasledujúcich iteráciách evolučného vývoja.

Na základe vypracovaného prototypu sme rozšírili špecifikáciu produktu a vypracovali jeho podrobný návrh. Na základe podrobného návrhu sme produkt vytvorili, a podľa vytvorenej metodiky neskôr otestovali.

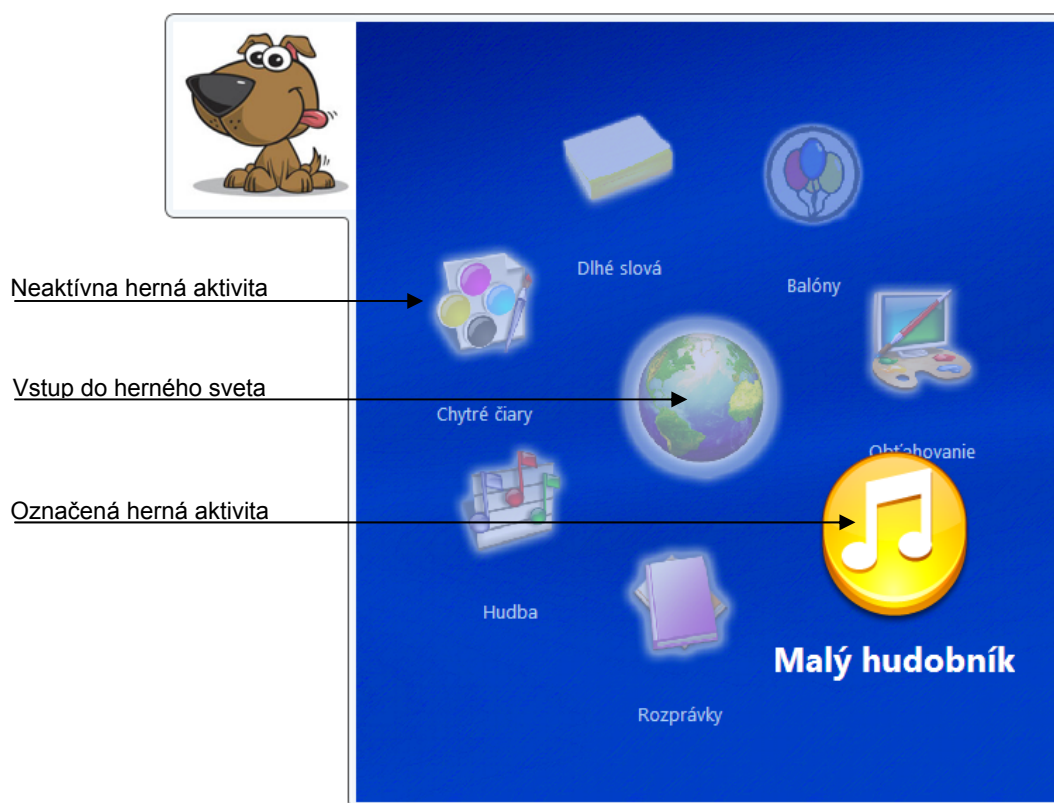
A Používateľská príručka

Príloha obsahuje používateľskú príručku ku hernému výukovému systému omega. V príručke je uvedený opis používateľského rozhrania aplikácie spolu so spôsobom práce na systéme.

A.1 Hlavné menu

Hlavné menu aplikácie z pohľadu hráča je podľa obrázku A-1 prístupné prostredníctvom Sidebar Gadgetu, v ktorom sa nachádza obraz zvieratka, ktoré ho sprevádza počas celej hry. V centre animovaného menu sa nachádza odkaz na vstup do herného sveta, okolo ktorého krúžia ikony hier, ktoré môže hráč hrať aj mimo herného sveta v špeciálnom výukovom režime.

Hlavné menu sa automaticky uzatvorí, keď hráč klikne mimo oblasť okna.



Obrázok A-1. Hlavné menu aplikácie.

A.2 Zvieratko

Zvieratko zobrazené na obrázku A-2 je samostatná aplikácia, ktorá dostáva povely z hernej aktivity, ktorá je práve spustená, a sprostredkováva hráčovi ďalšiu úroveň interaktívnosti. Animáciami a hovoreným prejavom spríjemňuje hráčovi hranie herných aktivít, povzbudzuje ho v napredovaní



Obrázok A-2. Prvok používateľského rozhrania: Zvieratko.

Zvieratko rado komentuje výkon hráča, väčšinou ho chváli a povzbudzuje, ale niekedy ho dokáže pokarhať. Vždy, keď sa v aplikácii stane niečo dôležité, objaví sa a informuje hráča o svojom postoji ku situácii.

A.3 Herné aktivity

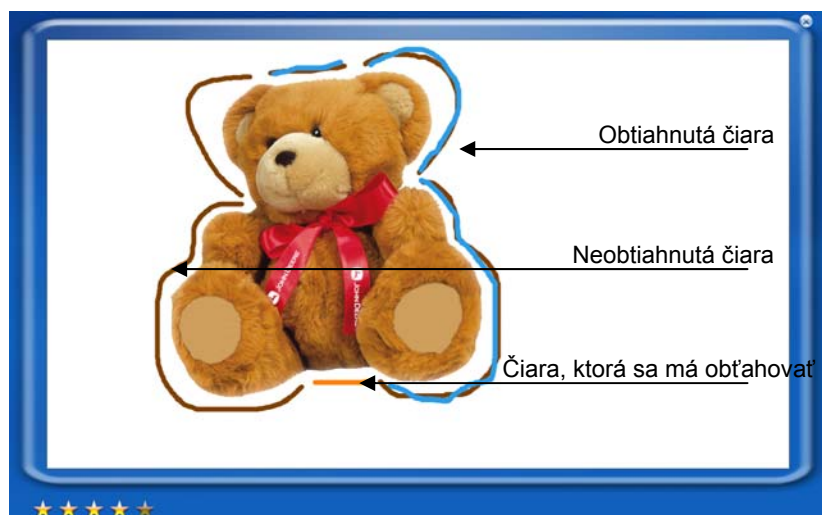
Za hernú aktivitu považujeme každú aplikáciu, v ktorej sa hráč prostredníctvom hrania učí. Každá herná aktivita pracuje v dvoch režimoch:

- Výukový režim: spúšťaný z hlavnému menu
- Režim skúšania: spúšťaný z herného sveta.

Používateľské rozhranie sa v rôznych režimoch príliš nemení. Mení sa správanie hry.

A.3.1 Obťahovanie

Obťahovanie je výuková hra, v ktorej hráč obťahuje vybranú predlohu. Vždy je zvýraznená aktuálna čiara, ktorú má hráč obtiahnuť. Každý ťah sa hodnotí zvlášť. Hra zobrazená na obrázku A-3 je vhodná na tréning jemnej motoriky.



Obrázok A-3. Používateľské rozhranie hry Obťahovanie.

Používateľské rozhranie

Grafické rozhranie aplikácie pozostáva z tabule a ukazovateľa správnosti. Na tabuli hráč obťahuje obrázok sledujúc sériu čiar. Systém mu ponúka vždy ďalšiu čiaru, keď dokončí kreslenie predchádzajúcej. Keď dokončí kreslenie poslednej čiary, hra sa ukončí.

Aplikáciu je možné ovládať aj s využitím rozpoznávania hlasových povelov: Zoznam povelov, ktoré systém rozpoznáva je nasledujúci :

- *Šírka pera* : (Príklad: „Width ten“) Nastavuje šírku pera, ktorým sa obkresľuje.
- *Farba pera*: (Príklad: „Color blue“) Nastavuje farbu pera, ktorým sa obkresľuje.
- *Ukončenie aplikácie*: (Povel: „Close that“) Predčasne ukončí aplikáciu

Prípady použitia (Pri spustení z hlavného menu)

Vytvorenie a spustenie hry jedného hráča

Po spustení hry sa hráč dostane do vstupnej obrazovky, kde vykoná nasledujúcu sériu krokov:

1. Hráč napíše svoje meno do políčka *name*
2. Hráč napíše meno hry do políčka pod tlačidlom open game
3. Stlačí tlačidlo Open Game, čím vytvorí hru a zaregistruje sa ako hráč
4. Stlačí tlačidlo Start Game, a spustí tak hru

Vytvorenie a spustenie hry niekoľkých hráčov

Vytvoriť hru viacerých hráčov je tiež veľmi jednoduché.

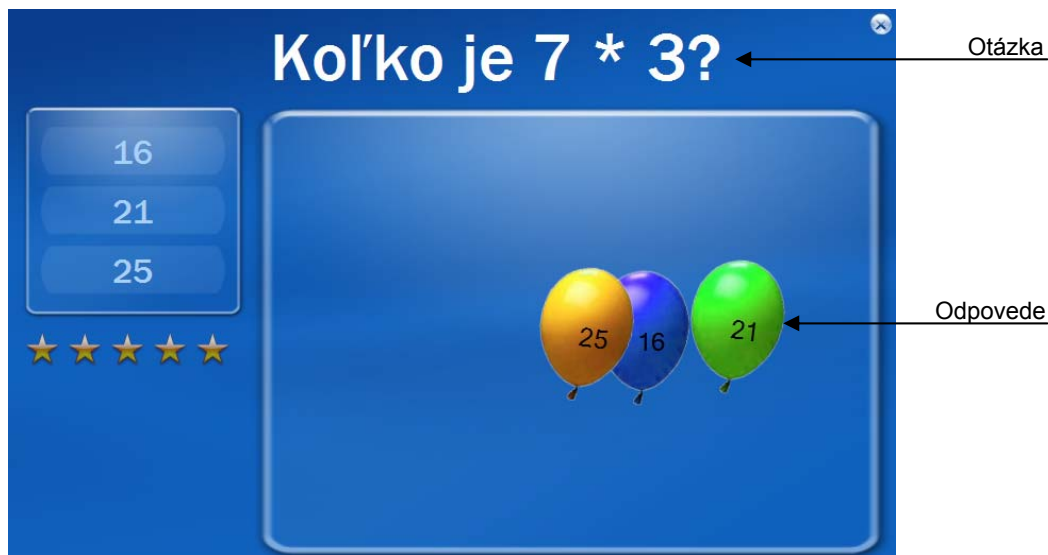
1. Hráč vytvorí hru podľa krokov 1 a 2 predchádzajúceho postupu
2. Druhý hráč napíše svoje meno, vyberie si otvorenú hru a stlačí tlačidlo Join, po stlačení ktorého čaká na prvého hráča, aby hru spustil.
3. Prvý hráč spustí hru

Hranie hry niekoľkými hráčmi

Hra viacerých hráčov je veľmi jednoduchá. Každému hráčovi sa ukazuje, ktoré čiary už nakreslili jeho spoluhráči, A aj ako sa im pri obkresľovaní darilo.

