



Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 3, 812 19 Bratislava



BLOODLESS
KANDIDÁT NA NAJLEPŠÍ MULTIMEDIÁLNY PRODUKT
ROKU 2008
(Tímový projekt)



Tím číslo: 4
Vedúci tímu: Mgr. Alena Kovárová
Odbor: Softvérové inžinierstvo
Šk. rok: 2007/2008

Bc. Sašo Kiselkov
Bc. Bianka Kováčová
Bc. Martin Kozmon
Bc. Lenka Litvová
Bc. Michal Poláčik
Bc. Jakub Tekel'

Obsah

OBSAH	I
ÚVOD	III
SLOVNÍK POJMOV A POUŽITÉ SKRATKY	IV
ZOZNAM TABULIEK A OBRÁZKOV	V
1 OPIS RIEŠENIA	1
1.1 EUROPRIX	1
1.2 NÁMETY.....	1
1.3 NÁVRH PROJEKTU	2
2 ANALÝZA	3
2.1 HRY S KOMÁROM AKO HLAVNOU POSTAVOU	3
2.1.1 <i>Mister Mosquito</i>	3
2.1.2 <i>Mosquito Ops</i>	4
2.1.3 <i>Prínos nášho projektu</i>	5
2.2 ANALÝZA MOŽNOSTÍ POUŽÍVATEĽSKEJ INTERAKCIE	5
2.2.1 <i>Ovládanie hlasom</i>	5
2.2.2 <i>Ovládanie zvukom</i>	9
2.2.3 <i>Ovládanie pohybom</i>	10
2.3 ANALÝZA SOFTVÉROVÝCH PROSTRIEDKOV	11
2.4 ZHODNOTENIE ANALÝZY	12
3 ŠPECIFIKÁCIA RIEŠENIA	13
3.1 NÁVRH KONCEPCIE HRY	13
3.1.1 <i>Hlavná myšlienka</i>	13
3.1.2 <i>Motivácia</i>	13
3.1.3 <i>Svet</i>	14
3.1.4 <i>Nepriatelia</i>	14
3.1.5 <i>Vybavenie</i>	14
3.1.6 <i>Priebeh a cieľ hry</i>	15
3.2 POUŽÍVATEĽSKÁ INTERAKCIA	16
4 HRUBÝ NÁVRH RIEŠENIA	17
4.1 ARCHITEKTÚRA SYSTÉMU	17
4.1.1 <i>Modul Interface</i>	18
4.1.2 <i>Modul 3DEngine</i>	19
4.1.3 <i>Modul GameEngine</i>	19
4.1.4 <i>Modul GraphicsDesign</i>	21
4.2 NÁVRH SVETA	22
4.2.1 <i>Návrh príbehu</i>	22
4.2.2 <i>Návrh postáv</i>	23
4.2.3 <i>Návrh grafického používateľského rozhrania</i>	24
5 PROTOTYPOVANIE	26
5.1 CIELE PROTOTYPOVANIA.....	26
5.2 ANALÝZA LETU KOMÁRA	26
5.2.1 <i>Sily pôsobiace na komára</i>	26
5.2.2 <i>Rýchlosť komára</i>	28
5.2.3 <i>Určenie pozície komára v priestore</i>	28
5.2.4 <i>Zatáčanie komára počas letu</i>	29
5.2.5 <i>Naklonenie komára počas zatáčania</i>	29
5.2.6 <i>Ovládanie komára používateľom</i>	30
5.2.7 <i>Aplikácia analýzy v prototype</i>	30
5.3 PROTOTYPOVANIE 3D MODELOVANIA.....	31

BloodLess

Kapitola 0 - Obsah

5.3.1	<i>Prototypovanie komára</i>	31
5.3.2	<i>Prototypovanie iných 3D objektov</i>	32
5.4	PROTOTYPOVANIE GRAFICKÉHO SVETA.....	33
5.5	URČOVANIE FREKVENCIE ZVUKU	34
5.6	VÝSLEDKY PROTOTYPOVANIA	36
5.6.1	<i>Realistický pohyb komára</i>	36
5.6.2	<i>3D modelovanie</i>	36
5.6.3	<i>Grafický svet</i>	36
5.6.4	<i>Určovanie frekvencie zvuku</i>	36
6	POUŽITÁ LITERATÚRA	37

Úvod

Každý rok prebieha medzinárodná súťaž The EUROPRIX Top Talent Award (TTA). Je to súťaž multimediálnych produktov, ktoré vytvorili mladí ľudia, sú niečím nové, neobyčajné. Ide skrátka o to, vytvoriť niečo, čo tu ešte nebolo a dokáže to uchvátiť. Cieľom projektu je navrhnúť a vytvoriť produkt, ktorý by bol dostatočne dobrý na to, aby sa uchádzal o nomináciu alebo dokonca o výhru v súťaži TTA08. Ešte pred návrhom je nevyhnutné spraviť si obraz o tom, aké produkty sa tejto súťaže zúčastňovali po minulé roky, vybrať kategóriu, v ktorej máme šancu uplatniť sa, či už preto, že v tíme je človek, ktorý ovláda nové technológie alebo preto, že daná kategória bola slabo zastúpená.

Fantázii sa medze nekladú, programovať môžete v čomkoľvek, podstatné je, aby to uchvátilo - bolo niečím nové, dobre fungovalo, dobre vyzeralo (preto v tíme musí byť aj človek, ktorý má cit pre dizajn), malo aj plne funkčnú ANGLICKÚ verziu!

Viac sa o súťaži dočítate na: <http://www.toptalent.europrix.org/tta07/>

Dokument je rozdelený na päť hlavných častí.

Kapitola 0 – Úvod predstavuje úvod do čítania dokumentu. Obsahuje zadanie projektu, prehľad dokumentu, slovník pojmov a skratiek a zoznam tabuliek a obrázkov. Mala by byť prvou kapitolou, ktorú si čitateľ prečíta.

Kapitola 1 – Opis riešenia opisuje konkretizáciu zadania, ktoré bolo zadané len všeobecne. Opisuje aj kontext zadania, jeho ciele a námety, ktoré pri vymýšľaní vznikli.

Kapitola 2 – Analýza analyzuje podobné existujúce riešenia a nástroje, ktoré by mohli byť využité v našom projekte.

Kapitola 3 – Špecifikácia riešenia vychádza z analýzy a špecifikuje výsledný produkt. Opisuje základný princíp hry, jej koncepciu a používateľskú interakciu.

Kapitola 4 – Hrubý návrh opisuje návrh špecifikovaného systému. Navrhne jeho architektúru a základné grafické prvky.

Kapitola 5 – Prototypovanie opisuje najdôležitejšie časti aplikácie a snahu o ich prototypovanie.

Slovník pojmov a použité skratky

Zoznam použitých skratiek:

- **GLUT** - OpenGL Utility Toolkit, knižnica služieb pre OpenGL počítačové programy
- **GPS** - Global Positioning System - satelitný navigačný systém používaný na zistenie presnej pozície
- **GUI** - Graphical User Interface, grafické používateľské rozhranie
- **NPC** - Non-player character, postava v hre, ktorá nie je kontrolovaná používateľom
- **OpenAL** – Open Audio Library, audio aplikačné programové rozhranie
- **OpenGL** - Open Graphics Library – knižnica, ktorá slúži na tvorbu aplikácií pracujúcich predovšetkým s trojrozmernou počítačovou grafikou prekresľovanou v reálnom čase
- **PC** - Personal Computer, osobný počítač
- **PS** – PlayStation, televízna hracia konzola
- **SCEI** - Sony Computer Entertainment, Incorporated, japonská firma špecializujúca sa na video hry
- **TTA** - Top Talent Award – európska súťaž inovatívnych projektov

Slovník pojmov.

- **6DOF** - Six Degrees Of Freedom – vzťahuje sa na pohyb v 3D priestore, konkrétne na pohyb dopredu/dozadu, hore/dole, vľavo/vpravo kombinovaný s rotáciou okolo troch osí.
- **Bullet-time** - Počítačová simulácia spomalenia času, ktorá umožňuje zachytiť inak nepostrehnuteľné udalosti, ako napr. letiacu guľku
- **ColDet** - Knižnica na 3D detekciu kolízií
- **Microsoft SAPI** - Microsoft Speech API – rozhranie na syntézu a rozpoznávanie hlasu pre Windows aplikácie

Zoznam tabuliek a obrázkov

Zoznam tabuliek v dokumente:

Tab. 1. Hry s komárom.	3
Tab. 2. 3D grafické a herné systémy spolupracujúce s C++.....	12
Tab. 3. 3D grafické a herné systémy spolupracujúce s Javou.	12
Tab. 4. Komponenty modulu 3D engine.	19
Tab. 5. Komponenty modulu GraphicsDesign.	21

Zoznam obrázkov v dokumente:

Obr. 1. Hra Mister Mosquito. [12]	4
Obr. 2. Hra Mosquito Ops. [13]	4
Obr. 3. Hra LifeLine. [2]	7
Obr. 4. PiLfluS! [3]	8
Obr. 5. Hra Pah! [4]	9
Obr. 6. Hra Ford Game. [6]	10
Obr. 7. Hra Sing Pong. [7].....	10
Obr. 8. Architektúra aplikácie.	17
Obr. 9. Schéma herného sveta.	22
Obr. 10. Character design hlavnej postavy komára.....	23
Obr. 11. Character design NPC postavy.....	24
Obr. 12. Návrh GUI.....	25
Obr. 13. Aerodynamické sily pôsobiace na letiaci objekt (pohľad z boku).	26
Obr. 14. Sily pôsobiace na komára (pohľad z boku)......	27
Obr. 15. Sily pôsobiace na teleso idúce po naklonenej zákrute.	30
Obr. 16. Pôvodný model komára.....	32
Obr. 17. Jednoduchý model komára určený na prototypovanie.	32
Obr. 18. Prototyp budovy do prostredia.	33
Obr. 19. Diagram tried pre prototyp identifikácie frekvencie.	34
Obr. 20. Diagram činnosti pre prototyp identifikácie frekvencie.....	35

1 Opis riešenia

1.1 *Europrix*

Europrix Top Talent Award je európska multimedialná súťaž pre študentov a mladých profesionálov, ktorí pracujú na inovatívnych projektoch v oblasti e-obsahu a návrhu. Neexistujú žiadne obmedzenia pre použité multimédia a platformy. Súťaží sa v ôsmich kategóriách.

Kľúčom pre úspech v tejto súťaži je inovatívnosť a kreativita. Prináša veľkú voľnosť vo výbere témy aj technológie, takže účastníkom nič nebráni v plnom rozvíaní ich fantázie. Cieľom nie je vytvoriť reálne použiteľný produkt, ale produkt, ktorý by prekvapil a priniesol niečo nové. V posledných ročníkoch súťaže sa napr. osvedčili nové prístupy k vstupu alebo výstupu dát do alebo z produktu.

Preto aj táto téma zaujala celý náš tím hneď od začiatku, lebo predstavovala možnosť vytvoriť zaujímavý multimedialný softvérový systém a predstaviť ho aj na európskej úrovni. Naším cieľom ako budúcich inžinierov je práve vytváranie takýchto systémov rôzneho zamerania, pričom súťaž Europrix TTA ponecháva dostatok priestoru pre výber konkrétneho zamerania projektu.

1.2 *Námety*

K celkovej predstave o podstate tejto témy pomohlo preštudovanie minuloročných projektov nielen študentov našej fakulty, ktoré sa súťaže zúčastnili či dokonca boli úspešné a získali známku kvality. Pochopili sme, že základným predpokladom na úspech v súťaži je vymyslieť a vytvoriť niečo nové, využiť najnovšie poznatky a technológie, preto neodmysliteľnou súčasťou zvolenej témy je mať výborný nápad na projekt. Aj projekty v tej istej kategórii boli väčšinou úplne iné spájalo ich len zameranie na určitú technológiu, čo potvrdilo našu prvotnú domnienku, že neexistujú hranice obmedzujúce nás v invenčnosti.

Naším prvotným nápadom bolo vytvorenie virtuálnej reality, kde by si používatelia mohli vymieňať informácie. Virtuálnu realitu by formovali samotní používatelia a vkladali do nej svoje vedomosti, pričom by mohli aj prezerat' vedomosti ostatných. Ovládanie sveta by bolo realizované hlasom.

Tento nápad však nebol dostatočne inovatívny a vhodný pre súťaž typu Europrix, takže sme sa ďalej zamýšľali a vymysleli nasledujúce námety:

- Svet príšer. Mohlo by sa jednať o nástroj, ktorý umožní takéto príšery vytvárať, alebo o hru, v ktorej sa treba s príšerou spriatelieť. Hráč by hru ovládal pomocou hlasu a pohyby jeho rúk by zaznamenávala kamera.
- Svet bez obrazu, v ktorom by sa musel používateľ orientovať pomocou sluchu. Keďže steny nevydávajú zvuky, ich polohu by mohol odhaliť detektor so zvukovým výstupom.
- Simulátor komára. Používateľ by ovládal komára v trojrozmernom svete pomocou bzučania a počítačovej myši. Výška hlasu predstavuje rýchlosť mávania krídlami, poloha myši určuje smer letu. Úlohou hráča by bolo zberať krv a nosiť ju do nemocnice. Okrem toho by hra mala aj ďalšie úlohy. Pri plnení cieľu si hráč musí dať pozor na lastovičky a ďalších nepriateľov.

- Systém umožňujúci snímanie predmetov pomocou kamery alebo fotoaparátu. Následne by sa tieto predmety vložili do počítaču, kde by ich bolo možné presúvať v priestore. Program by sa dal využiť na zariadenie izby.
- Hra, v ktorej by bolo možné prenášať zvuky medzi predmetmi. Predmet by bolo možné využiť jedine vtedy, ak má pridelený správny zvuk. Keďže počet zvukov by bol nižší ako počet predmetov, úlohou hráča by bolo vhodne presúvať zvuky, aby sa dostal do ďalších častí sveta.

1.3 Návrh projektu

Zhodli sme sa, že z uvedených námetov bude najvhodnejší simulátor komára. Inovatívny bude v spôsobe ovládania, keďže chceme zahrnúť aj ovládanie hlasom alebo frekvenciou hlasu. Tiež hlavný zámer hry, kedy komár bude plniť humanitné misie a zbierať krv pre nemocnicu, nie je zvyčajný. Priblížením života komára a rozlišovaním krvných skupín chceme dať hre aj náučný charakter.

2 Analýza

2.1 Hry s komárom ako hlavnou postavou

Analyzovali sme dve trojrozmerné počítačové hry, v ktorých hráč ovládal komára: Mister Mosquito a Mosquito Ops (pozri tab. 1).

Tab. 1. Hry s komárom.

Názov hry	Platforma	Použité knižnice	Licencia
Mister Mosquito [11]	Playstation 2		komerčná
Mosquito Ops [13]	PC, Mac	OpenGL, GLUT, FreeImage, OpenAL, ColDet and GLEW	2-semestrový školský projekt, freeware

2.1.1 Mister Mosquito

Dej hry Mister Mosquito sa odohráva v byte. Cieľom hry je nazbierať zásoby krvi na prežitie počas zimného obdobia. Hráč saje krv členom rodiny, zatiaľ čo sa venujú svojim každodenným činnostiam. Krv je možné sať len z určitých miest na tele, ktoré sú prístupné len v určitých časoch. Každý člen rodiny pravidelne opakuje rovnaký sled činností a pohybov. Hráč sleduje tieto pohyby, aby mohol úspešne vysať dostatočné množstvo krvi pre postup do nasledujúcej úrovni hry.

Rýchlosťou cicania krvi hráč ovplyvňuje mieru stresu svojej obete. Ak táto dosiahne kritickú hodnotu, komár umiera pod rukou svojej obete. Ak človek spozoruje letiaceho komára, hra sa prepne do bojového módu. Vtedy sa človek snaží zabiť komára vlastnými rukami, sprejmi alebo odpudzovačmi. Hráč musí svoju obeť upokojiť zatlačením na konkrétne miesta na tele človeka, čím uvoľní jeho napätie.

Komár nachádza v byte rôzne predmety, ktoré zvyšujú jeho zdravie, maximálnu hodnotu zdravia, zásobník krvi alebo zobrazujú rôzne informácie o hre. Hráč môže ovládať rôzne predmety ako mobil, rádio alebo televízor stláčaním ovládacích tlačidiel.

Hra má kreslenú grafiku (obr. 1).



Obr. 1. Hra Mister Mosquito. [12]

2.1.2 Mosquito Ops

V hre Mosquito Ops (obr. 2) je úlohou hráča plnenie úloh ako satie ľudskej krvi, hľadanie stratených vajíčok a kradmý pohyb v oblastiach chránených proti komárom. Na komára číhajú rôzne nástrahy: ľudia, elektrické odpudzovače, lampy a ventilátory.

Hra ponúka rôzne módy kamery vrátane röntgenovej. Satie krvi prebieha v bullet-time móde.



Obr. 2. Hra Mosquito Ops. [13]

2.1.3 Prínos nášho projektu

Nami navrhovaná hra prinesie viaceré nové rozmery:

- humánný hlavný dejový motív, ktorým je dopĺňanie krvi do nemocnice
- náučný charakter vďaka potrebe oboznámiť sa s klasifikáciou krvných skupín
- zvukové ovládanie komára
- boj so zvieratami
- používanie rôznych vylepšení, ktoré hráč buď nájde alebo si ich kúpi za body

2.2 Analýza možností používateľskej interakcie

Navrhovaná hra by nemala používateľa zaujať len z grafickej stránky, ale mala by mu umožniť aj vyskúšať si neštandardné formy interakcie. Klasické ovládanie pomocou myši a klávesnice je praktické, keďže používateľ je už naň zvyknutý a teda ho používa s väčšou istotou a rýchlosťou. Tento fakt je dôležitý najmä pri každodennej práci, kedy sme vďační, keď úloha nám zverená je čo najjednoduchšia. Hry však tvoria zvláštnu kapitolu, cieľom hráča je zvíťaziť. Hra, v ktorej prekážky prekonávame príliš jednoducho, nás rýchlo prestane baviť. A podobne je to s ovládaním – to síce musí byť používateľsky príjemné, ale na druhej strane nemusíme byť v ňom až takí zbehlí, aby sme hneď od začiatku podávali vrcholné výkony. Hráčovi prináša uspokojenie nielen samotné prekonávanie prekážok vo virtuálnom svete, ale aj fakt, že ich postupne napriek narastajúcej zložitosti zvláda jednoducho.

V navrhovanej aplikácii bude potrebné ovládať:

- smer letu komára
- výšku a rýchlosť letu komára
- akcie komára
- pohyb v menu aplikácie

Okrem klasického ovládania pomocou klávesnice a myši existuje viacero alternatív, v súčasnej dobe využívaných najmä ľuďmi, ktorí nemôžu využívať tie klasické zo zdravotných dôvodov. V ďalšom sa budeme venovať trom alternatívnym ovládaniam:

- ovládanie hlasom (pomocou rozpoznania hlasových povelov)
- ovládanie pomocou zvuku (bez jeho rozpoznania)
- ovládanie pohybom – a to buď pohybom očí, hlavy alebo celého tela

2.2.1 Ovládanie hlasom

Ovládanie hlasom sa stáva čoraz populárnejšie. Programy na rozpoznávanie hlasu sú vyvíjané už vyše 40 rokov, ale prakticky sa začali používať až v poslednom čase. Ako dobrý príklad slúži zabudované ovládanie pomocou hlasových povelov v operačnom systéme Windows Vista, či častá prítomnosť hlasového rozpoznávania v mobilných telefónoch. Podľa [1] je hlavným dôvodom doterajšieho neúspechu tohto ovládania chybovosť rozpoznania hlasu. Samotné rozpoznanie je často komplikované nárečím, prízvukom, hlukom z okolia a emóciami. Za posledné štyri roky však došlo k priemerne 25 percentnému zníženiu chybovosti ročne. Rozpoznávanie hlasu však stále má dosť veľké nedostatky. Rozpoznávanie

súvislej reči či reči v hlučným prostrediach je stále veľmi problematické, v porovnaní s dosť presným rozpoznávaním čísel a hlasových povelov.

Hláska vyslovená rôznymi ľuďmi má rôznu frekvenciu, a teda nie je možné pre jej identifikovanie stanoviť len jednu, preto sú pre rozpoznávanie reči najčastejšie využívané skryté Markove modely. Pre úspešné rozpoznanie sa musí vyslovená hláska nachádzať v danom rozmedzí frekvencií. Modely sú tvorené pomocou veľkých databáz zvukov, ktoré zahŕňajú rôzne výslovnosti. Okrem zvukov obsahujú aj štatistické dáta o ich zvyčajnej kombinácii. Pri pokuse o rozpoznanie počítač rozdelí prijatý zvuk na jednotlivé časti a skúma, či susediace časti súhlasia s jeho doterajšími vedomosťami. Tieto vedomosti je možno vytvárať aj tréningom daného softvéru, v ktorom je analyzovaná reč konkrétneho používateľa [1].

Softvér využívajúci hlasové rozpoznávanie možno rozdeliť do dvoch kategórií:

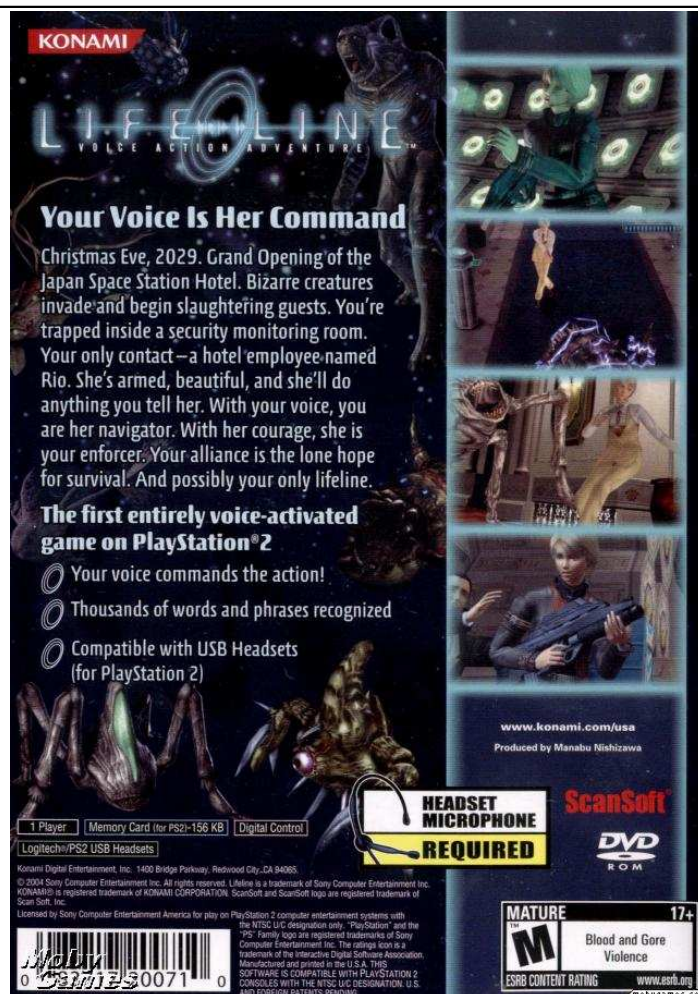
- softvér určený na konkrétnu činnosť s priamo zabudovaným hlasovým ovládaním
- softvér umožňujúci hlasovo ovládať iné aplikácie

V oblasti počítačových hier platí podobné delenie:

- hry s priamo zabudovaným ovládaním hlasovými povelmi – príkladom je hra LifeLine, v Japonsku mala táto hra (vydaná firmami SCEI and Konami v roku 2003) názov „The Operator’s Side“. Ide o hru určenú pre platformu PS2 plne ovládanú hlasom (obr. 3). Hra dokáže rozpoznať 5 000 slov a 100 000 fráz. Je založená na princípe, že hráč hovorí svojím postavám, čo majú robiť. Výhodou hry je aj rozvinutá umelá inteligencia. Zámer tvorcov bol zjednodušiť ovládanie pre hráčov-nováčikov, ale byť zaujímavým aj pre skúsených hráčov, ktorí chcú vyskúšať niečo nové. Príbeh hry sa odohráva v budúcnosti na vesmírnej stanici, ktorú ohrozujú obludy. Hráč ju hrá v tretej osobe, keď svojej postave dáva príkazy alebo otázky, na ktoré mu ona odpovedá. [2]

BloodLess

Kapitola 2 - Analýza



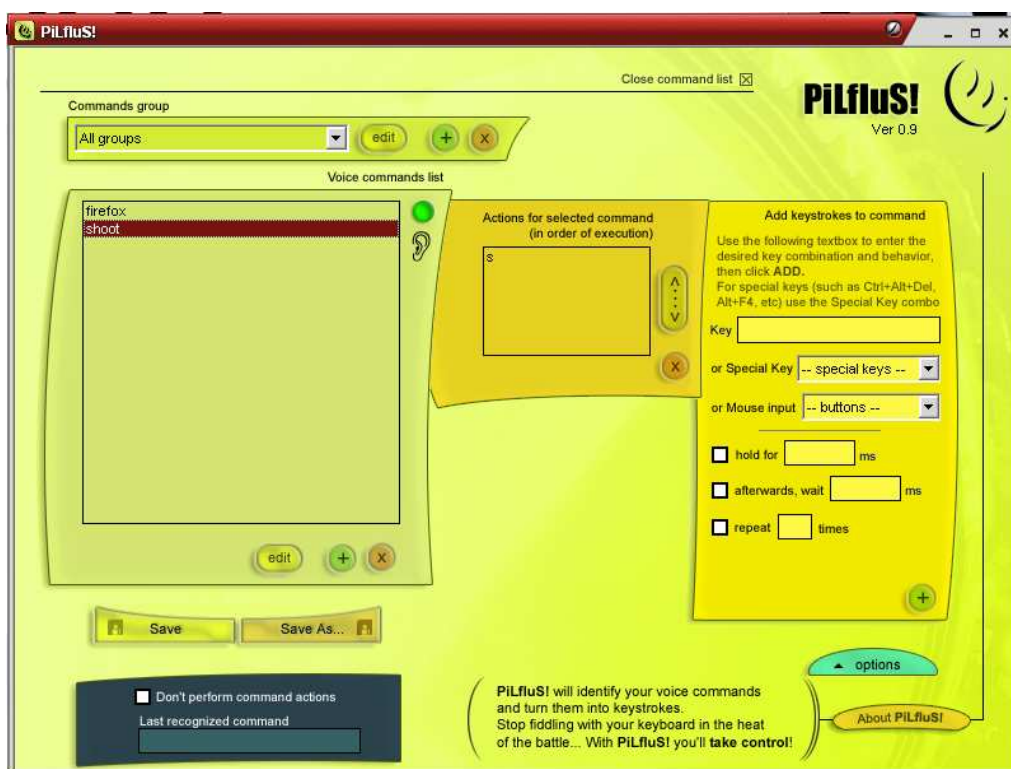
Obr. 3. Hra LifeLine. [2]

- druhú kategóriu tvoria aplikácie umožňujúce doplniť existujúce hry o možnosť ovládania hlasovými príkazmi. Väčšina aplikácií funguje na rovnakom princípe a to preklade hlasového príkazu do určitej kombinácie kláves alebo príkazu, ktorý daný program pozná. Rozdiel je väčšinou iba v správnosti rozpoznania príkazu, pričom táto sa hýbe do 95%, a v prítomnosti predkonfigurovaných príkazov. Predstaviteľmi tejto kategórie sú komerčné aplikácie: GameCommander, VoiceBuddy, VRCommander. Najzaujímavejším predstaviteľom tejto kategórie je nekomerčná aplikácia PiLfuS!

PiLfuS!

Ide o aplikáciu pod licenciou Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported license, ktorá umožňuje jej voľnú distribúciu pre nekomerčné účely. Jej distribúcia v komerčných aplikáciách si vyžaduje súhlas autora. Podľa [3] aplikácia prostredníctvom Microsoft SAPI premení zvukové príkazy na stlačenie kláves, pričom sa sústreďuje na použitie v počítačových hrách. Práve vďaka tejto orientácii disponuje (na rozdiel od aplikácií určených pre ovládanie operačných systémov) aj špeciálnymi možnosťami nastavenia (obr. 4):

- určenie poradia stlačených kláves – teda jeden hlasový povel vyvolá viacero stlačených kláves. Pri hrách toto výrazne napomáha efektívnosti ovládania.
- špecifikovať počet stlačených kláves na jeden hlasový povel
- špecifikovať dĺžku ich stlačenia
- nastaviť druh príkazu:
 - príkaz stále aktívny
 - príkaz na aktiváciu
 - príkaz na deaktiváciu
 - príkaz na posun



Obr. 4. PiLfuS! [3]

Čo sa týka použitia tejto aplikácie v rámci projektu, umožňovala by používateľom namiesto klávesnice používať hlasové povely. Výhodou je možnosť definovať zoznam príkazov a uložiť ich do súboru. Následne nie je problém do aplikácie takto vytvorený zoznam načítať. Aplikácia však nerieši dva ďalšie aspekty ovládania:

- Ovládanie intenzity mávania krídel komára bzučaním pomocou tejto aplikácie nie je možné, keďže aplikácia nijako neskúma ostatné aspekty používateľovho hlasu. Toto však nedokáže ani žiadna z komerčných aplikácií určených na hlasové ovládanie postavených na rozhraní Microsoft SAPI.
- Ovládanie smeru letu komára, keďže takéto ovládanie si vyžaduje citlivosť. Bolo by veľmi problematické také presné ovládanie pomocou hlasu, čo dokazujú aj

komerčné zvukovo ovládané hry, v ktorých pohyb je ovládaný myšou, aj keď všetko ostatné zvukovými povelmi.