A.3.2 Matematické balóny

Hra je určená na tréning postrehu a precvičovanie znalostí z domény matematika. Hráč odpovedá na zadanú otázku prasknutím balóna so správnou odpoveďou. Za svoje odpovede je hodnotený. Čím viac otázok správne zodpovie, tým vyššie hodnotenie v hre dosiahne a tým ťažšie otázky bude dostávať.



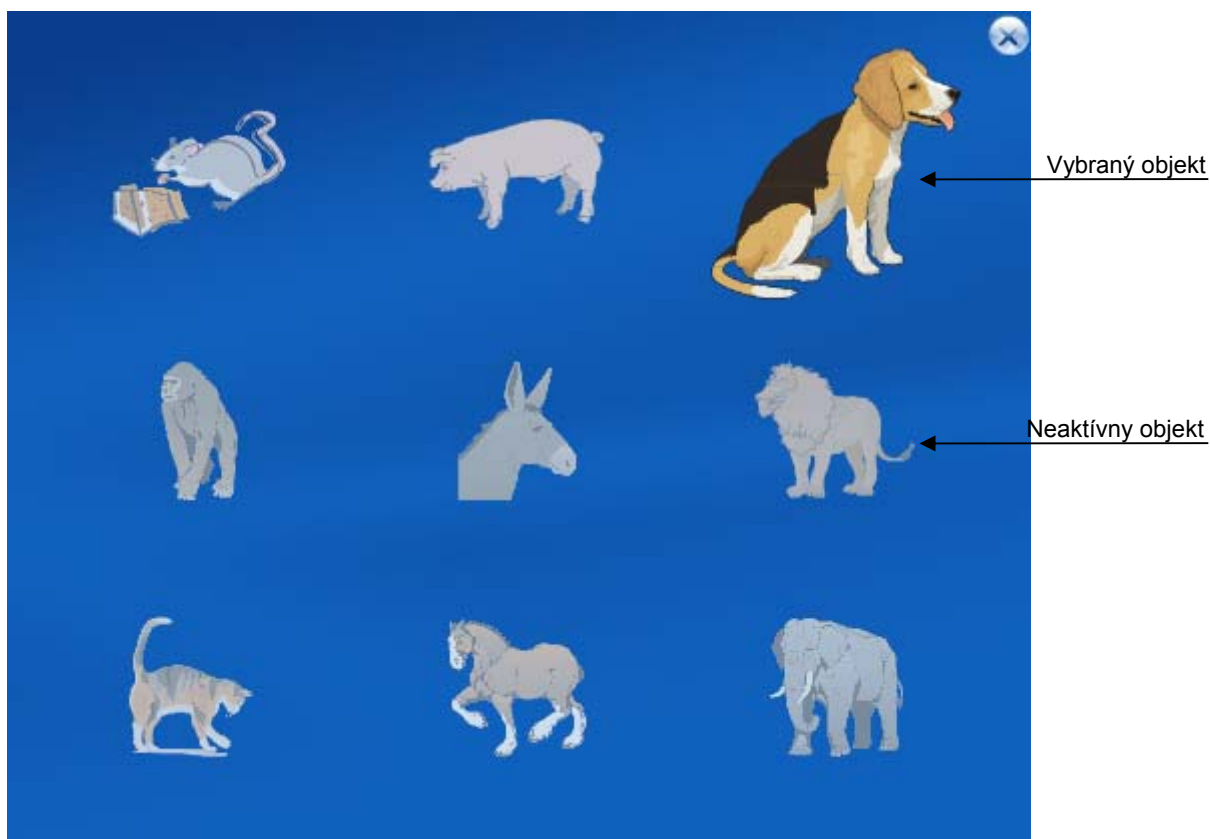
Obrázok A-4. Používateľské rozhranie hry Matematické balóny.

Používateľské rozhranie

Grafické rozhranie zobrazené na obrázku A-4 aplikácie pozostáva z tabule, ukazovateľa správnosti voľby odpovedí. Na tabuli hráč triafa balóny a odpovedá tak na zadanú otázku. Po správnom zodpovedaní otázky dostáva hráč ďalšiu otázku. V režime výučby je dĺžka hry neobmedzená, v režime skúšania je hra obmedzená počtom otázok.

A.3.3 Zvuky

Hra zvuky je určená na tréning asociatívnej pamäte pre menšie deti. Ku každému obrázku je priradený zvuk alebo množina zvukov, ktorá ho charakterizuje.



Obrázok A-5. Používateľské rozhranie hry zvuky.

Používateľské rozhranie

Grafické rozhranie aplikácie zobrazené na obrázku A-5 pozostáva z tabule, na ktorej je zobrazených 9 obrázkov, ktoré spolu tematicky súvisia. Správanie aplikácie sa mení v závislosti na režime

Prípady použitia

Hra vo výučbovom režime

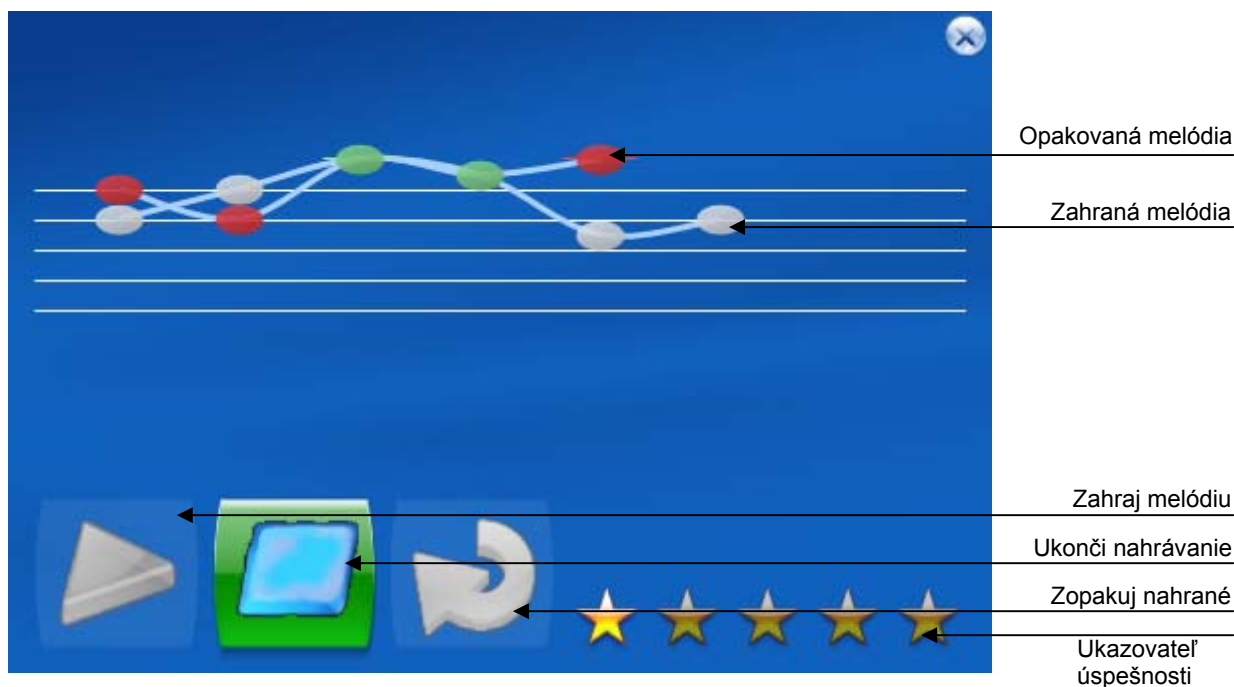
Vo výučbovom režime sa po kliknutí na ľubovoľný objekt ozve zvuk, ktorý je s ním asociovaný.

Hra v režime skúšania

V režime skúšania sa najprv ozve zvuk objektu, potom musí hráč kliknúť na zodpovedajúci obrázok. Ak hráč odpovie nesprávne, hra mu zvuk zopakuje a hráč si môže vybrať znova. Ak odpovie správne môže pokračovať v hádaní iného zvuku.

A.3.4 Opakovanie melódie

Hra opakovanie melódie má precvičovať melodické cítenie hráča a jeho schopnosť rozpoznať falošné tóny. Správnym rozpoznaním a zopakovaním melódie sa hráč zároveň učí nové melódie.



Obrázok A-6. Používateľské rozhranie hry hudba.