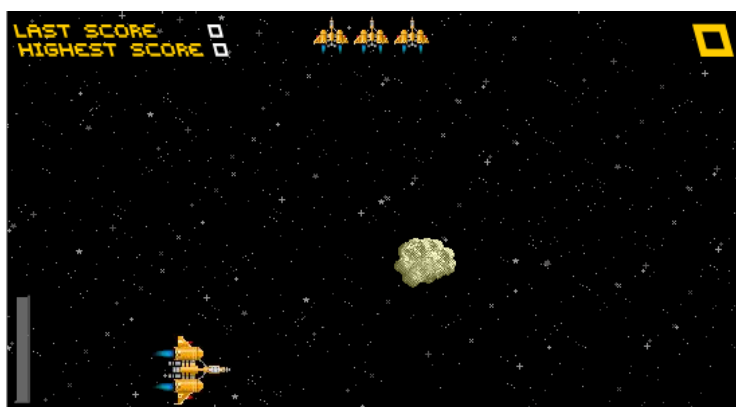
Aplikácia sa momentálne nachádza vo verzii 0.9 a doteraz nebola vydaná žiadna stabilná verzia. Výhodou aplikácie je, že beží na operačnom systéme Windows XP a aj na Windows Vista, pričom druhý spomenutý systém má inú verziu Microsoft SAPI. Jej funkčnosť na iných operačných systémoch je podmienená funkčnosťou danej knižnice, a teda sa dá v prípade iných systémov ako Windows vylúčiť.

2.2.2 Ovládanie zvukom

Pri vnímaní zvuku bez rozpoznania jeho vecného obsahu možno sledovať jeho dva aspekty: hlasitosť a výšku. Hlasitosť zodpovedá energii prenášanej zvukom. Výška priamo súvisí s frekvenciou a určuje, či je zvuk vysoký, ako napríklad zvuk violy, alebo nízky ako zvuk basy.

Oproti ovládaniu hlasovými povelmi sa s ovládaním zvukom bez jeho rozpoznania možno stretnúť len zriedka. Dôvodom sú malé kapacity takéhoto ovládania, keďže tento druh ovládania umožňuje väčšinou len zosilnenie či zoslabenie nejakej hodnoty. Napriek týmto nedostatkom našlo svoje uplatnenie v hrách ovládanie pomocou hlasitosti. Príkladom takto ovládaných hier sú:

- Pah! - Ide o klasickú 2D hru, v ktorej hráč ovláda vesmírnu loď a vyhýba sa asteroidom (obr. 5). Lodička letí horizontálne vyššie či nižšie podľa hlasitosti. Pri vyslovení slova „Pah!“ loď strieľa. Je realizovaná v Adobe Flash Player a na stránke vyhlasujú, že ide o prvú hlasovo ovládanú webovú hru. Grafika je veľmi jednoduchá ako i hra celá. Asteroidy letia vyššie takže človek nemusí nič robiť a nič sa nestane. Hlasové ovládanie má tiež svoje chyby. Ani pri najhlasnejšom kričaní sa loď nedostane až úplne hore. Rozoznanie slova „Pah!“ je veľmi zlé. Konflikt nastáva aj pri snahe ostať na danej úrovni a zároveň strieľať. [4] Podobných hier existuje viacero, napríklad hra Shout n Dodge [5].



Obr. 5. Hra Pah! [4]

- Ford Game - Princíp ovládania tejto 2D hry je podobný ako pri predošlej, akurát že sa nemení výška, ale rýchlosť. Čím hlučnejšie človek kričí, tým rýchlejšie auto ide. Jedinou zábavnou časťou hry je multiplayer. Dalo by sa povedať, že ide o súťaž silnejších hlasiviek. Grafika celej aplikácie je dosť jednoduchá, aj keď pekná, pričom aplikácia je opäť realizovaná v Adobe Flash Player (obr. 6). [6]



Obr. 6. Hra Ford Game. [6]

- Sing Pong – ide o obmenu klasického ping-pongu (obr. 7). Hráč začína hru, keď sa na plátne objaví tieň jeho ruky. Hlasitosť udáva výšku rakety, pričom jej pozícia je udávaná pohybmi hráča. Ten sa môže pohybovať dopredu, dozadu, hore a dole. Ide teda o hru, ktorá si vyžaduje aj fyzickú aktivitu. [7]



Obr. 7. Hra Sing Pong. [7]

Navrhovaná hra by bola priekopníkom v oblasti ovládania pomocou frekvencie zvuku medzi 3D hrami. Práve vďaka malému výskytu tohto druhu ovládania je však vyžadovaná kompletná implementácia z našej strany.

2.2.3 Ovládanie pohybom

Pri skúmaní možností ovládania pohybom sme sa zamerali len na ovládanie prostredníctvom pohybov hlavy. Snímanie pohybov hlavy je najčastejšie realizované prostredníctvom špeciálneho hardvéru, napríklad systém TrackIR, alebo pomocou bežnej webovej kamery. Systém TrackIR predstavuje špeciálne hardvérové snímacie zariadenia,

ktoré snímajú špeciálnu značku umiestnenú na hlave používateľa (prípadne na hlavovej súprave alebo šiltovke). Systém podporuje 6DOF a výrobca pravdaže poskytuje pre vývojárov knižnicu. Primárne je určený do komerčných 3D hier v prvej osobe na ovládanie pohľadu [8]. Najzaujímavejšie z pohľadu nami navrhovanej hry sú však aplikácie nevyžadujúce žiaden špeciálny hardvér. Ako príklad možno uviesť:

- Cachya – ide o aplikáciu podporujúcu 6DOF nevyžadujúcu extra hardvér a zároveň predikujúcu pohyb hlavy na základe histórie, čím výrobca sľubuje svižnejšie ovládanie. Aplikácia pracuje na základe analýzy dát z webkamery hľadajúc špecifickú značku na hlave používateľa. Pred samotným využívaním aplikácie je potrebná dosť náročná konfigurácia. [9]
- Eyetwig – slúži na ovládanie myši prostredníctvom pohybu hlavy. Ide o rozpoznávanie a stopovanie objektov v reálnom čase nevyžadujúce si žiaden extra hardvér. Na rozdiel od aplikácie Cachya nie je potrebná žiadna špeciálna značka na hlave, namiesto neho sa aplikácia sústreďí na identifikáciu polohy očí a brady. Kliknutie je realizované prostredníctvom žmurknutia, prídavné povely možno realizovať prostredníctvom ich hlasového rozpoznania. Primárne nie je určená pre hry, ale jej použitie v nich nie je vylúčené. [10]

Všetky spomínané aplikácie sú komerčné, takže ich využitie v rámci projektu neprichádza do úvahy. Naznačujú však možnosť implementácie ovládania pomocou pohybov hlavy bez využitia zvláštneho hardvéru, len pomocou bežnej webkamery.

2.3 Analýza softvérových prostriedkov

AnalYZovali sme niekoľko grafických i herných systémov a knižníc s cieľom nájsť ten najvhodnejší pre potreby projektu ([14], [16], [17], [18]). Analýzu dokumentujú tabuľky 2 a 3.

Do analýzy sme zahrnuli len systémy, ktoré vyhovovali požiadavkám nášho projektu:

- zadarmo pre nekomerčné použitie
- programovací jazyk C++ alebo Java
- platforma Windows

Najdôležitejšie kritériá, použité pri výbere najvhodnejšieho systému, zahŕňajú:

- plynulá grafika pre 3D herný svet
- rýchlo naučiteľné
- ľahko použiteľné, t.j. urýchľujúce vývoj

Tab. 2. 3D grafické a herné systémy spolupracujúce s C++.

Engine alebo knižnica	Klady	Zápory
OpenGL	rýchla grafika skúsenosti členov tímu	príliš nízkoúrovňové zdlhavá implementácia
Ogre3D	dobře zdokumentované ľahko pochopiteľné veľká vývojárska komunita	len 3D grafický engine
Crystal Space	kompletný herný engine zahŕňajúci zvuk, vstupné zariadenia, skripty a i.	ťažšie naučiteľné nedostatočne zdokumentované
The Nebula Device 2		ťažšie naučiteľné menšia vývojárska podpora
Blender Foundation		náročnejšie na používanie
OpenSceneGraph		ťažšie naučiteľné menšia vývojárska podpora
Obsidian		zastavený vývoj žiadna podpora
QuakeForge		ťažšie naučiteľné ťažšie použiteľné

Tab. 3. 3D grafické a herné systémy spolupracujúce s Javou.

Engine alebo knižnica	Klady	Zápory
JOGL		príliš pomalá grafika iba prepojenie Javy s OpenGL
JMonkey Engine		nie veľmi kvalitná grafika

2.4 Zhodnotenie analýzy

Z analýzy vyplynulo, že naša hra by síce nebola úplne jedinečná, keďže simulátory komárov už existujú, ale prináša dostatok nových faktorov, aby mala šancu uspieť na európskej súťaži Europrix. Pomáhať by jej v tom malo aj nie úplne štandardné ovládanie: či už ovládanie rýchlosti mávania krídel bzučaním používateľa (kde 3D hra ovládaná zvukom ešte neexistuje), alebo ovládanie aplikácie hlasovými povelmi.

Z analýzy ďalej vyplynulo, že použitie Javy ako vývojovej platformy nie je vhodné pre nízku rýchlosť grafických aplikácií, ktorú je možné kompenzovať len výrazným znížením kvality grafiky. Z porovnávaných grafických systémov sme vybrali Ogre3D, pretože najlepšie spĺňal zadané kritéria výberu.

3 Špecifikácia riešenia

3.1 Návrh koncepcie hry

Z pohľadu normovaných herných žánrov možno tento projekt zaradiť medzi simulátory letu s bohatými prvkami adventúry. Simulácia letu nie je nezvyčajný subjekt v hrách. Dalo by sa dokonca povedať, že je pomerne častá a patrí medzi hlavné herné žánre. Rovnako to platí aj o hernom žánre adventúry. Pomerne nezvyčajná je však ich vzájomná kombinácia. Atraktívnejšie a inovatívnejšie na našom projekte je aj to, že sa jedná o simulátor života hmyzu. Avšak nie akéhokoľvek, ale toho nám najznámejšieho a pritom najviac neznášaného - komára.

Chceli sme, aby hra obsahovala aj akési konštruktívne prvky, ktoré možno považovať za rovnako zábavné ako aj náučné. Hra sa preto neobmedzuje na obyčajný let komára svetom a cucanie krvi z náhodných okoloidúcich. Do hry sú zabudované pravidlá, ktoré určujú, ako má hráč krv dostávať, vyberať a čo má s ňou robiť. Pre účel pobavenia sú tieto prvky do hry zapracované humorným spôsobom.

3.1.1 Hlavná myšlienka

Ako už bolo spomenuté, hra sa točí okolo života komára, ale to nie je ani zďaleka všetko. V úvodnej animácii je hráčovi dané najavo, že komár, ktorého ovláda, je takpovediac "superkomár", ktorý vznikol náhodným kontaktom s mutagénnou substanciou v laboratóriu v nemocnici. Tu sme sa nechali inšpirovať komixovými klasikami, ako napríklad Spiderman.

Vo svete, v ktorom sa komár pohybuje, sa nachádza aj nemocnica, ktorej sa kráčia zásoby krvi v krvnej banke. Preto sa náš superkomár musí starať o ich včasné a dostatočné dopĺňanie novou krvou. Túto môže získavať z okoloidúcich a zo zvierat (pochopiteľne, keďže je superkomár, dokáže krv upraviť tak, aby mohol zmiešať ľudskú krv so zvieracou bez ujmy na zdraví pacientov). Pri zbere sa komár musí riadiť krvnými skupinami svojich obetí a naplniť príslušnú krvnú konzervu.

Za každé doplnenie krvi do správnej krvnej konzervy sú hráčovi pridané body, za ktoré si môže v laboratóriu nakupovať vylepšujúce prvky, ako napríklad prídavné nádrže na krv, silnejší vrták, či napríklad diaľkový detektor krvnej skupiny. Za každé nesprávne priradenie krvnej skupiny pochopiteľne body stráca, rovnako ako aj čiastočne znehodnocuje obsah krvnej konzervy.

Krvné konzervy sa počas hry postupne mívajú, preto musí hráč neustále krv dopĺňať. Aby sa hra nestala fádnu, sú hráčovi počas hry pridávané nové vylepšenia, ktoré si môže na komára pridať, ale zároveň sa stále rýchlejšie mívajú krv v konzervách. Toto je podobný princíp ako v prípade hry Tetris, kde sa sila hráča hodnotí primárne tým, koľko bodov dokáže získať predtým, než ho hra "premôže".

Okrem časových obmedzení na komára pôsobia ďalšie nástrahy, ktoré sa mu snažia v jeho konaní zabrániť. Jedná sa o klasické nástrahy ako napríklad hrozba zabitia rukou svojej obeť, ale aj o nezvyčajnejšie nástrahy, ako napríklad súboj s vážkou alebo vyhábanie sa náletom lastovičiek.

3.1.2 Motivácia

Každá hra má v prvom rade pobaviť. Náš projekt má však aj náučný rozmer - má poučiť svojich (primárne mladých) hráčov o krvných skupinách a zároveň upozorniť na to, o ako

nedostatkový materiál v skutočnosti ide. Tiež nám tento projekt umožňuje sklbiť niekoľko bežných herných konceptov, ako napríklad let, adventúrne prvky, prvky súbojov a náučné prvky do zaujímavého a nového projektu.

3.1.3 Svet

Dej hry je situovaný v nepomenovanom meste v blízkej budúcnosti. Väčšinu dostupného prostredia tvoria domy, parky a ulice. Okrem toho sa komár môže pohybovať aj v prostredí neprístupnom pre človeka, ako sú napríklad kanály a vetracie šachty. Jedinou pre hru významnou budovou je nemocnica, ktorá je centrálnym miestom príbehu. Rozloženie jednotlivých stavieb v meste je konštantné, a teda pre každú hru rovnaké. Svet nie je rozdelený na nezávislé časti, hráč má možnosť preskúmať krajinu podľa svojho uváženia. Niektoré miesta sú však ťažko prístupné, pokiaľ sa hráčovi nepodarí získať potrebné vybavenie.

Plocha hry je obmedzená, nie je možné ju opustiť. Toto ohraničenie môže byť riešené priamo, napríklad vysokými budovami po celom obvode plochy, pričom tieto budovy nie je možné nadletieť. Alternatívnym riešením je mesto umiestniť na ostrov, pričom limitovaný dolet komára znemožní prelet na iný ostrov.

Mesto je obývané ľuďmi a dá sa tu naraziť aj na rôzne zvieratá. Väčšinou sú to domáce zvieratá, ale v kanáloch je možné stretnúť aj krysy a na strechách budov vtáky. Hlavnou postavou je zmutovaný komár, ktorý nedokáže komunikovať s ľuďmi. Rovnako však nedokáže komunikovať so zvyšným nezmutovaným hmyzom. Dokáže však čítať, takže sa môže dozvedieť informácie z plagátov a tabuliek.

3.1.4 Nepriatelia

Hra je komplikovaná hlavne kvôli faktu, že obyvatelia mesta netušia nič o dobrých úmysloch komára. A preto najčastejšie nebezpečenstvo pre komára predstavujú práve ľudia, ktorí sa ho snažia zabiť rôznym spôsobom, či už úmyselne alebo nie. Používajú pritom prostriedky ako ruky, noviny alebo sprej proti hmyzu. Bežný človek však má na práci dôležitejšie veci ako zabíjanie komára. Ľudia sa ho preto pokúsia zabiť, iba ak ich komár obťažuje. Z hľadiska umelej inteligencie to znamená, že komára nebude prenasledovať jeden človek niekoľko minút podľa prepracovaného plánu. Človek, ktorý stratí komára z dohľadu, sa vráti k svojej bežnej činnosti.

Druhú skupinu nepriateľov tvoria zvieratá. Opäť niektoré zvieratá môže komár pripraviť o krv a tie sa mu v tom budú pokúšať zabrániť. Existujú aj pavúky a vtáky, ktoré sa snažia komára uloviť. Komár na ich zlikvidovanie musí použiť svoje schopnosti a zbrane, ktoré získava za splnené úlohy.

Okrem toho predstavuje nebezpečenstvo aj prostredie samotné. Komár napríklad môže zomrieť pri zavalení alebo sa môže utopiť pri nezvládnutom prelete ponad vodu. Ďalej hrozia nepriaznivé poveternostné podmienky, rozložené ventilátory alebo otvorený oheň.

Komár má len jeden život. Aktuálny stav hry si môže hráč uložiť na vyhradenom mieste (v spomínanej nemocnici). V prípade smrti hra pokračuje od posledného uloženia.

3.1.5 Vybavenie

Vzhľadom na svoju hmotnosť komár nemôže zdvíhať predmety z prostredia a brať ich so sebou. Vďaka svojej mutácii ale dokáže používať špeciálne vybavenie, ktoré získa v oddelení nemocnice. To mu prináša rôzne výhody, ako napríklad nočné videnie, vyššiu rýchlosť alebo väčšiu kapacitu nádrží na krv. Na používanie týchto vylepšení ale potrebuje

komár nadobudnúť skúsenosti s ovládaním svojich mutácií. Preto získava vybavenie priebežne, v závislosti od úspešnosti riešených úloh a bodov, ktoré za ne získa.

Komár spotrebúva na let malý objem krvi. Väčšie množstvo prídavných zariadení zvyšuje spotrebu krvi. Okrem toho využívanie niektorých technológií tiež vyžaduje krv.

Vybavenie, ktoré komár môže získať, rozdeľujeme na dve skupiny:

- vylepšenia zvyšujúce schopnosti
 - prídavné motory – rýchlejší pohyb, väčšie zrýchlenie
 - ďalekohľad – diaľkové sledovanie
 - špeciálne okuliare – širokouhlé videnie
 - prídavné nádoby – väčšia nosnosť krvi
 - zameriavač – presnejšia streľba
 - pancier, štít – väčšia výdrž v súbojoch
 - kompresor – rýchlejšie satie krvi
- vylepšenia pridávajúce nové schopnosti
 - GPS ukazovateľ pozície, navigácia v kanáloch podľa máp
 - radar – detekcia nepriateľov
 - nočné videnie
 - detektor krvnej skupiny (bez tohto prístroju je nutné nemiešať krv dvoch ľudí)
 - strelné zbrane rôznych druhov, riadené strely, časované bomby
 - plynová maska – umožňuje prelet cez oblasti zamorené chemikáliami proti hmyzu

3.1.6 Priebeh a cieľ hry

Ako bolo spomenuté, svet hry nie je rozdelený na nezávislé časti. Hráč si sám môže zvoliť svoje ďalšie napredovanie, čo však neznamená, že sa prechádza bezcieľne. Počas hry získava rôzne úlohy, ktoré musí plniť. Eventuálne splnenie všetkých úloh znamená víťazné ukončenie hry. Nesplnenie úlohy môže priniesť rôzne následky, a to v závislosti od príslušnej úlohy. Prvé úlohy sú zadané už na začiatku hry, takže hráč má cieľ po celý čas hry.

Základnou úlohou komára je zbierať krv a nosiť ju do nemocnice. Hráč je motivovaný zbierať krv rýchlejšie a efektívnejšie, pretože krv sa priebežne míňa. Môžu sa vyskytnúť rôzne prekážky, ako napríklad nutnosť získať špecifickú krvnú skupinu alebo protilátku. Nesplnenie týchto úloh znamená smrť pacientov, a teda prehru. Tieto úlohy vznikajú bez pričinenia hráča, pretože vyplývajú z činnosti nemocnice, a sú zvyčajne časovo limitované.

Okrem toho existujú aj vedľajšie úlohy, ktoré sa netýkajú prežitia pacientov. Komár napríklad bude mať za úlohu určiť na základe odobratej krvi otcovstvo strateného dieťaťa. Ďalšie úlohy môže získať navštívením rôznych oblastí mesta, kde bude musieť riešiť aktuálne problémy.

Plnenie úloh je pre hráča zdrojom bodov. Rýchlejšie a lepšie splnené úlohy prinášajú samozrejme viac bodov. Množstvo bodov reprezentuje lepšiu schopnosť ovládať mutáciu

komára a umožňuje hráčovi pridávať komárovi ďalšie schopnosti a používať technické vylepšenia.

Základné úlohy v hre sú nasledovné:

- dopĺňanie zásob krvi v nemocnici podľa krvných skupín
- získanie protilátky z krvi niektorých ľudí
- pomoc pri autonehodách a úrazoch dodaním krvi
- likvidácia nakazených zvierat, ktoré rozširujú choroby

V hre existuje aj viacero vedľajších úloh, napríklad:

- zistenie otcovstva na základe krvi, usvedčenie zločincov na rovnakom princípe
- nazbieranie maximálneho množstva krvi v obmedzenom čase (v rámci akciových zberov krvi)
- zbieranie súčiastok potrebných na aktiváciu nového vylepšenia
- likvidácia prirodzených nepriateľov za účelom otvorenia nových prechodov medzi dvoma miestami

3.2 Používateľská interakcia

Na základe analýzy jednotlivých možností ovládania s prihliadnutím na potreby aplikácie sme sa rozhodli skombinovať viaceré metódy ovládania. Na ovládanie letu komára sa nám ako najvhodnejšie javilo využiť ovládanie frekvenciou hlasu hráča. Konkrétne bude rýchlosť mávania krídel komára závisieť od frekvencie bzučania hráča. Tento spôsob nielen odbremení jednu ruku hráča, ale zároveň umožní hráčovi vžiť sa do úlohy komára.

Ovládanie smeru letu komára si vyžaduje presnosť, preto prichádzali do úvahy dve možnosti. Overeným riešením je využitie myši, ktoré síce vyžaduje využívanie jednej z rúk používateľa, ale ako protihodnotu poskytuje veľmi príjemné, známe a presné ovládanie a umožňuje tak používateľovi nerušene sa venovať hre. Dostatočnú presnosť by mohlo poskytovať aj ovládanie prostredníctvom pohybov hlavy snímaných pomocou bežnej webovej kamery. Táto technológia je však dosť náročná a výsledná presnosť nie je zaručená. Zároveň by potreba otáčania či nakláňania hlavy mohla pôsobiť na hráčov rušivo. Preto sme sa rozhodli pre overený spôsob a smer letu komára bude ovládaný pomocou myši.

Pri ovládaní akcií komára a pohybu v menu sme sa rozhodli umožniť používateľovi výber. Aplikáciu bude možné ovládať pomocou klávesnice a myši a zároveň hlasovými povelmi. V analýze bolo predstavených niekoľko aplikácií umožňujúcich transformovať hlasové povely na stlačenie kláves. V navrhovanej hre sme sa rozhodli využiť aplikáciu PiLfuS!. Jej licencia nám umožňuje jej voľné šírenie s našou nekomerčnou hrou. Na rozdiel od priameho využitia knižnice Microsoft SAPI, nad ktorou pracuje, už obsahuje veľmi vhodne preddefinované akcie spojené s hlasovým povelom.

Zo zvolených spôsobov ovládania vyplýva niekoľko obmedzení a požiadaviek aplikácie. Zachytávanie zvuku z mikrofónu je možné realizovať pomocou knižnice DirectSound, ktorá je súčasťou knižnice DirectX štandardne prítomnej na platforme Windows. Pre funkčnosť ovládania pomocou hlasových povelov je potrebná prítomnosť knižnice Microsoft SAPI, ktorá štandardnou súčasťou operačných systémov Windows Vista

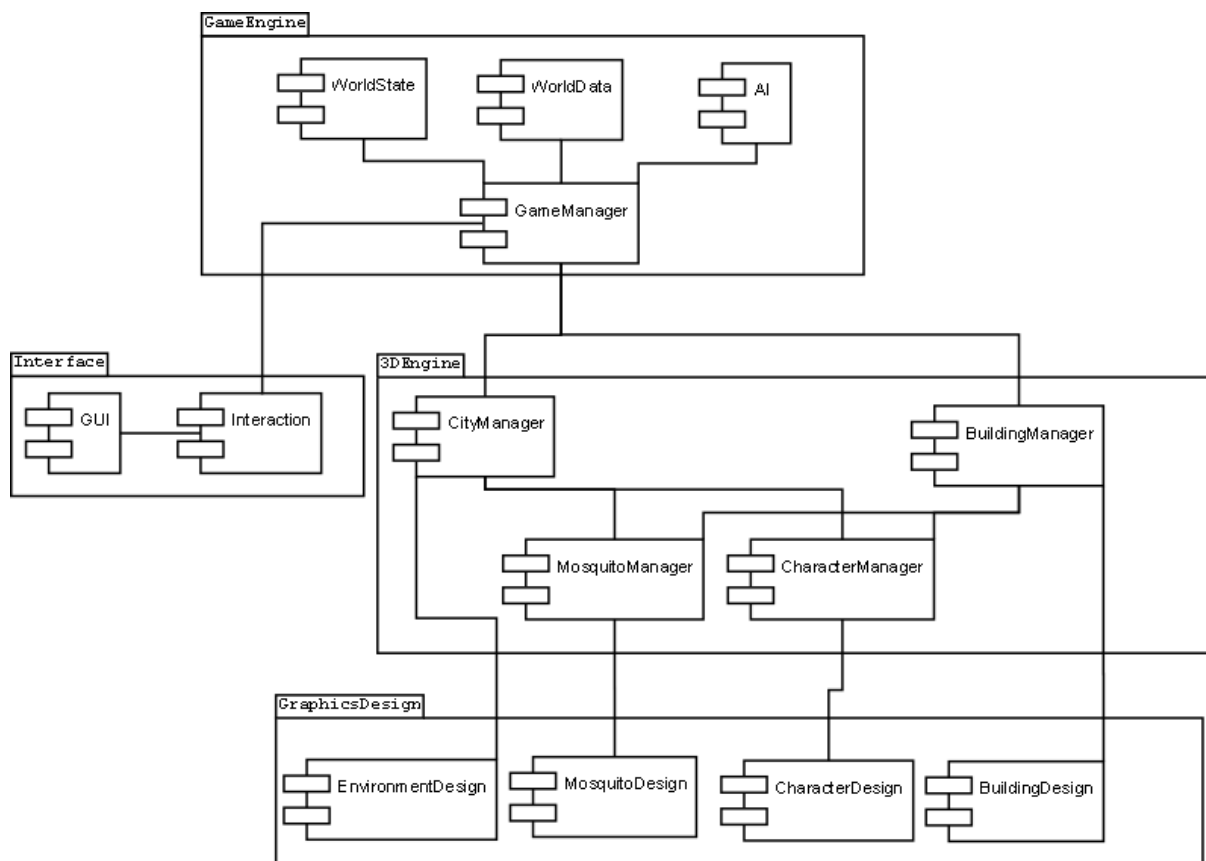
a pre predošlé verzie existuje samostatný inštalateľný balík, ktorý bude musieť byť súčasťou výslednej inštalácie našej hry.

Voľba ovládania zvukom, či už frekvenciou zvuku alebo hlasovými povelmi, však nesie so sebou aj niekoľko ďalších obmedzení. Problémom by mohli byť zvuky samotnej hry, ktoré by mohli byť zle interpretované. Hra bez akýchkoľvek zvukov by mohla pôsobiť fádne, preto nie je možné úplne tieto zvuky odstrániť. Možným riešením je použitie headsetu.

4 Hrubý návrh riešenia

4.1 Architektúra systému

Aplikácia pozostáva zo štyroch hlavných modulov (obr. 8), pričom zobrazené sú hlavné komponenty, z ktorých jednotlivé moduly pozostávajú. Architektúra bola vytváraná so zreteľom na špecifikáciu hry a na logickú a funkčnú súvislosť medzi jednotlivými komponentmi.



Obr. 8. Architektúra aplikácie.

Hlavným modulom je modul *GameEngine* obsahujúci celú aplikačnú a hernú logiku. Jeho úlohou je uchovávať informácie o svete, v ktorom sa hra odohráva, o dosiahnutých výsledkoch hráča, zmenách vo svete a zároveň reaguje na požiadavky hráča, ktoré dostáva prostredníctvom modulu *Interface*. Jeho prostredníctvom je hráč zároveň informovaný o udalostiach, ktoré sú priamym alebo nepriamym dôsledkom jeho konania alebo vyplynuli z pravidiel a zákonitostí hry.

Modul *Interface* je vlastne rozhraním medzi používateľom a aplikáciou. Nejde len o grafické, ale aj logické rozhranie schopné spracovať používateľské príkazy. Aplikácii sú teda poskytované len skutočne potrebné informácie a dáta, čím je samotná logika aplikácie odbremenená od ďalšieho spracovania získaných dát.

Modul *3DEngine* sa stará o vykresľovanie 3D sveta, v ktorom sa samotná hra odohráva. Od modulu *GameEngine* dostane informácie o objektoch, ktoré je potrebné vykresliť, za spôsob ich vykreslenia je zodpovedný tento modul. Pri vytváraní objektov úzko spolupracuje s modulom *GraphicsDesign*, od ktorého dostáva informácie o výzore jednotlivých objektov. Následné umiestnenie objektov do sveta sa riadi buď informáciami získanými z modulu *GameEngine*, alebo pravidlami umiestňovania konkrétnych objektov.

Modul *GraphicsDesign* obsahuje dáta potrebné na vytvorenie jednotlivých objektov. Ide najmä o dáta týkajúce sa rozmeru, tvaru a výzoru jednotlivých objektov.

4.1.1 Modul Interface

Úlohou tohto modulu je sprostredkovať komunikáciu medzi používateľom a samotnou aplikáciou. Modul pozostáva z dvoch komponentov a to *GUI*, teda grafického používateľského rozhrania, a *Interaction*.

GUI komponent zabezpečuje všetky časti grafického používateľského rozhrania. Z hľadiska charakteru hry sa snažíme, aby bolo prehľadné a čo možno najjednoduchšie. Preto okrem zameriavača, ktorý je statickou časťou rozhrania, ho môžeme rozdeliť na tri ďalšie časti:

- **Krvometer** – zobrazený ako skúmavka. Ten bude zobrazovať aktuálnu hladinu krvi, ktorú komár nesie. Hladina sa bude meniť v závislosti od hrania, ak bude komár len bezcieľne lietať, bude ubúdať ako palivo, ak ju bude vysávať z ľudí, bude pribúdať. Pravdepodobne by mali byť odlišené aj jednotlivé krvné skupiny.
- **Mapa** – v ľavom dolnom rohu pre lepšiu orientáciu v priestore. Bude ukazovať aktuálnu pozíciu v hre, ako aj kľúčové miesta a prípadne pozíciu nepriateľov.
- **Čas** – hodinky v tvare HH:MM:ss, ktoré ukazujú aktuálny čas hrania, prípadne sa môžu zmeniť na odpočet pri časovo závislých úlohách.