Používateľské rozhranie

Používateľské rozhranie aplikácie zobrazené na obrázku A-6 sa skladá z troch hlavných častí

- *Riadiaci pruh*: Je tvorený tlačidlami, ktoré riadia beh aplikácie
- *Notová osnova*: Tvorená piatimi hlavnými linkami a niekoľkými pomocnými linkami, ktoré sa zobrazujú v prípade potreby. Na notovej osnove sú zobrazené noty, ktoré má dieťa reprodukovať. Noty zodpovedajú prehranej ukážke. Počas reprodukcie sa na notovej osnove zobrazujú aj noty, ktoré zodpovedajú prejavu dieťaťa.
- *Ukazovateľ úspešnosti*: Hviezdičky ukazujú mieru úspešnosti pri reprodukcii melódie

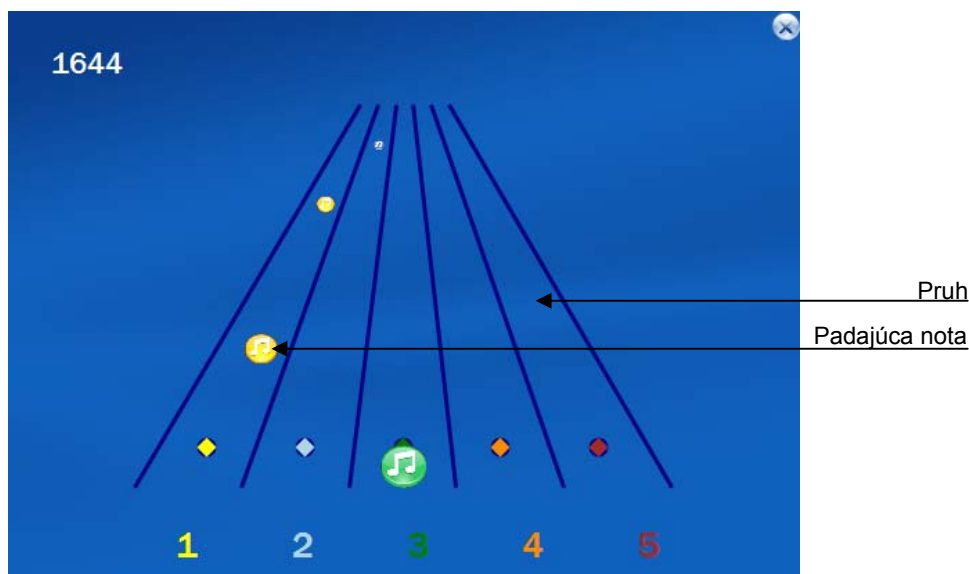
Prípady použitia

Postup hry

1. Hráč spustí hru.
2. Systém mu prehrá melódiu. Hráč stlačí tlačidlo nahrávanie
3. Zaspieva, prípadne zapíska, alebo zahrá na hudobnom nástroji melódiu
4. Stlačí tlačidlo ukončenie nahrávania
5. Systém ohodnotí nahranú melódiu
6. Hráč si môže vyžiadať opätovné prehranie hudby stlačením tlačidla Zahraj melódiu.

A.3.5 Malý hudobník

Malý hudobník je hra, ktorá testuje postreh a rytmické cítenie dieťaťa. Noty padajú smerom k hráčovi v závislosti od rytmu pesničky. Hráčovi sú za správne trafené noty priradené body.



Obrázok A-7. Používateľské rozhranie hry Malý hudobník.

Používateľské rozhranie

Používateľské rozhranie aplikácie zobrazené na obrázku A-7 sa skladá z troch hlavných častí

- *Ukazovateľ skóre*: Číselné ohodnotenie úspešnosti hráča
- *Herné pruhy*: V závislosti od počtu pruhov sa na obrazovke zobrazí zodpovedajúci počet oddeľujúcich čiar.
- *Tlačidlo znova*: Stlačením tlačidla znova sa načíta nová hra.

Prípady použitia

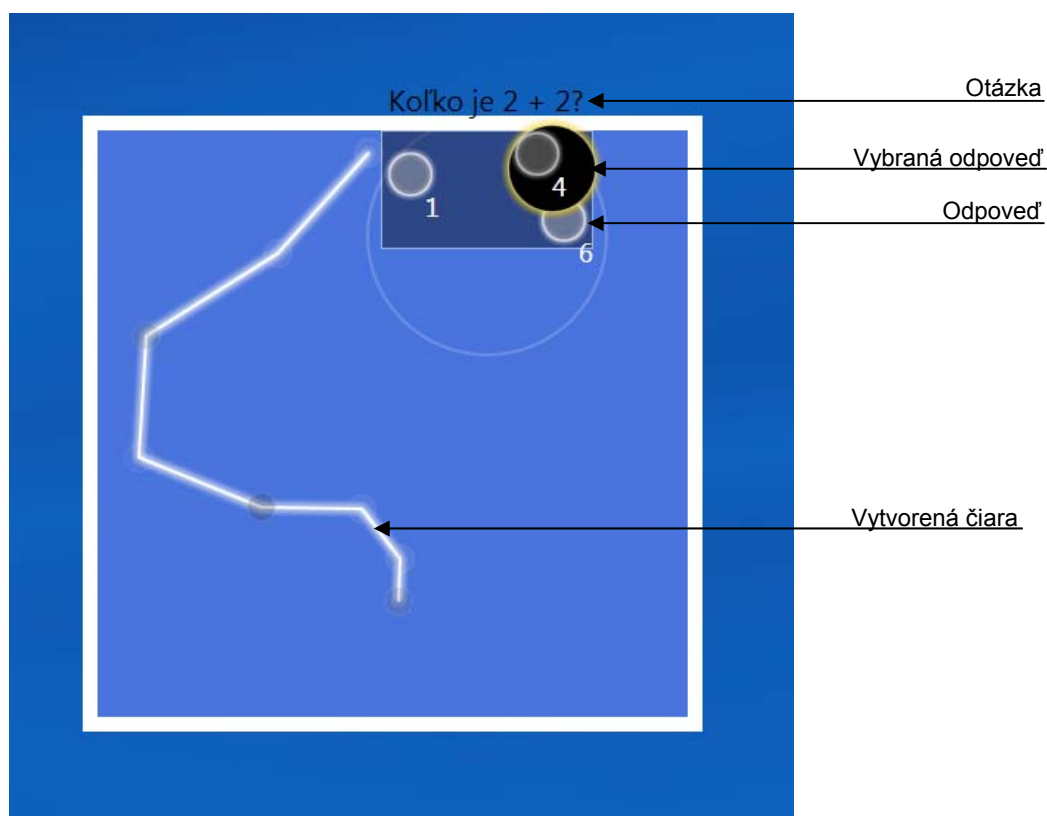
Postup hrania

1. Hráč spustí hru.
2. Zapne sa melódia a z hornej časti herných pruhov začnú zostupovať farebné noty.
3. V čase, keď sa nota/noty priblížia na spodnú časť obrazovky, musí hráč pridržať zodpovedajúcu klávesu/klávesy a stlačiť *medzerník*.
4. V prípade, že boli podržané správne klávesy, hráčovi sa započíta skóre a zodpovedajúce pruhy sa ofarbia oranžovou farbou.
5. Hra sa skončí zároveň s dohramím pesničky.
6. Hráč má možnosť pokračovať v hre aj po skončení

A.3.6 Chytré čiary

Chytré čiary sú hrou, ktorá slúži na rozvoj doménových znalostí zaujímavou cestou obkresľovania obrázku, ktorý môže mať taktiež súvis s doménovou oblasťou.

Hráč odpovedaním na otázky postupne získava prehľad o obrázku, ktorý obkresľuje.



Obrázok A-8. Používateľské rozhranie hry Chytré čiary.

Používateľské rozhranie

Používateľské rozhranie zobrazené na obrázku A-8 sa skladá z nasledujúcich elementov:

- *Plocha obrázku:* Plocha v ktorej sa po obkreslení zobrazí obrázok.
- *Chytrá čiara:* Všetky otázky, ktoré hráč správne zodpovedal, tvoria body lomenej čiary
- *Oblasť otázky:* Aktuálne položená otázka vždy zaberá plochu, ktorá je rezervovaná pre text otázky a pre body, ktoré reprezentujú jednotlivé odpovede

Aplikáciu je možné ovládať aj s využitím rozpoznávania hlasových povelov: Zoznam povelov, ktoré systém rozpoznáva je nasledujúci :

- *Správna odpoveď:* (Príklad: „minus three“) Odpovedá na otázku priamo za použitia hovoreného slova.
- *Ukončenie aplikácie:* (Povel: „Close that“) Predčasne ukončí aplikáciu

Prípady použitia

Hra jedného hráča mimo herného klienta

Hra jedného hráča mimo klienta prebieha nasledovne:

- Hráč spustí hru a vyberie si obrázok

- Hra sa spustí, hráč si vyberá bod, ktorý označí, čím odpovedá na otázku. Ak odpovedá správne, zjaví sa mu ďalšia otázka, prvý bod získal. Ak už nejaké body predtým získal, pripojí sa získaný bod do čiary.
- Ak získal posledný bod, čiara je kompletná, a hráčovi sa zjaví obrázok, ktorý získal.
- Hra sa ukončí.

Hra jedného hráča spustená z herného klienta

Po spustení z herného klienta si hráč nevyberá obrázok, ktorý bude obkresľovať, ale vyberá mu ho počítač na základe jeho vedomostí, prípadne na základe prednastaveného správania vyplývajúceho z herného prostredia, v ktorom sa nachádza, resp. z úlohy, ktorú hráč dostal.

Hra viacerých hráčov v dueli

K duelu medzi hráčmi dochádza vtedy, keď sedia za jedným počítačom ku ktorému je pripojených niekoľko polohovacích zariadení. Vtedy sa hra automaticky prepne do režimu duel, v ktorom hráči dostávajú rovnaké otázky, a zvíťazí ten, ktorému sa podarilo najskôr zodpovedať všetky otázky.

A.3.7 Rozprávky

Rozprávky sú oddychová herná aktivita, v ktorej systém číta rozprávku, ktorú si hráč zvoli.



Obrázok A-9. Používateľské rozhranie hry Rozprávky.

Používateľské rozhranie

Rozhranie je zložené z troch častí.

- *Ovládací pruh*: Obsahujúci tlačidlá na prehrávanie zvuku a nahrávanie vlastnej verzie rozprávky.
- *Obrázok*: Obrázok súvisiaci s rozprávkou.
- *Text rozprávky*: Text rozprávky v písomnej podobe

A.3.8 Dlhé slová

Oddychová herná aktivita, v ktorej si môže hráč prezerať rôzne anglické zloženiny, a pobaviť sa na ich počúvaní.



Obrázok A-10. Používateľské rozhranie aktivity Dlhé slová.