Komponent *Interaction* je pre používateľa neviditeľnou súčasťou interakcie. Hlavnou úlohou tohto komponentu je prekladať informácie od používateľa do formy zrozumiteľnej pre zvyšok aplikácie a prezentovať používateľovi informácie poskytované aplikáciou vyžadujúce špeciálne spracovanie. Komponent sa skladá z nasledujúcich častí:

- Rozpoznávanie hlasových povelov – hlasové povelý používateľa sú rozpoznané a na základe tabuľky podporovaných povelov je aplikácia informovaná o požiadavke na danú akciu.
- Rozpoznávanie frekvencie hlasu – monitorovaná je frekvencia bzučania používateľa a aplikácii sú podávané informácie o jej aktuálnej hladine, na základe ktorých je možné určiť rýchlosť komára.
- Spracovanie stlačení kláves – aplikácia je informovaná o požiadavke na vykonanie daného povelu.
- Spracovanie pohybu myši – aplikácia je informovaná o zmene pozície myši, na základe ktorej sa zmení smer pohybu komára.

- Prehrávanie hudby – počas niektorých udalostí je vyslaná požiadavka na prehranie konkrétneho zvuku alebo hudby, pričom samotné prehranie zabezpečuje práve tento komponent.
- Prehrávanie videa – kľúčové momenty v hre sú prezentované aj prostredníctvom krátkeho videa, ktorého prehranie je úlohou tohto komponentu.

4.1.2 Modul 3DEngine

Modul *3DEngine* zabezpečuje vykreslenie 3D scény na obrazovku. Použitie Ogre3D zjednodušuje túto úlohu na zadefinovanie 3D objektov a ich vlastností. Modul sa podľa kontrolovaných objektov rozdeľuje na komponenty, ako ukazuje tab. 4.

Tab. 4. Komponenty modulu 3D engine.

Komponent	Spravované 3D objekty
Manažér mesta	Povrch mesta (trávnik, chodník, cesta a i.) Obloha Budovy (vonkajší vzhľad) Predmety a objekty v meste (stromy, lampy a i.) Neprevzaté vylepšenia pre komára
Manažér budov	Vnútorne priestory budov Predmety a objekty vo vnútri budov Neprevzaté vylepšenia pre komára
Manažér postáv	Ľudia Zvieratá (lastovičky, pavúky a i.)
Manažér komára	Komár Kamera

Manažér mesta zabezpečuje vykresľovanie 3D scény ulíc mesta, v ktorom sa dej odohráva, t.j. povrchu mesta, oblohy a všetkých 3D objektov okrem postáv. Tiež zobrazuje efekty súvisiace s počasím ako dážď alebo hmla.

Manažér budov zobrazuje vnútorné priestory budovy, v ktorej sa hráč aktuálne nachádza. V budove sa tiež môžu nachádzať rôzne predmety.

Vždy je aktívny iba jeden z manažérov mesta a budov, aby sa znížili grafické nároky aplikácie.

Manažér postáv riadi animáciu herných postáv a s tým spojené grafické efekty. Zabezpečí zobrazenie len tých postáv, ktoré zodpovedajú polohe komára, napr. postavy v budove, v ktorej sa komár nachádza, alebo postavy mimo budov.

Manažér komára kontroluje animáciu hlavnej postavy, pohľad kamery a vizuálne módy kamery. Keďže tento manažér kontroluje hráčom ovládanú postavu, musí byť aktívny vždy.

4.1.3 Modul GameEngine

Tento modul reprezentuje samotnú hernú logiku, ktorá funguje „za scénou“. Jeho obsahom sú podmoduly, ktoré zabezpečujú tieto dôležité funkcie:

- GameManager – obsluhuje stav sveta. Tento modul je hlavný riadiaci prvok programu. Presadzuje vstupy od používateľa do akcií vo svete, kontroluje jeho zmenu, generuje udalosti a rieši ich dopad, rieši fyzikálny model svetu, atď. Interne obsluhuje 3DEngine a všetky ostatné komponenty systému, čím prezentuje stav hry hráčovi.
- WorldState a WorldData – sú v podstate podobiekty v Game Engine, ktorých úloha je fungovať ako repozitár objektov herného svetu a ich stavov. GameManager realizuje zmeny vo svete práve zmenou týchto objektov a následne ich stav prezentuje hráčovi pomocou ostatných častí systému, ako napríklad 3D grafikou alebo zvukovými sprievodmi.
- AI – tento subsystém bol vydelený z GameManager modulu, pretože obsluhuje vcelku komplexnú črtu hry – umelú inteligenciu postáv obsiahnutých v hre. Z pohľadu GameManager-a je AI modul teda akýsi blackbox, čím je ponechaná implementácia správania sa postáv v samostatnom module a zjednodušuje sa tým návrh GameManager modulu.

GameEngine modul teda reprezentuje akúsi centrálnu výmennú stanicu v hre. Je prepojený prakticky so všetkými ostatnými subsystémami a zodpovedá za ich riadenie. Pritom výhodne využíva ich programové rozhrania, čím sa nemusí samotný zaoberať vnútornými komplexitami ich jednotlivých úloh.

Modul *AI* má na starosti umelú inteligenciu nepriateľov v hre. Nepriatelia sa delia na dve skupiny – na zvieratá a na ľudí. Prvá skupina vykazuje jednoduchú inteligenciu, väčšinou je to útočenie alebo stráženie svojho brlohu. Ľudia však musia vykonávať činnosti aj v neprítomnosti komára.

Budovy v meste majú rôzne funkcie, hoci pre komára sú nepodstatné. Obyvatelia mesta sa počas dňa presúvajú medzi týmito lokalitami. Mapa mesta je konštantná, a v každej hre rovnaká. Program bude obsahovať zoznam ciest, po ktorých sa štandardne postavy pohybujú. Postavy budú voliť podľa určitých vzorcov náhodnú z týchto ciest. Vďaka tomu pohyb nebude náhodný, ale postavy sa ani nebudú periodicky prechádzať po jednej ceste. Vzorec, na základe ktorého si postava zvolí cestu, bude pre každú postavu iný. Každá osoba má niekoľko preferovaných lokalít, ktoré zvyčajne navštevuje (dom, zamestnanie). Okrem toho si osoba bude pamätať niekoľko posledných navštívených miest, aby sa nepohybovala dokola.

Človek v prítomnosti komára dočasne pozmení svoj cieľ, aby komára odplašil, prípadne zabil. Stále si ale pamätá svoj pôvodný cieľ, ku ktorému sa vráti, ak sa komár vzdialí. Rovnako sa človek nenechá odtiahnuť komárom preč od svojho cieľu. Počas boja s komárom sa človek nepohybuje po predvolených cestách, ale je ovládaný niektorým z algoritmov, v závislosti od spôsobu boja. Každý spôsob boja sa aspoň sčasti opiera o hľadanie cesty (ku komárovi, ku dverám), pretože túto cestu nemá v žiadnom dopredu vypočítanom zozname.

Poslednou zložkou inteligencie postáv je správanie. Dve osoby s rovnakou cestou sa po nej pohybujú rôznym spôsobom. Napríklad starec je pomalý, zatiaľ čo školák prebieha cez cestu aj na červenú.

Vykonávanie týchto úloh má na starosti modul umelej inteligencie. Skladá sa zo 4 častí:

- Mapa ciest
- Manažér postáv
- Manažér správania

- Manažér zvierat

Mapa ciest obsahuje významné body v meste a cesty medzi týmito bodmi. Jednotlivé miesta sú zaradené do skupín podľa svojich funkcií (obytné domy, obchody, zamestnania).

Manažér postáv si pre každú postavu pamätá dom, v ktorom býva. Okolie domu tvorí prirodzené prostredie, v ktorom sa postava pohybuje. Tento manažér má za úlohu voliť pre postavu na základe mapy a zoznamu navštívených miest ďalší cieľ pohybu.

Manažér správania ovláda postavu tak, aby dosiahla krátkodobý cieľ svojím charakteristickým spôsobom chovania. Jeho úlohou je napríklad zabezpečiť obchádzanie prekážok. Odplašenie komára je z tohto pohľadu krátkodobý cieľ, a tiež ho spravuje tento komponent.

Manažér zvierat sa stará o pohyb zvierat a o ich ovládanie počas boja. Zvieratá majú pre jednoduchosť len krátkodobé ciele. Tento komponent sa teda funkciou podobá na predošlý.

4.1.4 Modul GraphicsDesign

Modul *GraphicsDesign* obsahuje modely, ktoré bude *3DEngine* vykresľovať, spolu so zodpovedajúcimi materiálmi, textúrami, *bump* mapami a podobne. Pri modeloch by sa malo jednať o objekty s nízkym počtom polygónov, kvôli nižším hardvérovým nárokom a rýchlosti hry. Textúry sú obrazové súbory, ktoré pokrývajú 3D objekty. Tieto sú v klasických formátoch .jpg alebo .png kvôli transparentnosti. Jednotlivé objekty budú obsluhované príslušnými manažermi v module *3DEngine*. Modul sa podľa objektov a častí rozdeľuje na komponenty, ako ukazuje tab. 5.

Tab. 5. Komponenty modulu GraphicsDesign.

Komponent	Objekty a textúry
<i>EnvironmentDesign</i>	Povrch mesta (trávnik, chodník, cesta a i.) Obloha Predmety a objekty v meste (stromy, lampy, fontána a i.)
<i>MosquitoDesign</i>	Model komára Predmety a vylepšenia pre komára
<i>CharacterDesign</i>	Ľudia Zvieratá (lastovičky, pavúky a i.)
<i>BuildingDesign</i>	Budovy (vonkajší vzhľad) Vnútorné priestory budov vrátane objektov

EnvironmentDesign obsahuje textúry a povrchy pre trávnik, chodníky, cesty a podobne. Obloha sa mení v závislosti od dennej doby a od počasia, preto je viacero rôznych textúr pre oblohu. Ostatné podmienky ako svetlo a hmlu bude zabezpečovať samotný *3DEngine*. Rovnako táto časť obsahuje aj objekty, ktoré sa v meste budú opakovať a vyskytovať vo väčšom počte (lavičky, lampy, hydranty, stromy a kríky).

MosquitoDesign obsahuje modely komára ako hlavnej postavy. Tento model je detailnejší ako ostatné objekty, keďže sa jedná o hlavnú a jedinou hrateľnú postavu v hre. Tu sa nachádzajú aj všetky predmety a vylepšenia, ktoré môžeme získať a používať v hre, teda napríklad prídavné nádrže, okuliare na nočné videnie a podobne. Takisto aj iné textúry, ktorými bude možné meniť povrch komára v priebehu hry.

CharacterDesign tvoria všetky ostatné nehrateľné postavy vyskytujúce sa v hre. Teda ľudia a zvieratá, s ktorými bude možná interakcia. V hre budú existovať zvláštne postavy, ktoré budú mať iné schopnosti oproti „bežným“ ľuďom, ako napríklad policajt, mafián, hipík, a potom „normálna“ populácia, ktorú by malo tvoriť niekoľko rôznych modelov postáv s náhodne vyberanými textúrami, aby sa dosiahla rôznorodosť. Ďalej sa tu nachádzajú nepriatelia, ako napríklad pavúk a lastovička.

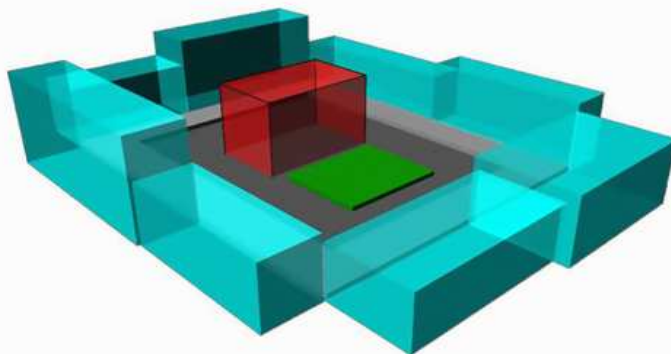
BuildingDesign časť obsahuje modely všetkých budov v hre vrátane vnútorných priestorov. Hlavnou budovou by mala byť nemocnica, prípadne ešte nejaké menšie budovy.

4.2 Návrh sveta

Dej sa odohráva v nie veľmi vzdialenej budúcnosti v bližšie nešpecifikovanom čase a mieste. Prostredie je časť mesta, logicky izolovaná. Bude ju tvoriť blok domov, ktorý obkolesuje hlavné budovy a objekty, ktoré majú zmysel z hľadiska hrateľnosti (obr. 9). Kľúčovou budovou je nemocnica, ktorá priamo súvisí s príbehom hry a poskytuje niektoré z hlavných funkcií hry (vylepšovanie vlastností, úprava hlavnej postavy, ukladanie aktuálnej pozície), ako aj funkcie súvisiace so zmyslom hry (odkladanie krvi a pod.). Predbežne sa navrhuje hlavne oddelenie transfúzie a rádiológie, súvisiace s premenou hlavnej postavy. Ďalšie časti sveta by mal tvoriť park pred nemocnicou s lavičkami, fontánou, prípadne detským ihriskom.

Predpokladajú sa ešte aj nejaké ďalšie časti, ale ich zahrnutie bude závisieť od prvotného prototypu a neskoršieho podrobnejšieho návrhu. Svet by mal obsahovať klasické objekty ako lampy, smetné koše, autá, rastlinstvo a podobne.

Grafika sa sústreďí skôr na komixovú farebnosť a štylizovanosť než na fotorealisticnosť. Preto sa u objektov a budov predpokladá trochu prehnaná geometria, nepravouhlosť a podobne. Svet by mal mať jasnejšie farby a možno menej detailov, skôr na úrovni symboliky (nemocnica nebude vysoká budova s veľa oddeleniami, ale bude zachytávať základné charakteristické črty).



Obr. 9. Schéma herného sveta.

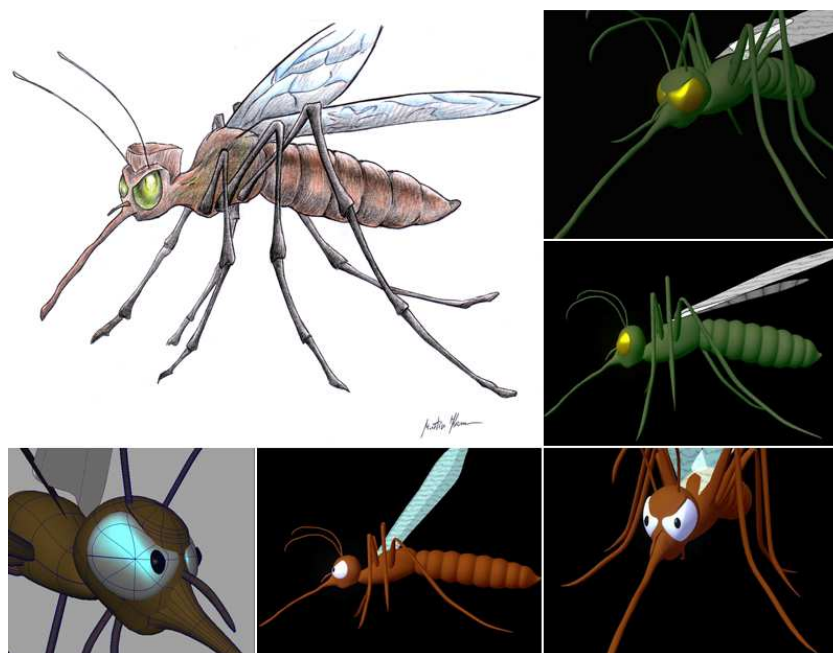
4.2.1 Návrh príbehu

Hlavnou postavou príbehu je komár, ktorý zháňa krv. Popri tom sa dostane do nemocnice, kde by jej mal byť dostatok. Miesto toho sa však ocitne omylom na inom oddelení, kde sa vyvíjajú nové experimentálne prístroje. Pri skúmaní vletí do podivného nukleárneho zariadenia, ktoré spôsobí, že komár zmutuje, narastie a získa schopnosť

vylepšovania pomocou rôznych technologických prístrojov (prídavné nádrže, nočné videnie a rôzne pohľady, zameriavač...). Rozhodne sa pokračovať v pôvodnom zámere, keď vtom zistí, že krvi je nedostatok pre pacientov a rýchlo sa mína. Preto sa rozhodne, že pomôže zachrániť ľudské životy a bude plniť rozličné úlohy, pomocou ktorých sa to bude snažiť dosiahnuť.

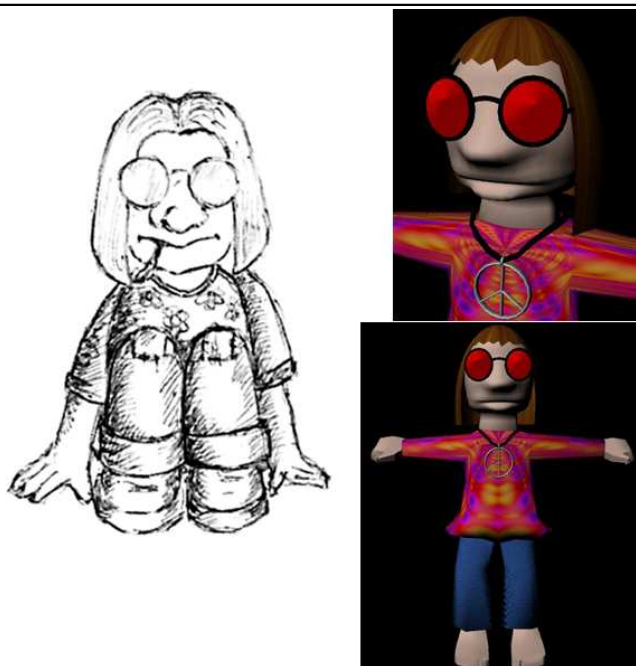
4.2.2 Návrh postáv

Hlavná postava je komár, ktorý je jedinou ovládateľnou postavou v celej hre (obr. 10). Táto postava by mala byť detailnejšia a s vyšším počtom polygónov ako ostatné postavy. Pôvodná myšlienka bola vytvoriť komára ako sci-fi postavu z budúcnosti. Neskôr sa z hľadiska konceptu ukázalo vhodnejšie, aby vyzeral skôr ako veselé postavičky z animovaných filmov, čo viac zodpovedá zábavnému zámeru hry. Zároveň to umožňuje napríklad znázorniť väčšiu škálu výrazov tváre.



Obr. 10. Character design hlavnej postavy komára.

V našom svete by sa mali vyskytovať rôzne druhy NPC postáv (Non-player character). Niektoré by mali byť špeciálne a mať špeciálne vlastnosti (policajt, mafián, hipík, mama s dieťaťom...) a zvyšok by mali byť náhodne generované postavy, kde by sa len obmieňali niektoré komponenty a tvorili by hlavnú časť „populácie“. Tieto postavy by mali byť logicky jednoduchšie s nižším počtom polygónov (obr. 11), keďže ich bude treba v reálnom čase vykresliť niekoľko, takže príliš zložité modely by značne spomalili hru a spôsobili značné zvýšenie nárokov na hardvér. Je možné, že využijeme aj niektoré voľne sťahuteľné postavy z internetu, ak sa ukážu ako vhodnejšie na použitie.



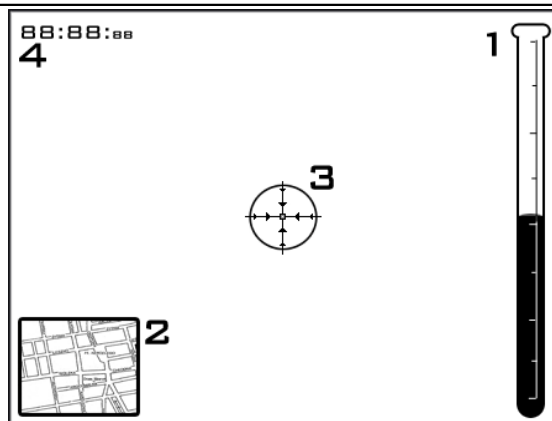
Obr. 11. Character design NPC postavy.

4.2.3 Návrh grafického používateľského rozhrania

Grafické používateľské rozhranie (GUI) priamo v hre je čo najjednoduchšie a čo najviac intuitívne. Nikto predsa nechce, aby mu počas hry blikali na obrazovke kontrolky, ukazovatele, číselníky a ikonky. Preto obsahuje len niekoľko komponentov:

1. **Krvometer** – zobrazuje množstvo krvi, ktorú komár práve nesie. Mala by sa používať aj ako palivo na lietanie, preto jej bude postupom času aj ubúdať.
2. **Mapa** – ukazuje našu polohu v meste a kľúčové lokácie, kam treba ísť pri plnení nejakej úlohy. Rovnako môže ukazovať aj polohu nepriateľa.
3. **Zameriavač** – stredový ukazovateľ, ktorý slúži na zameriavanie v priestore.
4. **Ukazovateľ času** – zobrazuje celkový čas hrania, ale môže napríklad zobrazovať aj odpočítavanie času pri časovo obmedzených úlohách.

Rozloženie jednotlivých komponentov GUI je znázornené na obrázku 12.



Obr. 12. Návrh GUI.

5 Prototypovanie

5.1 Ciele prototypovania

Identifikovali sme nasledujúce ciele, ktoré sú pre aplikáciu kritické a ktoré by sme teda chceli v rámci prototypovania vyskúšať aspoň čiastočne implementovať:

1. Realistické správanie komára.
2. Modelovanie objektov, ktoré sa budú nachádzať v hre.
3. Použitie Ogre3D na implementáciu grafického sveta.
4. Ovládanie komára pomocou bzučania.

Keďže jedným z cieľov hry je aj náučný charakter, komár by mal vyzerat' a aj sa správať čo najviac realisticky. Preto v rámci bodu 1 sme sa rozhodli analyzovať let komára, sily, ktoré na neho pôsobia, a vplyv používateľských akcií na jeho správanie. Túto analýzu sme potom chceli použiť aj v prototypu na určenie správania sa komára.

Bod 2 by mal overiť možnosť namodelovať objekty z hry (komár, budovy, ľudia a pod.) a vložiť ich do prostredia Ogre 3D. Body 1 a 2 by sme chceli spojiť a vytvoriť vďaka nim prvú verziu hry s ovládateľným komárom a čiastočným prostredím v rámci bodu 3.

V rámci bodu 4 chceme overiť ovládanie komára pomocou bzučania. Keďže ide ešte len o prototypovanie, táto časť bude samostatná a naším cieľom je len zachytávanie zvuku a určovanie jeho frekvencie.

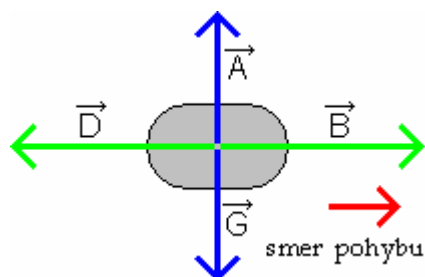
Opis jednotlivých častí prototypovania sa nachádza v nasledujúcich kapitolách.

5.2 Analýza letu komára

5.2.1 Sily pôsobiace na komára

Poznámka: v nasledujúcom texte budú použité názvy skalárov kurzívou a názvy vektorov tučným písmom.

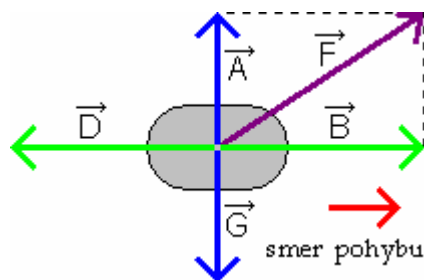
Kľúčovú úlohu v navrhovanej aplikácii zohráva komár, preto je dôležitým aspektom celej aplikácie simulácia jeho letu. Let vtáka a lietadla je založený na podobnom princípe. Základom letu sú štyri aerodynamické sily pôsobiace na objekt. Na obr. 13 je znázornené pôsobenie síl na objekt vykonávajúci rovnomerný pohyb bez zrýchlenia smerom doprava.



Obr. 13. Aerodynamické sily pôsobiace na letiaci objekt (pohľad z boku).

Sila **A** je vztlaková sila, **G** je gravitačná sila. **B** je hnacia sila a **D** trecia sila, spôsobená odporom vzduchu, pôsobiaca vždy smerom opačným ako je smer pohybu. Z orientácie trecej sily **D** môžeme dedukovať, že objekt na obrázku sa pohybuje smerom doprava. Hnacia sila vo všeobecnosti ale môže pôsobiť iným smerom ako je smer pohybu (napríklad počas zatáčania).

Dôležitou je otázka vzniku vztlakovej sily. Lietadlá majú krídla statické a pomocou nich je tvorená vztlaková sila. Hnaciú silu zabezpečujú motory lietadla. Hmyz produkuje mávaním krídiel ťažnú silu **F**, ktorú je možné rozdeliť na jednotlivé zložky ($\mathbf{F} = \mathbf{A} + \mathbf{B}$). Môžeme teda uvažovať o takýchto silách pôsobiacich na komára (obr. 14). Všimnime si, že smer letu nie je zhodný so smerom ťažnej sily.



Obr. 14. Sily pôsobiace na komára (pohľad z boku).

5.2.1.1 Ťažná sila

Veľkosť a orientácia ťažnej sily **F** je určená používateľom. Predpokladajme, že ťažná sila nesmeruje do strán, takže komár sa dokáže pohybovať len v zvislej rovine (vie stúpať, klesať, ísť dopredu a cúvať). Ťažná sila sa rozloží na zložku vztlakovú a hnaciú (určíme ich pomocou funkcií sínus, kosínus).

5.2.1.2 Trecia sila

Trecia sila **D** je orientovaná vždy proti smeru pohybu objektu. Veľkosť trecej sily vzduchu sa ráta zložitým vzorcom. Dôležité na tomto vzorci je, že veľkosť trecej sily je priamo úmerná *druhej* mocnine rýchlosti predmetu vzhľadom na okolie, *nezávisí* od hmotnosti predmetu a *závisí* od tvaru predmetu. Hustotu vzduchu budeme považovať za konštantnú.

Určenie trecej plochy komára nie je triviálne. Odvođením vzťahu medzi povrchom a hmotnosťou nahradíme vo vzorci neznámu plochu známym údajom – hmotnosťou. Pre jednoduchosť ďalej predpokladajme, že komár má tvar podobný kocke alebo guľi s rovnomernou hustotou. Plocha komára sa zväčšuje spolu s nákladom komára, ktorý tvoria napr. rôzne vylepšenia. Tiež predpokladajme, že všetky predmety, ktoré komár nesie, majú rovnakú hustotu ρ ($m = O \cdot \rho$, kde m je hmotnosť predmetu). Objem komára O je rovný tretej mocnine jeho priemeru d ($O = d^3$). Plocha jeho prierezu S je rovná druhej mocnine jeho priemeru ($S = d^2$). Potom vzťah medzi hmotnosťou m a veľkosťou trecej plochy S je

$$S = (m \cdot \rho^{-1})^{2/3}$$

Trecia sila **D** sa teda bude počítat ako

$$\mathbf{D} = -\mathbf{x} \cdot n \cdot m^{2/3} \cdot v^2 \quad (1)$$

kde \mathbf{x} je jednotkový vektor určujúci smer pohybu komára, n je konštantný koeficient trenia, m je hmotnosť predmetu a v je rýchlosť predmetu vzhľadom na okolie. Pri rovnakej hmotnosti sa rôzny tvar komára prejaví zmenou koeficientu trenia.

5.2.1.3 Gravitačná sila

Gravitačná sila \mathbf{G} je priamo úmerná hmotnosti komára. Jej smer je vždy konštantný.

$$\mathbf{G} = \mathbf{g} \cdot m \quad (2)$$

kde \mathbf{g} je konštantné gravitačné zrýchlenie a m je hmotnosť komára.

5.2.2 Rýchlosť komára

Ťažná sila \mathbf{F} je rovná súčinu zrýchlenia komára a jeho hmotnosti

$$\mathbf{F} = m \cdot \mathbf{a} \quad (3)$$

kde \mathbf{a} je vektor zrýchlenia.

Aplikovaním rovnice (3) na (1) a (2) teda vyplýva:

1. Komár s väčšou hmotnosťou potrebuje vyvinúť väčšiu ťažnú silu na dosiahnutie rovnakého zrýchlenia.
2. Tretia sila stúpa s objemom nákladu komára, a teda jeho maximálna rýchlosť bude s väčším nákladom klesať.
3. Gravitačná sila bude s hmotnosťou nákladu rásť, komár preto potrebuje väčšiu vztlakovú silu na jej prekonanie. To môže dosiahnuť zvýšením ťažnej sily (teda výkonu) alebo znížením uhla medzi ťažnou silou a zvislou osou (čo spôsobí pokles zrýchlenia v smere letu).

Pre výpočet letu v hre predpokladáme, že sa údaje neobnovujú spojito, ale v pravidelných časových intervaloch t_0 . Zmena rýchlosti $\Delta \mathbf{v}$ v tomto intervale je potom

$$\Delta \mathbf{v} = \mathbf{a} \cdot t_0 \quad (4)$$

5.2.3 Určenie pozície komára v priestore

Z praktického hľadiska nás nezaujímajú sily pôsobiace na komára, ale jeho pozícia v priestore v danom čase. Pre komára si budeme pamätať jeho pozíciu danú súradnicami x , y , z . V pravidelných intervaloch t_0 sa bude vykonávať nasledujúci výpočet:

1. Načítajú sa vstupné údaje – veľkosť ťažnej sily \mathbf{F} (zvukový vstup) a smer ťažnej sily \mathbf{x} (pozícia myši).
2. Určia sa veľkosti a smerovania jednotlivých síl (podľa vzťahov (1), (2)). Všetky sily rozložíme na zložky vo vodorovnom a zvislom smere.
3. Z veľkosti síl a hmotnosti komára sa určí zrýchlenie (podľa vzťahu (3)) vo vodorovnom a zvislom smere.
4. Z rýchlosti komára v predošlom kroku a zrýchlenia sa určí nová rýchlosť (podľa vzťahu (4)) vo vodorovnom a zvislom smere.
5. Z aktuálnej rýchlosti a pozície sa určí nová pozícia.

Tento postup platí za predpokladu, že sily **A**, **B**, **G**, **D**, **F** ležia v jednej rovine. Potom by komár vedel letieť dopredu, prípadne stúpať a klesať. Stále však nedokáže meniť smer - zatáčať.

5.2.4 Zatačanie komára počas letu

Predpokladajme, že kolmo na smer letu komára pôsobí sila, ktorej veľkosť je daná používateľom - dostredivá sila. Táto sila spôsobí, že komár sa začne pohybovať po kružnici. Pre jednoduchosť predpokladajme, že komár sa vie otáčať len vo vodorovnej rovine.