Používateľské rozhranie

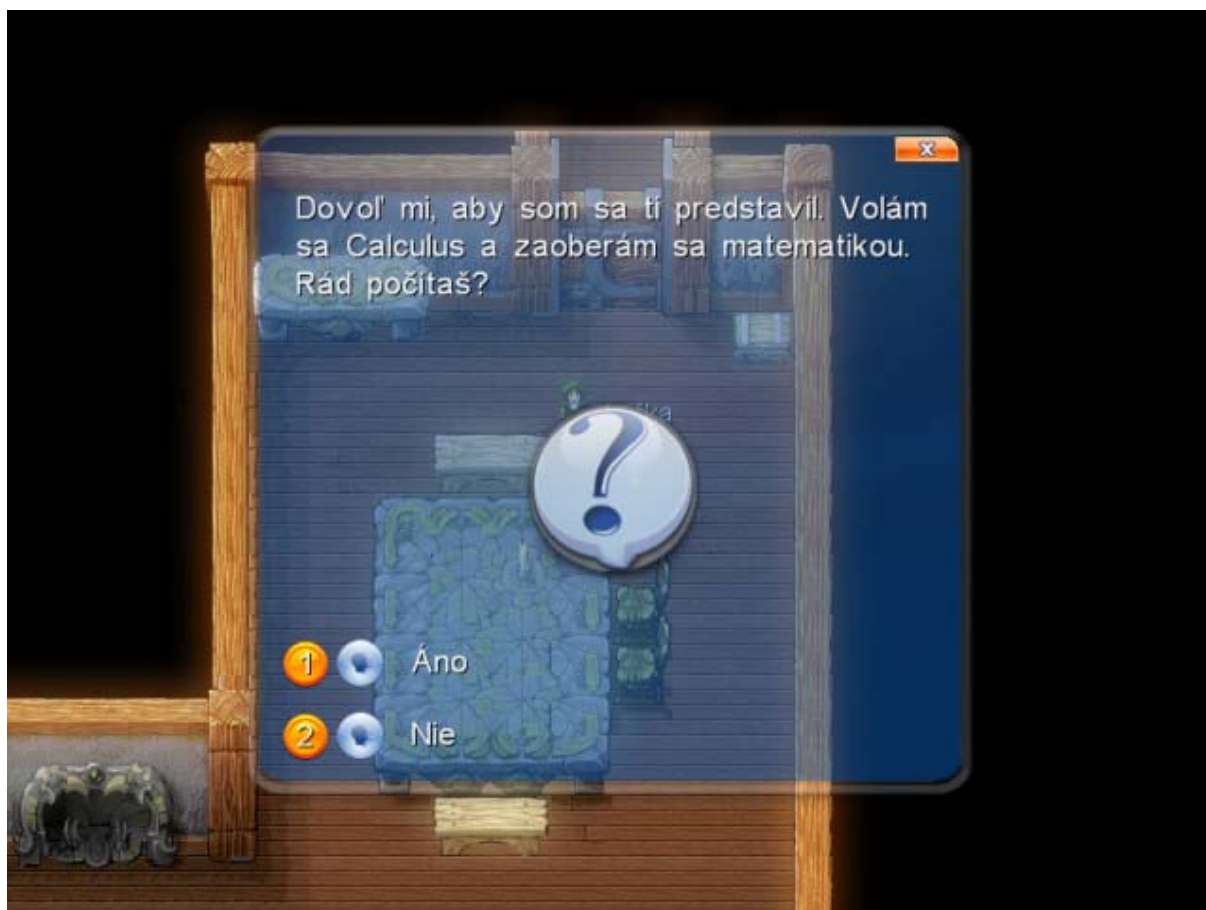
Rozhranie poskytuje možnosť ukázania nového dlhého slova, prípadne prečítať napísané dlhé slovo.

A.4 Herný klient

Aplikácia herného klienta reprezentuje virtuálny herný svet, v ktorom prebieha vzájomná interakcia medzi postavami hráčov a prostredím. Prostredníctvom neho sa spúšťajú všetky herné aktivity v závislosti od polohy vo svete alebo komunikácie s postavami ovládanými počítačom.

Používateľské rozhranie

Rozhranie pozostáva z hlavnej obrazovky znázorňujúcej graficky reprezentáciu herného sveta. Rozhranie je možné ovládať pomocou myši, klávesnice a hlasových povelov. Myš slúži na ovládanie grafického používateľského rozhrania a interakciu so svetom, ale umožňuje tiež pohyb hráča vo svete. Na pohyb vo svete a ovládanie prvkov a dialógových okien používateľského rozhrania sa používa ľavé tlačidlo, na interakciu so svetom a postavami pravé. Klávesnica poskytuje niekoľko funkčných skratiek pre nastavenie rozhrania, ale môže byť tiež použitá pre pohyb hráča ako náhrada namiesto myši. Používateľské rozhranie je taktiež možné ovládať pomocou hlasových povelov.



Obrázok A-11. Dialógové okno pre komunikáciu s postavami v hernom svete.

Komunikácia s postavami v hernom svete prebieha pomocou dialógového okna zobrazeného na obrázku A-11. Samotnú komunikáciu je možné iniciovať stlačením pravého tlačidla myši s kurzorom ukazujúcim na postavu s ktorou chce používateľ komunikovať. V hornej časti dialógu sa nachádza aktuálna postavou vypovedaná replika sprevádzaná hovorenou rečou, predmet dialógu je vizuálne indikovaný ikonou a v spodnej časti poskytuje možné odpovede alebo reakcie. Každá odpoveď je doplnená o možnosť nechať si prehrať znenie textu v zvukovej podobe hovoreným slovom, určená pre hráčov, ktorí ešte nevedia čítať.



Obrázok A-12. Dialógové okno s požiadavkou o vstup do tímu.

Ak používateľ v hre narazí na ďalšieho hráča, môže ho požiadať o vytvorenie tímu, prípadne požiadať o vstup do existujúceho tímu. Požiadavka sa iniciuje stlačením pravého tlačidla s kurzorom ukazujúcim na hráča, s ktorým chce používateľ vstúpiť do tímu. Druhému hráčovi je zaslaná požiadavka, po prijatí ktorej sa objaví dialógové okno s požiadavkou o vstup do tímu zobrazené na obrázku A-12. Požiadavka je indikovaná názornou ikonou a dialógové okno poskytuje dve tlačidlá – jedno na prijatie požiadavky a druhé na odmietnutie. V oboch prípadoch sa vráti žiadateľovi odpoveď v podobe ďalšieho dialógového okna informujúceho o prípadnom prijatí alebo odmietnutí požiadavky.

Zoznam hlasových povelov rozpoznávaných používateľským rozhraním je nasledujúci:

- *Pohyb hráča:* (Príklady: „move left“, „right“, „go north three“) Umožňuje pohyb hráča vo svete
- *Presun hráča na políčko:* (Príklad: „move to two six“) Slúži na presun hráča na políčko s konkrétnymi súradnicami
- *Prepnutie herného klienta z okna na plnú obrazovku:* (Povel: „toggle fullscreen“)
- *Vypnutie aplikácie:* (Povely: „exit“, „quit“)

A.5 PDA klient

PDA klient zobrazený na obrázku A-13 je aplikácia, ktorá sprevádza hráča na cestách. Nachádza sa v nej to isté zvieratko, ktoré ho sprevádza aj v hernom svet. Hráč sa môže o zvieratko starať a podporovať ho



Obrázok A-13. PDA klient.

Používateľské rozhranie

Aplikácia je rozdelená do troch obrazoviek:

- *Obrazovka herného sveta:* Hráč sa môže navigovať v hernom svete za pomoci tlačidiel, ktoré sú umiestnené na ľavej strane obrazovky. Stlačením tlačidla sa pohľad presunie do inej lokácie. Na pravej strane obrazovky sa nachádza zoznam herných aktivít, ktoré môže zvieratko v danej lokalite vykonať.
- *Obrazovka inventáru:* V nej si hráč môže prezerať, aké predmety zvieratko aktuálne vlastní.
- *Obrazovka štatistík:* Hráč má prehľad o aktuálnom vnútornom stave zvieratka.
-

B Systémová príručka

B.1 Systémové prerekvizity

Zoznam prerekvizít s nutnosťou inštalácie pred použitím systému:

- .NET Framework 3.0, .NET Compact Framework
- DirectX 9.0c End-User Runtime
- XNA Framework 1.0 Refresh
- Speech SDK 5.1¹
- Tablet PC SDK (verzia 1.5 alebo vyššia)
- Microsoft IIS (verzia 5.1 alebo vyššia)

Operačný systém Windows Vista nie je nutnou podmienkou pre beh systému, ale v prípade nižšej verzie nie je možné nainštalovať Windows Vista Sidebar Gadget a je potrebné nainštalovať Speech SDK verzie 5.1, pretože rozpoznávanie hlasu nie je štandardnou súčasťou operačného systému Windows XP a nižších verzii.

Tabuľka B-1. Prehľad prerekvizít v pre jednotlivé ciele nasadenia.

Cieľ nasadenia	Prerekvizity
Klient	.NET Framework 3.0 DirectX 9.0c End-User Runtime XNA Framework 1.0 Refresh Speech SDK 5.11 Tablet PC SDK
PDA	.NET Compact Framework 2.0
Server (služby)	.NET Framework 3.0 Microsoft IIS

B.2 Postup inštalácie

Táto časť popisuje postup inštalácie v závislosti od cieľu nasadenia. Plné nasadenie systému vyžaduje inštaláciu na klientskom počítači, PDA a serveri.

B.2.1 Postup inštalácie na klientovi

Počiatočným krokom nasadenia na klientovi je inštalácia prerekvizít podľa hore uvedeného zoznamu v ľubovoľnom poradí, okrem rámca XNA Framework 1.0 Refresh, ktorého inštalácia musí nasledovať až po inštalácii rámca .NET Framework 3.0 a redistribučnej verzie DirectX 9.0c pre koncových používateľov.

¹ Nie je nutné v prípade nasadenia na operačnom systéme Windows Vista, kde je štandardne nainštalované Speech API 5.3, vrátane schopnosti rozpoznávania hlasu

Po inštalácii prerekvizít sa pomocou inštalačného balíka nainštaluje klient herného sveta s jednotlivými hernými aktivitami a ich editormi a v prípade operačného systému Windows Vista aj aplikácia pre Windows Vista Sidebar.

B.2.2 Postup inštalácie na PDA

Pre spustenie aplikácie na PDA je nutná inštalácia rámca .NET Compact Framework. Následne je možná inštalácia aplikácie, ktorú je možné realizovať skopírovaním a inštaláciou inštalačného balíka aplikácie na žiadané PDA.

B.2.3 Postup inštalácie na serveri

Beh systému na serveri vyžaduje inštaláciu modulu webového servera z balíka služieb Microsoft Internet Information Services a rámca .NET Framework 3.0. Odporúča sa inštalácia v uvedenom poradí, aby bol po inštalácii .NET Framework 3.0 korektne nainštalovaný modul ASP.NET na webový server.

Po inštalácii webového servera je potrebné na server(y) nasadiť dátové a doménové služby systému. Následne sa z inštalačného balíka inštaluje samotný server systému, ktorý okrem spustiteľnej aplikácie servera zahŕňa aj všetky ďalšie závislosti v podobe knižníc a dát.