Na predmet pohybujúci sa po kružnici rovnomernou rýchlosťou pôsobí odstredivé zrýchlenie (rovnako veľké ako dostredivé zrýchlenie, $a_{\text{odst}} = a_{\text{odst}}$)

$$a_{\text{odst}} = v^2 \cdot r^{-1}$$

kde v je veľkosť rýchlosti predmetu, r je polomer kružnice. Vektor odstredivého zrýchlenia je kolmý na smer pohybu a smeruje od stredu kružnice.

Z veľkosti odstredivej sily F_{odst} je tiež možné vypočítať veľkosť odstredivého zrýchlenia a_{odst} (podľa vzťahu (2)). Platí

$$F_{\text{odst}} = m \cdot a_{\text{odst}}$$

a z toho pre polomer kružnice platí

$$r = m \cdot v^2 \cdot F_{\text{odst}}^{-1}$$

kde m je hmotnosť komára, v je rýchlosť pohybu a F_{odst} je veľkosť odstredivej sily.

Pri pohybe po kružnici sa mení otočenie komára. Opäť predpokladáme, že údaje neobnovujeme spojito, ale v pravidelných časových intervaloch t_0 . Zaujímá nás, o aký stupeň α sa komár otočí počas intervalu t_0 .

Obvod kružnice je $2 \cdot \pi \cdot r$, kde r je polomer kružnice. Ak sa komár pohybuje rýchlosťou v , potom túto vzdialenosť prekoná za čas T . Čas T je rovný t -násobku intervalu t_0 . Platí teda

$$2 \cdot \pi \cdot r = v \cdot t \cdot t_0 \quad (5)$$

Zároveň platí, že pri obletení celého kruhu sa komár otočí o 360° . V každom intervale t_0 sa komár otočí o uhol α , pričom celkovo sa o tento uhol otočí t krát. Platí

$$360^\circ = \alpha \cdot t \quad (6)$$

Z rovníc (5) a (6) dostávame, že komár sa v každom intervale t_0 otočí o uhol

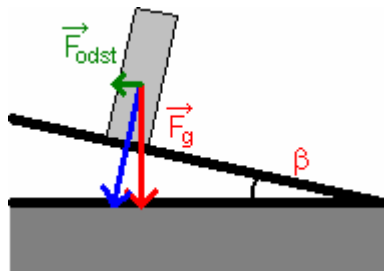
$$\alpha = 360^\circ \cdot v \cdot t_0 \cdot (2 \cdot \pi \cdot r)^{-1} \quad [7]$$

kde v je rýchlosť letu po kružnici (veľkosť priemetu vektoru rýchlosti komára do roviny kružnice) a r je polomer kružnice.

5.2.5 Naklonenie komára počas zatačania

Na predmet pohybujúci sa po kružnici pôsobí odstredivá sila (viď obr. 15). Tá spôsobuje, že predmet sa môže v dostatočne veľkej rýchlosti prevrátiť smerom von zo zákruty. V prípade, že sa predmet pohybuje po naklonenej zákrute, gravitačná sila nepôsobí kolmo na podložku, ale zvislo nadol. To môže naopak spôsobiť prevrátenie predmetu do vnútra zatáčky. Predmet sa na naklonenej dráhe neprevráti práve vtedy, keď výslednica odstredivej sily a gravitačnej sily prechádza cez základňu predmetu. Z veľkostí týchto síl je

možné vypočítať uhol β , o ktorý sa musí predmet (tj. v našom prípade komár) nakloniť, aby zostal v rovnovážnej polohe.



Obr. 15. Sily pôsobiace na teleso idúce po naklonenej zákrute.

Veľkosť gravitačnej sily je $F_g = m \cdot g$ (podľa vzťahu (2)). Veľkosť odstredivej sily je rovná veľkosti dostredivej sily. Veľkosť dostredivej sily je ovládaná používateľom. Pre uhol naklonenia β platí:

$$\tan \beta = F_{\text{odstr}} \cdot F_g^{-1} = m \cdot a_{\text{odstr}} \cdot (m \cdot g)^{-1} = a_{\text{odstr}} \cdot g^{-1}$$

5.2.6 Ovládanie komára používateľom

V predošlých výpočtoch sa za externe dané údaje považovali

- veľkosť ťažnej sily F v zvislej rovine letu
- smer ťažnej sily \mathbf{x}
- veľkosť dostredivej sily F_{dost} (ktorá je rovná odstredivej sile F_{odstr}) kolmej na smer letu

Tieto údaje zadáva používateľ prostredníctvom dvoch vstupov. Ide o zvukový vstup, ktorého výsledkom je frekvencia bzučania do mikrofónu. Tá bude priamo úmerná veľkosti celkovej ťažnej sily \mathbf{X} . Druhým vstupom je myš, ktorej poloha bude rozhodovať o tom, v akom pomere sa celková ťažná sila \mathbf{X} rozloží na hnaciu silu \mathbf{B} , dostredivú silu \mathbf{F}_{dost} a vztlakovú silu \mathbf{A} . Sily \mathbf{A} , \mathbf{B} a \mathbf{F}_{dost} sú na seba navzájom kolmé. Platí $\mathbf{A} + \mathbf{B} + \mathbf{F}_{\text{dost}} = \mathbf{X}$ a pre ich veľkosti platí

$$A^2 + B^2 + F_{\text{dost}}^2 = X^2 \quad (7)$$

Veľkosť celkovej sily \mathbf{X} je daná zvukovým vstupom. Zvislá poloha myši bude určovať percentuálnu veľkosť hnacej sily \mathbf{B} , vodorovná poloha myši bude určovať veľkosť dostredivej sily F_{dost} . Posledná zložka - veľkosť vztlakovej sily \mathbf{A} - sa vypočíta odmocnením zo vzťahu (7).

Poloha myši úplne hore znamená, že 100% sily \mathbf{X} je využitých ako ťažná sila a zvyšné sily sú nulové, teda $\mathbf{X} = \mathbf{A}$, $\mathbf{B} = \mathbf{F}_{\text{dost}} = 0$. Poloha myši v strede obrazovky znamená, že 0% sily \mathbf{X} je použitých na sily \mathbf{B} a \mathbf{F}_{dost} . Všetka sila \mathbf{X} je potom použitá ako vztlaková sila. Poloha myši úplne vpravo znamená, že celá sila \mathbf{X} je použitá ako dostredivá sila \mathbf{F}_{dost} na prudké zaobčenie smerom doprava.

5.2.7 Aplikácia analýzy v prototypu

Všetky vyššie uvedené vzorce sa použili v prototypu. Okrem toho sa aj kamera, ktorá sledovala komára, dokázala nakláňať nahor a nadol. Zvukový vstup bol substituovaný za

koliesko na myši. Používateľ môže zväčšiť alebo zmenšiť ťažnú silu otáčaním kolieska, inak zostáva veľkosť sily konštantná.

Ako čiastočný problém sa ukázalo otáčanie komára. Použitý vzorec je totiž vhodný pre vyššie rýchlosti, napríklad pre stíhačku. Pre helikoptéru, ktorá sa dokáže otáčať na mieste, sú vzorce problematické. Pri veľmi nízkych rýchlostiach by bol polomer zatačky taký malý, že by sa komár začal otáčať neuveriteľne rýchlo. V prototypy sa preto stanovil minimálny polomer zatačky, ktorú komár dokáže vykonávať. Ide o paralelu s autom, ktoré má tiež minimálny polomer zatačky.

Vyriešené je aj otočenie komára okolo zvislej osi a naklonenie pri zatačaní. Veľkou otázkou zostáva, ako natočiť kameru a komára nahor a nadol. V prototypy bola kamera orientovaná priamo v smere letu komára. To však nemusí byť najlepšie riešenie, pretože napríklad počas voľného pádu sa kamera otočí priamo k zemi. Smeru letu sa počítal na základe rozdielu predošlej pozície a aktuálnej pozície. Pri minimálnych rýchlostiach tak aj najmenší výkyv spôsobuje rapídne otáčanie kamery. Tento problém najviac vidieť pri štarte (nulová rýchlosť) a pri zaradení spiatočky. Okrem toho pri stúpaní sa kamera otočí smerom k oblohe, a používateľ vidí len modrú obrazovku. Aby sa tomuto problému predišlo, kamera sa nemohla dívať vyššie ako je úroveň horizontu.

Ovládanie komára je v prototypy takmer okamžité. Myš je totiž možné presunúť cez celú obrazovku v jedinom momente. Reálne lietadlo alebo vrtuľník ale vyžaduje na takýto manéver aspoň malý časový úsek. Tím sa už dohodol, že v ďalšej verzii sa bude používať imaginárna myš, ktorá bude sledovať pozíciu reálnej myši. Jej pohyb ale bude obmedzený maximálnou rýchlosťou, takže prudká zmena pozície myši má za následok dlhšie trvajúcu zmenu pozície imaginárnej myši.

Na záver je vhodné opäť spomenúť, že prototyp neobsahoval ovládanie zvukom. Frekvencia hlasu s veľkou pravdepodobnosťou nebude stála, a preto bude ovládanie ťažšie ako v prototypy. Prototyp tiež neobsahoval detekciu kolízií, takže ešte bude nutné vyriešiť nárazy do predmetov (respektívne nárazy predmetov do komára) [20].

5.3 Prototypovanie 3D modelovania

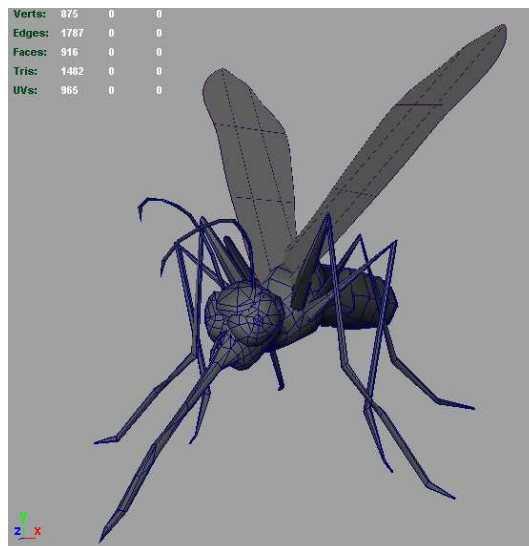
Keďže Ogre 3D je len grafický engine, všetky používané 3D objekty, modely, budovy, postavy a ostatné objekty je nutné vymodelovať v externom programe a následne ich upraviť do formy vhodnej pre export do prostredia Ogre 3D. Pri prototypovaní sme využili 3D modelovací nástroj *Maya* od firmy *Autodesk*. Ogre 3D poskytuje voľne dostupné exportéry pre rôzne verzie tohto nástroja od 6.5 do 8.5.

Účelom prototypovania bolo zistiť, či dokážeme externý trojrozmerný model, obsahujúci určité vlastnosti materiálu a animáciu, vhodne upraviť a prispôbiť pre prostredie Ogre 3D. Toto prostredie používa pre model formát súboru s príponou *.mesh* a pre materiály súbor s príponou *.material*.

5.3.1 Prototypovanie komára

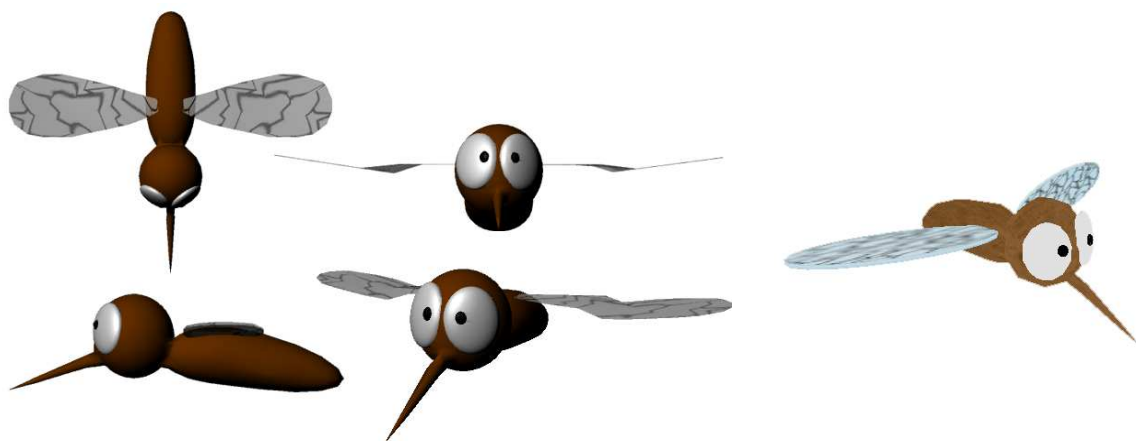
Prvou úlohou bolo vytvoriť model aktívnej hlavnej postavy, ktorú bude možné ovládať. Vytvorili sme teda polygónový model komára s malým počtom polygónov (obr. 16). Tento pôvodný model bol však vytvorený vo verzii, pre ktorú ešte neexistuje oficiálny funkčný exportér do Ogre 3D. Ukázalo sa, že ani nižšie verzie *Maya* nepodporujú otváranie súborov z vyššej verzie a nie je ani možné uložiť súbor do formátu kompatibilného s nižšou verzio. Keďže by časovo náročné vytvárať tento model nanovo v nižšej verzii, rozhodli sme sa

počkať na novú verziu exportéra, ktorá by sa mala objaviť začiatkom budúceho roku. V prípade problémov vytvoríme model do finálnej funkčnej verzie projektu nanovo.



Obr. 16. Pôvodný model komára.

Na prototypovanie sme teda použili veľmi zjednodušený model, na ktorom však dokážeme otestovať všetky kľúčové vlastnosti – teda exportovanie modelu, animácie a materiálov. Pritom tento zjednodušený model zachováva približne rovnaký počet polygónov, ako má pôvodný model (obr. 17). Model využíva UV mapovanie externej textúry a obsahuje jednu animáciu mávania krídel.



Obr. 17. Jednoduchý model komára určený na prototypovanie.

5.3.2 Prototypovanie iných 3D objektov

Keďže výsledný projekt bude obsahovať veľké množstvo rôznych objektov, ďalšou úlohou bolo vytvorenie iného modelu, konkrétne jednoduchého modelu budovy do prostredia. Vytvorili sme teda model budovy bez zadnej steny. Ako textúru sme použili obrazový súbor

vo formáte *.jpg* (rovnako ako v predchádzajúcom prípade) a využili sme UV mapovanie. Výsledok prototypovania budovy je na obrázku 18.



Obr. 18. Prototyp budovy do prostredia.

5.4 Prototypovanie grafického sveta

Cieľom prototypu grafického sveta bolo vytvoriť jednoduchú 3D scénu, v ktorej by hráč ovládal postavu komára, s použitím Ogre 3D.

Podarilo sa nám vytvoriť 3D scénu štyroch budov umiestnených do štvorca. Priestor medzi nimi pokrýva trávnatý povrch a nad budovami vidieť animovanú oblohu s oblakmi. V tomto svete sa pohybuje komár, ktorého pohyb je animovaný mávaním krídel.

Používateľ má možnosť ovládať pohyb komára pomocou klávesnice. Pohyb je zjednodušený na zmenu natočenia komára v 3D priestore a posun v smere alebo proti smeru jeho orientácie. Poloha komára je obmedzená na minimálnu a maximálnu výšku, v ktorej môže letieť. Na komára nepôsobí gravitácia.

Používateľ si ďalej môže zvoliť z troch polôh kamery:

1. Kamera umiestnená staticky za komárom.
2. Kamera za komárom, ktorá nasleduje komára s miernym odstupom. Táto kamera sa hýbe automaticky a plynulo, takže prudké zmeny pohybu komára (napríklad okamžité zastavenie) nespôsobia používateľovi šok.
3. Pohľad prvej osoby, pri ktorom nie je vidno postavu komára.

Počas tvorby prototypu bolo nutné vyriešiť niekoľko problémov. Prvý z nich sa týkal osvetlenia scény. Budovy neboli takmer vôbec viditeľné, hoci sme skúšali rôzne druhy i intenzity svetla. Nakoniec sme problém vyriešili zmenou svetelných vlastností použitých materiálov pre model budovy.

Ďalším odhaleným nedostatkom bola závislosť rýchlosti pohybu komára od počtu snímok (prekreslení scény) za sekundu. Tento sme odstránili tak, že pri pohybe a otáčaní komára i jeho animácii sme zmeny vykonávali na základe času, ktorý ubehol od poslednej snímky.

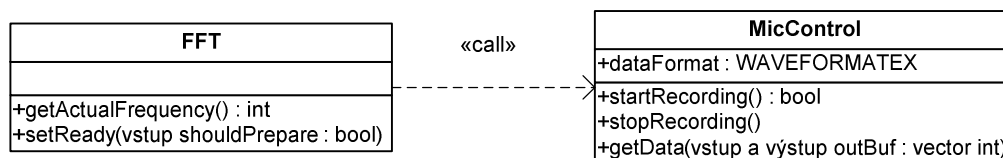
Ani realizácia otáčania hlavnej postavy do štyroch smerov nebola jednoduchá. Aplikované rotácie sa navzájom ovplyvňovali, čo spôsobovalo rotáciu do neželaných smerov. Riešenie sme našli v podobe vrátenia vertikálnej orientácie (dohora a nadol) komára to počítačového stavu pred každou zmenou horizontálneho natočenia (do strán). Potom sa aplikuje požadované horizontálne a následne i vertikálne natočenie.

Implementácia niektorých polôh kamery tiež nebola triviálna záležitosť. Tu sme s úspechom využili vlastnosti grafického enginu Ogre3D, konkrétne vzájomné naviazanie 3D objektov a ich spájanie do sústav. Hlavným bodom použitej sústavy objektov bola samotná hlavná postava. Na ňu bol naviazaný neviditeľný bod nachádzajúci sa v konštantnej vzdialenosti pred komárom, reprezentujúci smer jeho pohľadu. Za komárom sa nachádzala kamera, ktorá neustále sledovala neviditeľný bod pred komárom. Keďže spolu tvorili sústavu prepojených objektov, každé posunutie či otočenie komára sa automaticky aplikovalo aj na neviditeľný bod pred ním a na kameru. Pri pohľade prvej osoby sa skryl model komára a kamera sa umiestnila do rovnakej polohy ako komár. Pri kamere, ktorá sledovala komára s miernym odstupom, sme vektor posunu kamery pri sledovaní komára prenásobili konštantou z intervalu (0; 1), čo spôsobilo oneskorenie posunu.

5.5 Určovanie frekvencie zvuku

Kritickým bodom hry je ovládanie letu komára bzučaním. V rámci prototypovania sme sa rozhodli implementovať rozoznávanie frekvencie zvuku, ktoré je základom pre ovládanie bzučaním. Implementované boli dve triedy (obr. 19):

- trieda *FFT* – zabezpečuje analýzu zvuku a identifikovanie frekvencie zvuku
- trieda *MicControl* – zabezpečuje získanie zvuku prostredníctvom mikrofónu počítača



Obr. 19. Diagram tried pre prototyp identifikácie frekvencie.

Implementované triedy budú súčasťou komponentu *Interaction* modulu *Interface*.

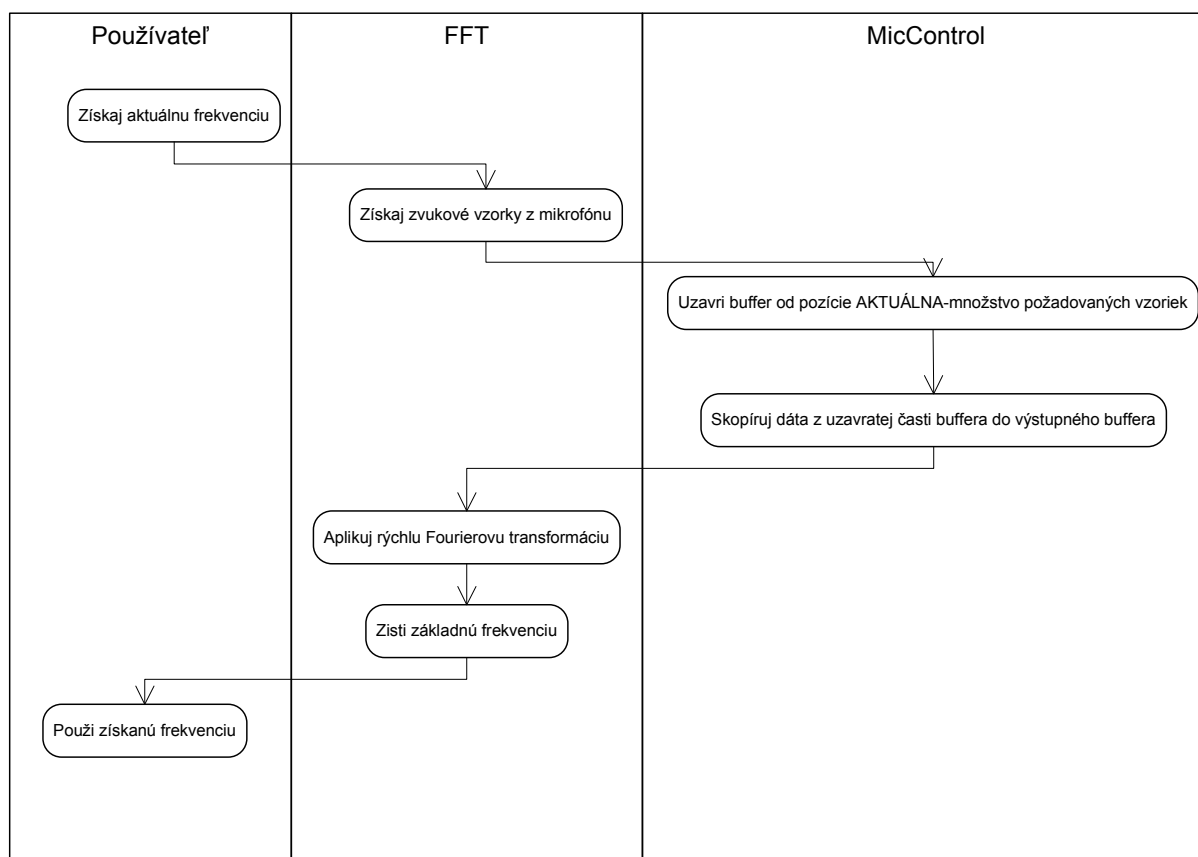
Odchytávanie zvuku v triede *MicControl* bolo implementované pomocou knižnice *DirectSound*. Základom je cyklický buffer umožňujúci zaznamenanie jednej sekundy zvuku. Po naplnení buffera sú postupne prepisované dáta na jeho začiatku, až kým sa opäť nenaplní. Dáta vo zvyšnej časti buffera zostávajú nedotknuté až po moment ich prepísania. Týmto postupom je zabezpečené efektívne narábanie s pamäťou. Trieda je schopná poskytnúť požadovaný počet vzoriek, ktorý nepresahuje veľkosť cyklického buffera.

Trieda *FFT* zabezpečuje transformáciu dát získaných pomocou triedy *MicControl* prostredníctvom Fourierovej transformácie, ktorou možno dekomponovať vstupný signál na frekvenčné zložky. Tieto zložky sú vyjadrené komplexnými číslami, ktoré zachytávajú nielen amplitúdu (veľkosť prvku), ale aj fázu vlnenia danej frekvencie (vyjadrenej ako arkus tangens podielu reálnej a imaginárnej zložky prvku). Dopredná a inverzná transformácia majú tvar:

$$t_k = \sum_{j=0}^{n-1} x_j e^{-2\pi i j k / n}, \text{ resp. } x_k = \frac{1}{n} \sum_{j=0}^{n-1} t_j e^{2\pi i j k / n}$$

kde t_k je k-ty prvok výstupného poľa, x_j je j-ty prvok vstupného poľa a n je veľkosť vstupného poľa. Vzhľadom na potrebu prevedenia transformácie v čase blízkom reálnemu bol využitý Cooley-Turkeyho algoritmus [21]. Využívaný je princíp „rozdeľuj a panuj“, keď je diskretná Fourierova transformácia aplikovaná na menšie časti poľa a pomocou motýlieho algoritmu sú výsledky čiastočných transformácií skombinované. Vstupom musí byť N-rozmerné pole zvukových vzoriek, kde N je mocnina čísla 2. Vďaka tomuto algoritmu je zložitosť Fourierovej transformácie redukovaná z $O(n^2)$ na $O(n \cdot \log n)$. Hľadaná frekvencia je index najväčšieho prvku prvej polovice poľa (druhá polovica je redundantná, čo vyplýva z Niquistovej redundancie).

Celková činnosť prototypu rozoznávania frekvencie a aktivity jednotlivých častí je zobrazená na obrázku 20. Používateľom v tomto prípade nie je človek, ale iná časť aplikácie používajúca zistenú frekvenciu (v našom prípade na let komára).



Obr. 20. Diagram činnosti pre prototyp identifikácie frekvencie.

Prototyp potvrdil správny prístup k riešeniu problému identifikácie frekvencie zvuku. Jeho výsledky boli porovnané s výsledkami profesionálneho voľne dostupného analyzátora frekvencie spoločnosti Reliable Software [22]. Z časového hľadiska je výpočet frekvencie uskutočnený v čase blízkom reálnemu aj pri veľkom počte zvukových vzoriek (testované na poli veľkosti 65536).

5.6 Výsledky prototypovania

Výsledkom nášho prototypovania je prvá verzia hry, na ktorú chceme nadviazať v letnom semestri. Výsledky jednotlivých častí prototypovania sú opísané v nasledujúcich kapitolách.

5.6.1 Realistický pohyb komára

Vďaka použitým vzorcom sa komár hýbe realisticky. Dôsledkom je napríklad realistická balistická krivka a maximálna rýchlosť pohybu ovplyvnená odporom vzduchu. Prototyp pohybu môžeme teda hodnotiť za úspešný.

5.6.2 3D modelovanie

Dokázali sme vytvoriť model obsahujúci animáciu a externú textúru a exportovať ho do formátu *.mesh*, ktorý je potrebný pre zobrazenie v prostredí Ogre 3D. Dokážeme vytvoriť a exportovať aj väčšie objekty s väčšími textúrami, teda nebude problém do výsledného projektu vytvoriť akýkoľvek potrebný model. Rovnako je možné využiť v projekte aj voľne dostupné objekty na internete. Pri vytváraní a používaní využijeme rovnaké postupy, ako sme otestovali pri prototypovaní.

5.6.3 Grafický svet

Prototyp dokázal, že knižnica Ogre 3D sa dá použiť pri implementácii celej hry. Umožňuje vkladanie animovaných modelov. Prídavné knižnice k Ogre 3D sú použiteľné aj na ďalšie funkcie, ako napríklad detekcia kolízií alebo model fyzikálnej interakcie. Tieto funkcie ale neboli v prototypu využité. V ďalšom semestri bude našim cieľom zakomponovať do programu aj tieto knižnice.

5.6.4 Určovanie frekvencie zvuku

Počas prototypovania boli zistené niektoré problémy súvisiace so získavaním zvuku pomocou knižnice DirectSound. Tieto súviseli so zlým uzavretím cyklického buffera pri opakovanom prístupe k nemu. Riešením bolo zväčšenie intervalov prístupu. Počas behu hry je však možné, že tento interval nebude môcť byť dodržaný, a preto bude musieť byť zefektívnený proces získavania zvukových vzoriek. Jednou z možností je aj využitie nižšie úrovňovej knižnice.

Celkovo treba hodnotiť výsledky prototypu za pozitívne, lebo nielenže bol implementovaný správny prístup k získavaniu frekvencie zvuku, ale boli odhalené aj riziká s ním spojené.

6 Použitá literatúra

1. Brooks, M.: No one understands me as well as my PC. In: New Scientist. 11/2003.
2. Lifeline, <http://www.mobygames.com/game/ps2/lifeline> (15.11.2007)
3. PiLfluS! – Speech recognition with gamers in mind, <http://www.pilflus.com.ar/download.php> (15.11.2007)
4. Pah! - The first voice-controlled game on the web, www.designer.co.il/pah/ (15.11.2007)
5. Shout n Dodge, <http://www.weebls-stuff.com/games/shout+n+dodge/> (15.11.2007)
6. Ford Game, <http://www.fordgame.be/> (15.11.2007)
7. Sing Pong - A Voice Controlled Pong, <http://www.singpong.co.uk/> (15.11.2007)
8. TrackIR, <http://www.naturalpoint.com/trackir/> (15.11.2007)
9. Cachya, <http://www.cachya.com/> (15.11.2007)
10. Eyetwig, <http://www.eyetwig.com/> (15.11.2007)
11. Mister Mosquito, http://en.wikipedia.org/wiki/Mister_Mosquito (15.11.2007)
12. Villoria, G.: Mister Mosquito – review. GameSpot. 03/2002. <http://www.gamespot.com/ps2/action/mistermosquito/review.html> (15.11.2007)
13. Mosquito Ops, <http://mosquitoops.damn6.com/> (15.11.2007)
14. McDonald, D., Erbacher, R. F.: Hardware Accelerated Graphics in Java. Proceedings of the International Conference on Geometric Modeling, Visualization & Graphics 2005. Salt Lake City. UT. July 2005. pp. 1708-1712.
15. Davidson, A.: Killer Game Programming in Java: Introducing Java 3D. In: *Gamasutra*. 12/2005 http://www.gamasutra.com/features/20051216/davison_02.shtml (15.11.2007)
16. DevMaster's Game and Graphics Engines Database, <http://www.devmaster.net/engines/> (15.11.2007)
17. Game Programming Libraries, <http://www.freeprogrammingresources.com/gamelib.html> (15.11.2007)
18. Collins, W.: Crystal Space or Ogre 3D, <http://www.arcanoria.com/CS-Ogre.php> (15.11.2007)
19. U.S. Department of Transportation, Federal Aviation Administration: Rotorcraft Flying Book. Flight Standards Service. 2004. <http://www.faa.gov/library/manuals/aircraft/media/faa-h-8083-21.pdf> (15.11.2007)
20. Detekcia kolízií, knižnica Ogre OPCODE, <http://conglomerate.berlios.de/wiki/doku.php?id=ogreopcode> (17.12.2007)
21. Rýchla Fourierova transformácia, <http://www.relisoft.com/Science/Physics/fft.html> (17.12.2007)
22. Analyzátor frekvencie, <http://www.relisoft.com/Freeware/index.htm> (17.12.2007)