B.2.4 Konfigurácia na serveri

Na serveri je potrebné nakonfigurovať v súbore app.config servera porty a url na ktorých služby servera počúvajú. Toto sa špecifikuje v elemente baseAddresses pre každú službu zvlášť. Odporúča sa použiť jeden port (predvolene 8080) pre všetky služby hostované v aplikácii servera.

B.2.5 Konfigurácia na klientovi

Konfigurácia SmartLines

V súbore app.config je potrebné nastaviť hodnotu nastavenia SmartPictureClient_PictureStorage_Service na aktuálnu URL webovej služby SmartPictureStorage.

Konfigurácia Strokes

V súbore app.config je potrebné nastaviť:

- V časti klient, pod komentárom „stroke client“ hodnotu address na aktuálnu URL služby Strokes
- V časti klient, pod komentárom „pet client“ hodnotu address na URL služby Zvieratko
- Hodnotu StrokeGame_StrokeStorage_StrokeService v časti ApplicationSetting na aktuálnu URL webovej služby StrokeStorage

Konfigurácia hier Matematické balóny, Zvuky, Opakovanie melódie a Malý hudobník

V súbore app.config je potrebné nastaviť v časti klient hodnotu address na URL služby Zvieratko.

B.3 Napíňanie dátami

Táto kapitola popisuje postup pri napíňaní dátami. Dáta sú dvojitej povahy. Jednak dáta reprezentujúce znalosti konkrétnej, uložené v službách doménových znalostí, a tiež dáta reprezentujúce špecifické problémy, ktoré predstavujú zadania úloh konkrétnych herných aktivít.

B.3.1 Manipulácia s dátami v službách doménových znalostí

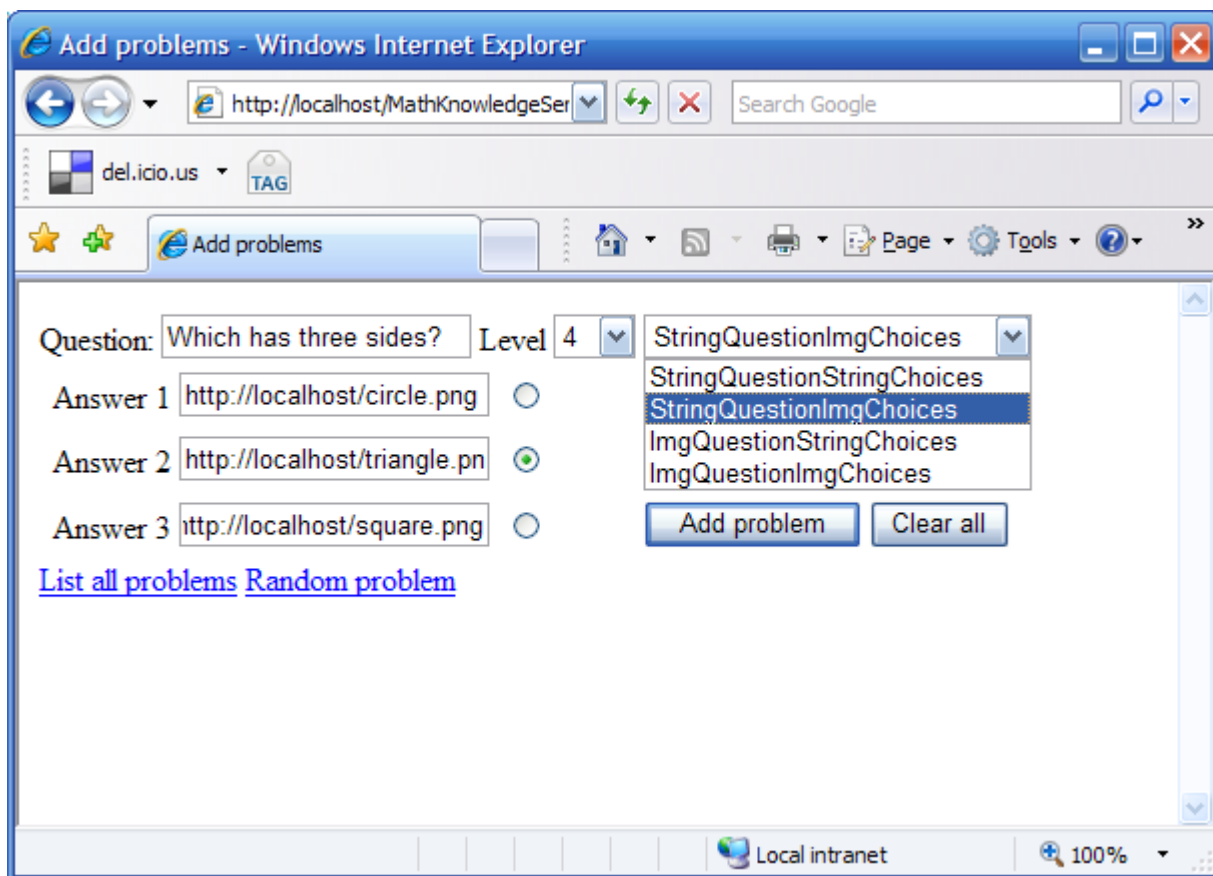
Služby doménových znalostí umožňujú napíňanie dátami prostredníctvom veľmi jednoduchého webového rozhrania dostupného na adrese služby pre konkrétnu doménovú oblasť.

Diagram na obrázku B-1 zobrazuje rozhranie pre pridávanie doménových znalostí pre doménu matematiky.

Znalosti pre doménu matematiky sú reprezentované vo forme otázky s výberom možností. Otázky aj možnosti môžu byť buď text, alebo obrázky.

Pridanie úlohy

1. Určí sa typ úlohy. Je možné zvoliť nasledovné typy:
 - `StringQuestionStringChoices` - otázka aj odpovede predstavujú text
 - `StringQuestionImgChoices` - otázka predstavuje text, odpovede url k obrázku
 - `ImgQuestionStringChoices` - otázka predstavuje url k obrázku, odpovede text
 - `ImgQuestionImgChoices` - otázka aj odpovede predstavujú url k obrázku
2. Typ úlohy sa zvolí výberom z comboboxu (viď obr.).
3. V rozhraní pre pridávanie znalostí sa do textového poľa „Question“ zapíše text otázky alebo url obrázku reprezentujúceho otázku.
4. Do textových polí „Answer 1“ až „Answer 3“ sa uvedú možnosti, vo forme textu alebo url k obrázkom.
5. Zvolením zaškrŕavacieho políčka (radio button) pri správnej odpovedi sa označí jedna z odpovedí za správnu. Výberom v comboboxe „Level“ sa vyšpecifikuje úroveň otázky.
6. Stlačením tlačidla „Add problem“ sa vyplnená otázka pridá do databázy znalostí.



Obrázok B-1. Rozhranie pre pridávanie doménových znalostí z matematiky.

Vypísanie všetkých úloh

Kliknutím na link „List all problems“ sa zobrazia všetky úlohy v danom úložisku. Táto možnosť sa používa len pre testovacie a ladiace účely. Rovnako možnosť „Random problem“ slúži na overenie výberu z úloh.

Mazanie úloh

Zmazanie úloh zo služby doménových znalostí sa vykoná stlačením tlačidla „Clear all“. Vymažú sa tým všetky úlohy uložené v danom úložisku doménových znalostí. Mazanie individuálnych úloh grafické rozhranie neumožňuje, umožňuje to len programové rozhranie služby úložiska doménových znalostí.

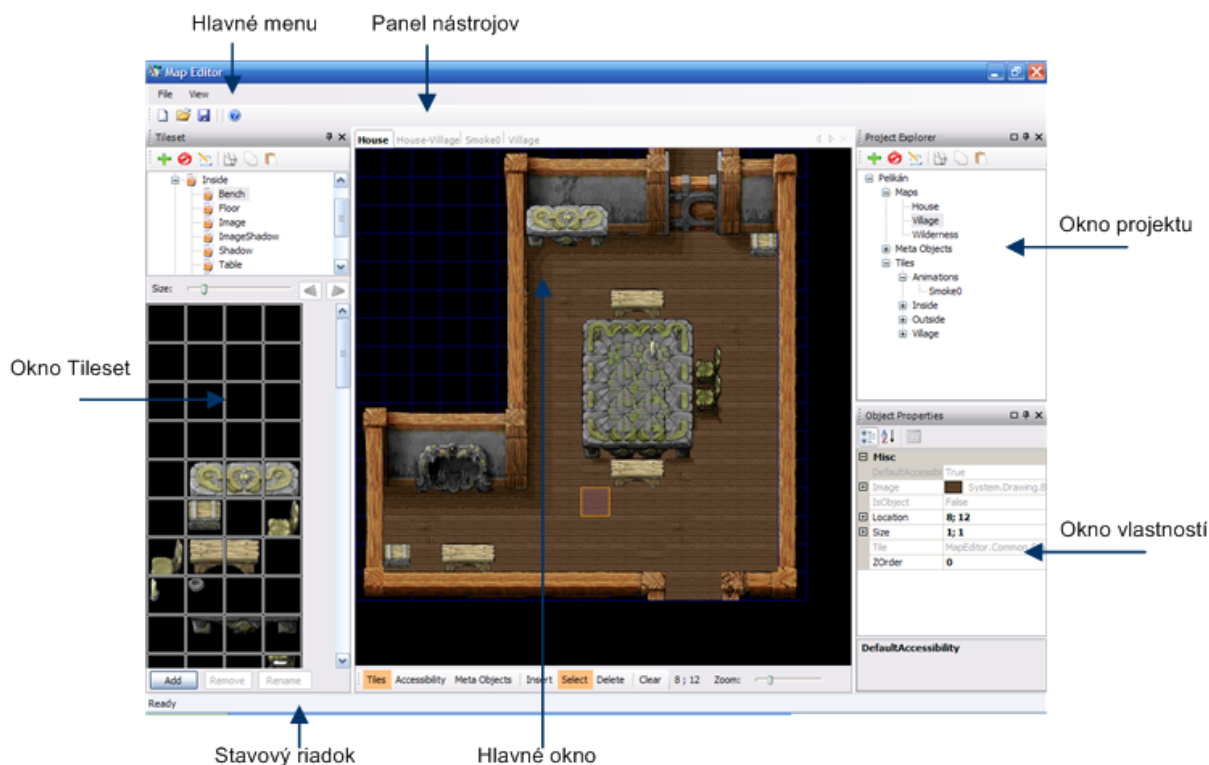
B.3.2 Použitie editorov

Editor Máp

V tejto časti sa nachádza popis práce s editorom máp, ktorý slúži na vytváranie a údržbu herných máp.

Používateľské rozhranie

Používateľské rozhranie editora máp spolu s popisom jeho základných častí sa nachádza na obrázku B-2.



Obrázok B-2. Popis používateľského rozhrania editora máp.

Všetky okná sú dokovateľné, to znamená, že sa dajú ľubovoľne rozmiestniť po obrazovke alebo zadokovať na ľubovoľný okraj. Takisto je možné okno skryť alebo ho prepnúť do vysúvateľného režimu. Nasleduje zoznam a stručný popis okien:

- *Hlavné okno*: Predstavuje hlavnú pracovnú plochu a ako jediné nie je dokovateľné a ani sa nedá vypnúť. Na jeho hornom okraji sa nachádzajú záložky, pomocou ktorých sa dá prepínať medzi jednotlivými otvorenými časťami projektu napríklad mapami alebo meta objektami
- *Okno projektu*: V adresárovej štruktúre zobrazuje jednotlivé časti projektu, ktorými sú mapy, tily a meta objekty a umožňuje pridávanie nových
- *Okno Tileset*: V adresárovej štruktúre zobrazuje všetky tily, pričom v dolnej časti zobrazuje ich obrázky v rámci jedného adresára. Služi pre prácu s mapou
- *Okno vlastností*: Umožňuje nastavovať vlastnosti celému projektu, mapám, políčkam a meta objektom

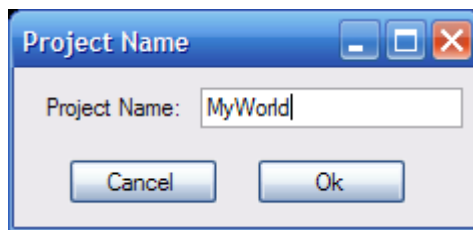
Tvorba nového projektu

Pri práci s editorom je potrebné pred začatím práce explicitne vytvoriť projekt. Vytvorí sa nasledovne:

V hlavnom menu je potrebné kliknúť na položky *File->New* alebo v paneli nástrojov na prvú ikonu zľava

Otvorí sa okno, v ktorom je potrebné špecifikovať miesto kam sa projekt uloží. Projekt si sám vytvára samostatný adresár, ktorého meno je rovnaké ako jeho názov

Zobrazí sa dialóg (vid' obrázok B-3), kam treba vyplniť meno projektu a potvrdiť stlačením tlačidla OK



Obrázok B-3. Dialóg pre zadanie mena projektu.

Načítanie existujúceho projektu

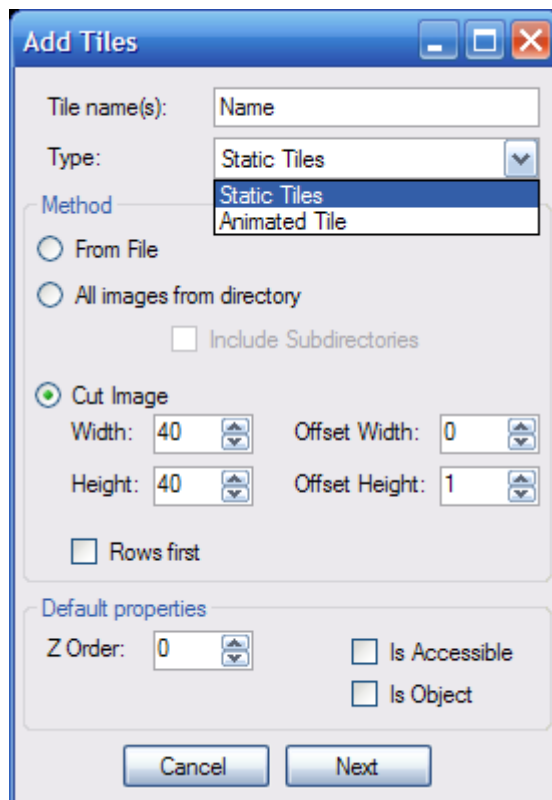
Pre načítanie existujúceho projektu treba v hlavnom menu zvoliť položky *File->Open* alebo v paneli nástrojov kliknúť na druhú ikonu. Objaví sa dialóg pre otvorenie súboru do ktorého treba zadať cestu k projektu.

Práca s editorom

Vytvorenie tilesetu

Na začiatku tvorby sveta je vhodné vytvoriť si tileset, čiže obrázky políček, z ktorých sa mapy budú skladať. Pre tieto účely slúži okno Tileset. Vytvorenie tilesetu sa uskutoční nasledujúcimi krokmi:

1. Pridávanie nových adresárov je možné kliknutím pravého tlačidla na adresár, kam chceme nový adresár umiestniť. Zobrazí sa nový adresár s prednastaveným menom, ktoré je možné hneď zmeniť
2. Pre pridávanie nových políček do aktuálne zvoleného adresára treba kliknúť na tlačidlo *Add* na spodnej časti okna Tileset. Zobrazí sa okno pre vytváranie nových políček (viď obrázok B-4)



Obrázok B-4. Dialóg pre pridávanie nových políček.

V ňom je možné určiť či sa budú načítavať statické alebo animačné políčka a metóda načítavnia, ktorá môže byť:

- Z obrázku (nedá sa použiť pre animačné políčka)
- Z adresára, pričom sa dá zvoliť prehľadávanie do hĺbky alebo len prvá úroveň
- Nasekaním obrázku, pričom sa dá určiť veľkosť a posunie políček

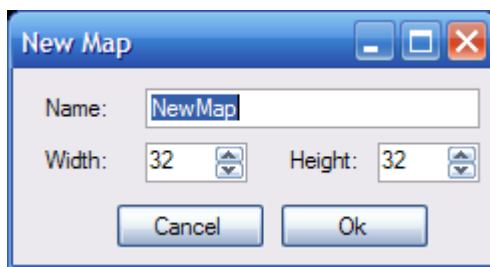
V dialógu je možné špecifikovať aj ďalšie vlastnosti načítaných políček ako priechodnosť alebo poradie vykresľovania.

3. Presúvanie políček (tilov) a adresárov v štruktúre je umožnené štandardnými funkciami *Cut*, *Copy*, *Paste*, ktoré sa nachádzajú v kontextovom menu, ktoré sa dá otvoriť po kliknutí pravým tlačidlom na príslušný objekt. Pomocou tlačidla *Rename* sa dajú políčka premenovávať a tlačidlo *Delete* slúži na ich vymazanie.

Vytváranie máp

Nasleduje popis krokov ktoré naznačujú spôsob vytvárania a práce s mapami:

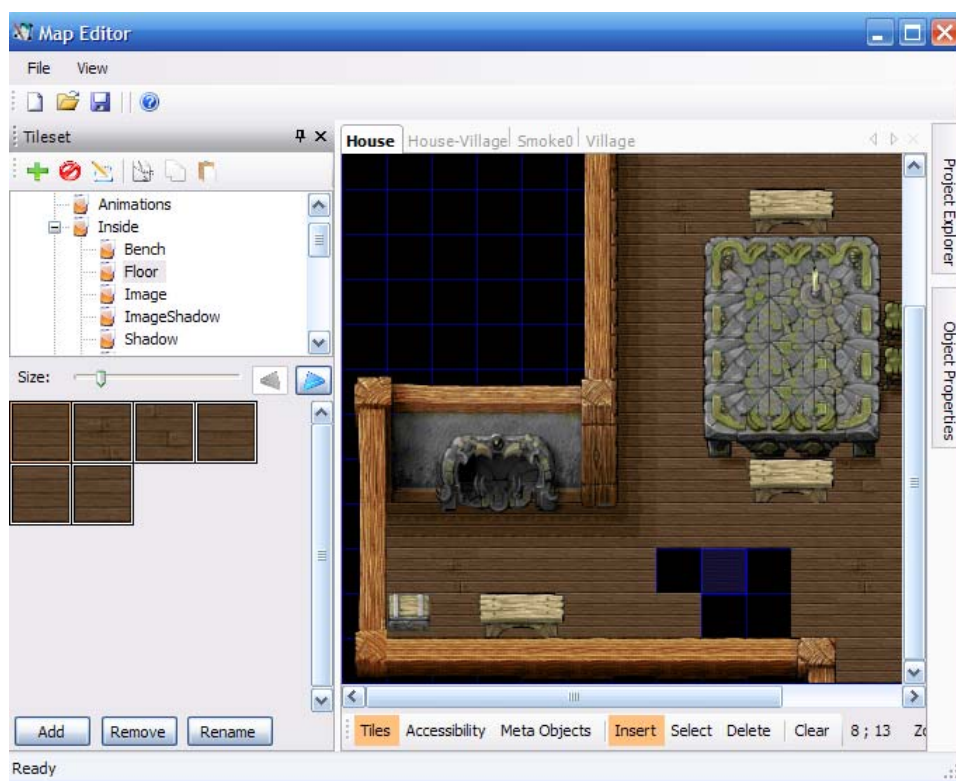
1. Nová mapa sa dá vytvoriť pravým kliknutím na adresár Maps v okne projektu. Zobrazí sa kontextové menu, treba zvoliť položku *Add Map*.
2. Zobrazí sa dialóg pre pridanie mapy (viď obrázok B-5). V ňom treba zadať meno mapy a jej šírku a výšku a potvrdiť stlačením tlačidla *Ok*.



Obrázok B-5. Dialóg pre vytváranie máp.

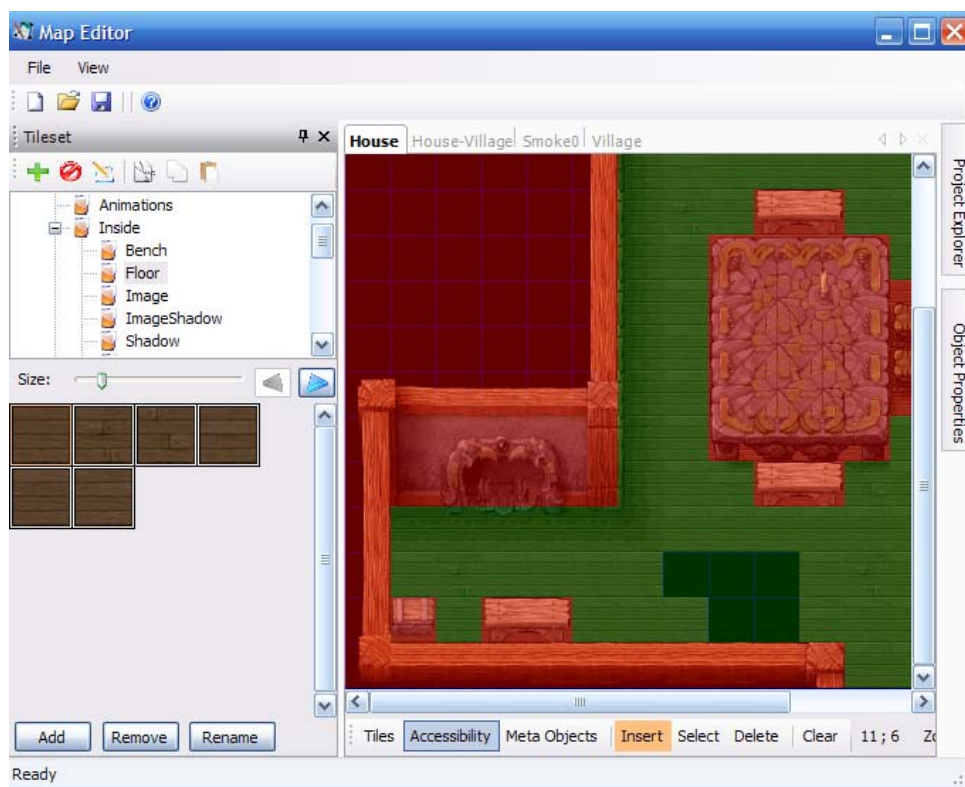
3. V hlavnom okne sa otvorí záložka s novou mapou. Každá mapa má tri vrstvy, medzi ktorými sa je možné prepínať pomocou tlačidiel *Tiles*, *Accessibility* a *MetaObjects* na spodnej časti hlavného okna.
4. Vrstva *Tiles* slúži na pridávanie políček mapy (viď obrázok B-6). Pomocou tlačidiel *Insert*, *Select* a *Delete* na pravej spodnej strane hlavného okna je možné určiť funkciu ľavého kliknutia na políčko mapy:
 - o *Insert*: Pridanie políčka do mapy, pričom toto políčko je možné určiť v okne *Tileset*.
 - o *Select*: Výber políčka, jeho vlastnosti je následne možné meniť v okne vlastností.
 - o *Delete*: Vymazanie aktuálneho políčka.

Editor umožňuje existenciu viacerých políček na jednom mieste, pričom poradie ich vykresľovania je určené hodnotami ich vlastnosti *ZOrder*.



Obrázok B-6. Úprava vrstvy Tiles.

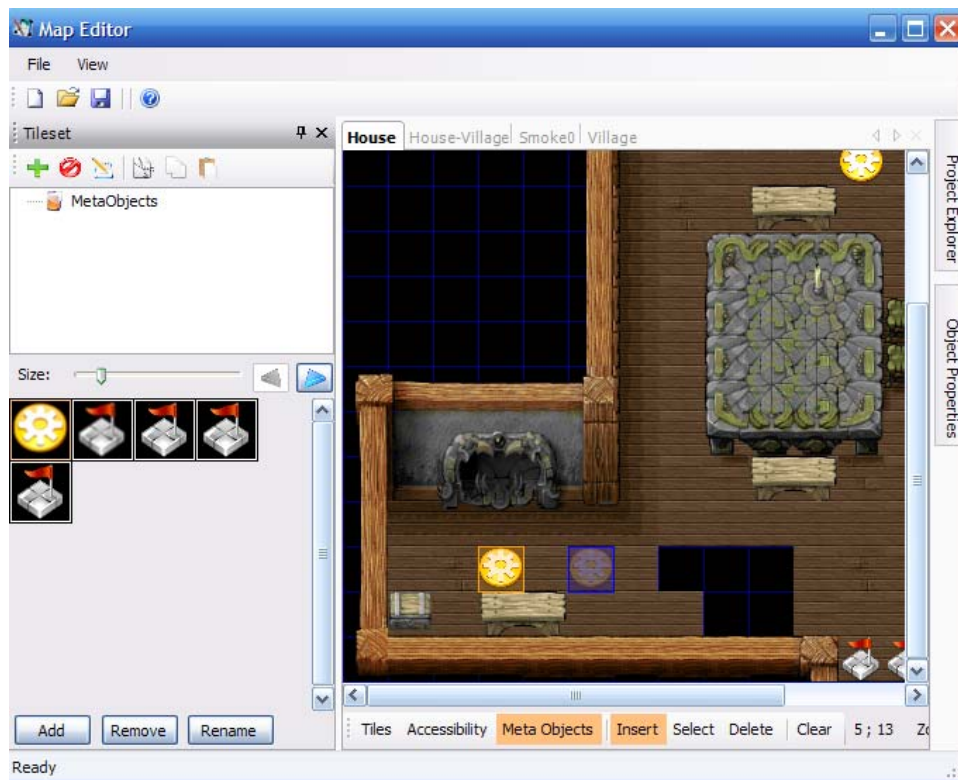
5. Pomocou vrstvy *Accessibility* je možné špecifikovať priechodnosť políček, pričom priechodné políčka sú označené zeleným odtieňom a nepriechodné červeným (viď obrázok B-7).



Obrázok B-7. Vrstva Accessibility.

Všetky políčka sú nastavené podľa ich vlastnosti *DefaultAccessibility*, pokiaľ túto vrstvu používateľ nezmenil. Pomocou tlačidla *Insert* sa nastavujú políčka na priechodné a pomocou tlačidla *Delete* na nepriechodné.

6. Vrstva *MetaObjects* slúži na pridávanie meta objektov, čiže objektov typu *MapTransition*, ktorý špecifikuje prechody medzi mapami a *Spawner*, pomocou ktorého je možné nastavovať vytváranie počítačom riadených postáv (viď obrázok B-8).



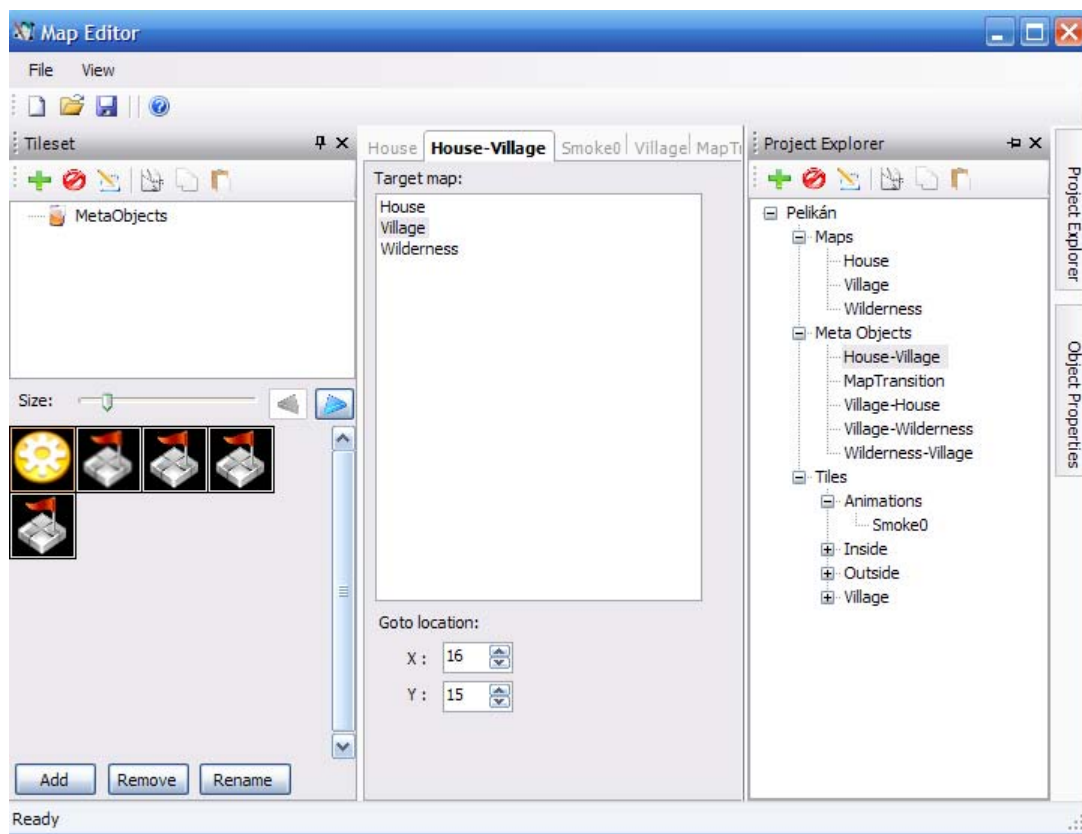
Obrázok B-8. Vrstva *MetaObjects*.

Po prepnutí sa do tejto vrstvy obsah okna *Tileset* zobrazuje namiesto políčok meta objekty definované v projekte. Pomocou tlačidla *Insert* možno objekty do mapy vkladať, tlačidlom *Select* je umožnené objekty vyberať a meniť ich vlastnosti v okne vlastností a tlačidlo *Delete* slúži na zapnutie režimu mazania.

Vytváranie prechodov medzi mapami

Pre vytvorenie prechodu je potrebné, aby projekt obsahoval aspoň dve mapy. Treba vykonať tieto kroky:

1. Vytvoriť prechod pravým kliknutím na adresár *MetaObject* a zvolením položky *Add Map Transition*.
2. V hlavnom okne sa zobrazí záložka (viď obrázok B-9), kde je potrebné nastaviť cieľovú mapu ako aj pozíciu, kde sa hráč na cieľovej mape objaví.
3. Vo vrstve *MetaObjects* treba pridať vytvorený prechod na miesto, na ktorom sa má prechod realizovať.



Obrázok B-9. Nastavovanie prechodov medzi mapami.

Uloženie projektu

Projekt je možné uložiť voľbou *File->Save* v hlavnom menu alebo v paneli nástrojov tretím tlačidlo zľava. Informácie o projekte sa zapíšu do súboru s príponou wpr a obrázky políček sa uložia ako obrázky typu png do adresára *Tiles* spolu so svojou adresárovou štruktúrou.

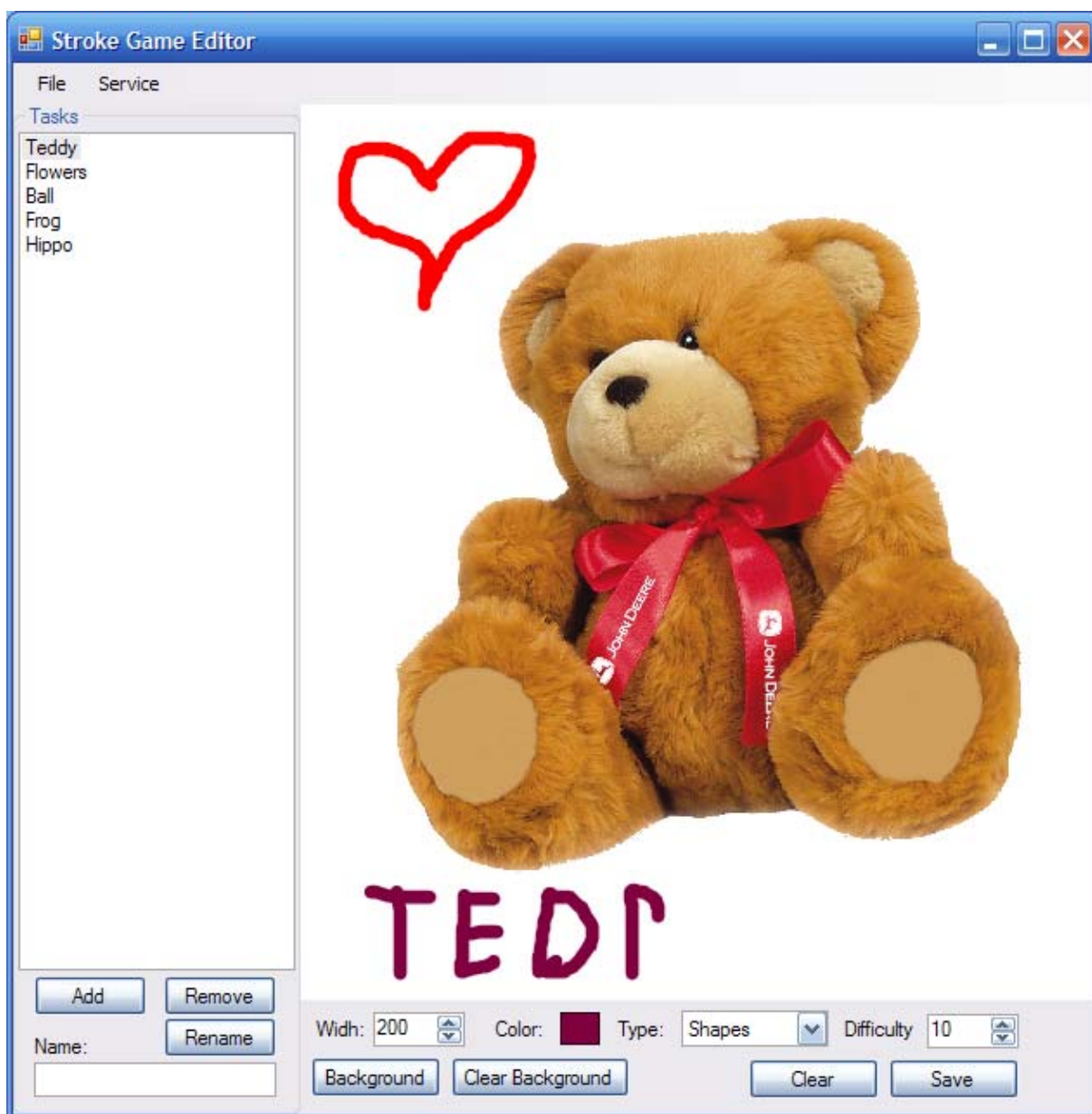
Ukončenie práce

Prácu s editorom máp je možné ukončiť voľbou *File->Exit* v hlavnom menu. Všetky neuložené informácie sa stratia.

Editor hry Strokes

Po spustení editora sa objaví jeho hlavná obrazovka, ktorá je zobrazená na obrázku č. B-10. Pozostáva z týchto hlavných častí:

- Panel naľavo, kde je zobrazený zoznam vytvorených úloh a tlačidlá na pridávanie novej úlohy a premenovanie a vymazanie existujúcej.
- Panel dole slúži na nastavenie farby a hrúbky pera, nastavenie obrázku v pozadí a nastavenie parametrov úlohy, ktorými sú náročnosť a typ.
- Časť v strede je určená na kreslenie vzorových ťahov perom so zvolenou farbou a hrúbkou.



Obrázok B-10. Editor úloh pre hru Strokes.

Ukladanie a načítavanie súborov na disk

Pre tieto účely slúži položka *File*, nachádzajúca sa v hlavnom menu. Umožňuje:

- Vytvorenie novej sady úloh pomocou položky *New*
- Uloženie existujúcej sady do súboru na disk pomocou položky *Save*
- Načítanie existujúcej sady z disku výberom položky *Open*

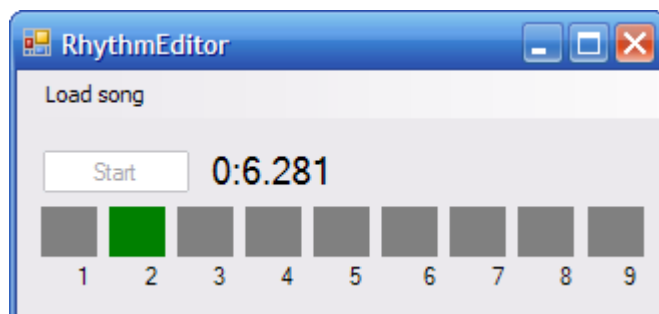
Ukladanie a načítavanie úloh na server

Táto funkcionality je obsiahnutá v položke *Service* v hlavnom menu. Umožňuje:

- Načítanie aktuálnych úloh služby voľbou položky *Get Tasks*
- Uloženie nových úloh voľbou *Send New Tasks*

Editor hry Malý hudobník

Na obrázku B-11 sa nachádza obrazovka editora. Pomocou voľby *LoadSong* v hlavnom menu je možné načítať skladbu, ktorú je potrebné následne pustiť tlačidlom *Start*. Počas prehrávania hudby sa jednotlivé noty zadávajú rovnako ako pri samotnej hre, pričom práve držané rady sú vyznačené zelenou farbou. Po ukončení prehrávania editor uloží xml súbor opisujúci úlohu s menom skladby do rovnakého adresára, z akého sa skladba načítala.



Obrázok B-11. Editor úloh hry Malý hudobník

Formát úloh pre hru Zvuky

Úlohy sa dajú pridávať bez editora pridaním potrebných súborov do adresára s obrázkami a adresára so zvukmi. Meno zvuku musí byť rovnaké ako meno obrázku. Je možné pridať aj obrázok v neaktívnom móde, ten sa musí nachádzať v obrázkovom adresári a mať meno v tvare *NázovObrázka_small*.

Formát úloh pre hru Opakovanie melódie

Táto hra nevyžaduje žiadny špeciálny formát, pretože náhodne vyberá mp3 súbory z adresára *Sounds*. Pri pridávaní nových skladieb treba počítať s obmedzeniami rozoznávania výšky tónov, nakoľko ten funguje len pre zreteľné jednoglasné melódie.

Editor hry SmartLines

Obrazovka editora hry (viď obrázok B-12) Smart Lines zobrazeného na obrázku pozostáva z dvoch častí:

- *Riadiaca lišta*: Lišta v ktorej používateľ zadáva všetky riadiace povely pre editor Smart Lines
- *Zobrazovacia a editačná oblasť*: Oblasť, v ktorej je zobrazený obrázok a v ktorej sa dá pridávať nové body a vytvárať tak čiaru.



Obrázok B-12. Editor hry SmartLines

Načítanie obrázku z lokálneho disku

V prípade že používateľ chce načítať obrázok z lokálneho disku, musí vykonať nasledujúcu sériu krokov:

1. Stlačiť tlačidlo *Local* na riadiacej lište
2. Zvoliť názov súboru s obrázkom

Načítanie obrázku z webu

Ak chce používateľ pracovať s obrázkom, ktorý sa nachádza na webe, potom vykonáva nasledujúcu postupnosť:

1. Stlačiť tlačidlo *Web* na riadiacej lište
2. Vložiť URL obrázku
3. Stlačiť tlačidlo *Go*

Editácia obrázku

Používateľ načítaný obrázok môže editovať tak, že naň kliká, čím vytvára čiaru.

Uloženie obrázku

Procedúra uloženia pozostáva z nasledujúcich krokov:

1. Používateľ obrázok pomenuje v poli s vyznačeným obsahom *<identifier>*
2. Kliknutím na tlačidlo *Upload* sa obrázok uloží, a je automaticky prístupný v systéme

C Obsah média

Dodatok slúži ako opis obsahu média.

Tabuľka C- 1. Opis obsahu digitálneho média

Fyzická cesta	Popis
\Prerequisites	Prerekvizity potrebné pre nasadenie a chod systému
\Deployment	Skompilované časti systému určené na nasadenie
\Deployment\Client	Klientská časť systému určená na nasadenie
\Deployment\Client\PelikanClient	Klient herného sveta
\Deployment\Client\Pet.Application.WPF	Zvieratko spievádzajúce hráča pri vykonávaní herných aktivít
\Deployment\Client\Pet.Gadget	Windows Vista Sidebar Gadget
\Deployment\Server	Serverová časť systému určená na nasadenie
\Deployment\Services	Dátové a doménové služby
\Source	Zdrojové kódy projektu
\Documentation	Projektová dokumentácia
\Presentation	Prezentácia použitá na národnom finále súťaže Imagine Cup 2007
\Page	Statická verzia stránky