

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Diplomová práce**

# **Analýza, návrh a vedení týmu v projektu webové hry**

Plzeň, 2010

Bc. Zbyněk Neudert



## **Poděkování**

Mé poděkování patří především Ing. Přemyslu Bradovi, MSc., Ph.D. za vedení mé práce, dále pak Ing. Romanu Moučkovi, Ph.D. jako zákazníkovi, pro kterého je výsledná aplikace určena. Za podnětné konzultace děkuji Ing. Petru Vaněčkovi Ph.D. a RNDr. Karlu Neudertovi, Ph.D. Za implementaci konceptu, který předkládám děkuji Petru Kellnhoferovi a dále Alexi Goubaremu za veškeré grafické návrhy.

Zvláštní poděkování věnuji své rodině a přítelkyni za podporu a trpělivost nejen během tvorby této práce, ale i během celého průběhu svého studia.



### **Prohlášení**

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni 17.5.2010

Bc. Zbyněk Neudert



## **Abstract**

### ***Analysis, design and team leading of webgame project***

This diploma thesis deals with a creation of an on-line web game for a focus group of teenage students also usable as a web presentation of Department of Computer Science and Engineering at University of West Bohemia in Pilsen. First, according to the assignment, properties of games and ways of their designing were studied. Next, the main focus was set on the ways of leading minor software projects with agile components and the corresponding tools. The subject, range and space-trade game properties were specified in the "gamedesign" document. After a selection of an appropriate technology for project development and its validation, the main analysis has been done and a necessary part of the concept was implemented. The last part of the thesis deals with the leading of a project team during the whole development process and with the description of support works. Analytic and design parts of the thesis are documented by software engineering methods using the UML language.





# OBSAH

Abstract .....	ix
OBSAH .....	xi
1. ÚVOD .....	1
2. PRINCIPY NÁVRHU WEBOVÝCH HER .....	3
2.1 Hry a jejich dělení .....	3
2.1.1 Dělení online her .....	4
2.1.2 Výběr vhodné hry k zadanému projektu .....	5
2.2 SPECIFICKÉ VLASTNOSTI MMO HER .....	5
2.3 RIZIKA A VHODNOST HER .....	6
2.3.1 Kvalifikační systém PEGI .....	8
2.3.2 PEGI Online .....	9
2.4 NÁVRH HER .....	10
2.5 SOFTWARE-INGENÝRSKÁ SPECIFIKA HER .....	12
3. PŘÍSTUPY K VEDENÍ SOFTWAREVÝCH PROJEKTŮ .....	15
3.1 Klasické vedení projektů a používané modely .....	16
3.2 Iterativní přístup k vývoji softwaru .....	17
3.3 Agilní přístup k vedení projektů .....	18
3.3.1 Metodiky agilního vývoje software .....	19
3.4 Nástroje pro agilní vedení projektů .....	21
3.4.1. IBM Rational Team Concert a platforma Jazz .....	23
4. SCÉNÁŘ WEBOVÉ HRY PRO KIV .....	25
4.1 Space Traffic – Gamedesign .....	25
5. VEDENÍ TÝMU PŘI REALIZACI PROJEKTU .....	33
5.1 Použité metodiky a nástroje .....	33
5.2 Sestavení a představení řešitelského týmu .....	36
5.3 Plánování, tvorba a rozdělování pracovních úkolů .....	38
6. TECHNICKÉ ZPRACOVÁNÍ PŘEDLOŽENÉHO KONCEPTU .....	43
6.1 Použití nástrojů pro definici vlastností projektu .....	43
6.2 Požadavky .....	44
6.3 Analýza .....	49
6.4 Výběr technologie a ověření vhodného výběru .....	51
6.5 Návrh .....	54
6.6 Implementace .....	55
6.7 Testování .....	56
6.8 Shrnutí objemu prací během iterací .....	57
7. DALŠÍ PRÁCE BĚHEM PROJEKTU .....	59
8. ZÁVĚR .....	61
9. PŘEHLED POUŽITÝCH ZKRATEK .....	65
10. POUŽITÉ ZDROJE INFORMACÍ .....	67

A	Plány jednotlivých iterací.....	69
B	Design reviews .....	75
C	Seznam grafických prvků hry a jejich priority .....	79
D	Zadání semestrální práce pro předmět KIV/DB2.....	81
E	Zadání semestrální práce pro předmět KIV/KPG .....	83
F	Poznámky testerů .....	85
G	Tutoriál ke hře .....	87

# 1. ÚVOD

Cílem této diplomové práce je nalezení vhodného herního tématu a jeho následné zpracování v takové formě, která bude vhodná pro prezentaci Katedry výpočetní techniky a informatiky Západočeské univerzity v Plzni. Motivací projektu se stala snaha přivést na webové stránky zmíněné katedry více středoškolských studentů, zvýšit jejich povědomí o katedře a tím také podnítit jejich zájem o studium na této škole. Jako vhodný způsob zviditelnění katedry byla zvolena webová hra, jejíž téma a zaměření nebylo blíže specifikováno. Jediným vodítkem pro další práci se tak stalo zadání v podobě interaktivní webové aplikace zaměřené na hráče ve věku 15-19 let.

Při hledání vhodné interaktivní aplikace bylo shledáno jako nejlepší řešení vytvořit webovou online hru pro více hráčů, která při vhodně zvoleném tématu a provedení dokáže přilákat cílovou skupinu. Bude-li hra určena pro současnou hru více hráčů, tzv. multiplayer, bude také podpořeno ústní předávání informací o hře v cílové skupině a hra by se tak mohla dostat do povědomí velkého množství středoškolských studentů.

Pro úspěšný průběh projektu bylo nutné stanovit rámcové téma hry, naleznout vhodné postupy a technické prostředky pro analýzu a návrh daného systému, sestavit tým řešitelů úlohy a nalézt vhodnou metodu vedení řešitelského týmu.

Předložená práce se nejprve věnuje obecnému návrhu her, jejich vlastnostem a specifikám jejich návrhu. Dalším bodem je pak představení různých metodik vedení menších softwarových projektů a nalezení nejlepšího postupu použitelného pro náš projekt. Po jeho nalezení je předložen gamedesign online webové hry pro více hráčů na jehož základě je provedena celá analýza a návrh aplikace. Druhá polovina práce se zabývá vedením řešitelského týmu, tvorbou podpůrných prací a v závěru také zhodnocením celého projektu.



## 2. PRINCIPY NÁVRHU WEBOVÝCH HER

Tato kapitola je věnována užitečným informacím, které pomáhají formovat další práci při vývoji webové hry. V několika následujících kapitolách se věnujeme problematice kategorizace her, specifikům her určených pro více hráčů a v neposlední řadě také rizikům her a principům jejich návrhu.

Poznámka: Na téma kategorizace počítačových her, jejich vlastností a vhodností pro hráče, existuje pouze minimální množství relevantních zdrojů. Práce proto na tomto místě využívá i populárně-naučných zdrojů a vlastních zkušeností autora.

### 2.1 Hry a jejich dělení

Přestože návrh dobré hry nezávisí na konkrétním typu hry, resp. na prostředcích její implementace, je vhodné znát možnosti realizace her a jejich druhy aby bylo možné hru později jednoznačně kvalifikovat. Tato kvalifikace usnadňuje oslovení cílové skupiny a podává základní informaci o tématu a provedení hry.

Pro naši práci není podstatné zabývat se jinými realizacemi her, než jsou hry počítačové. I zde však existuje několik různých typů realizací, vzájemně někdy značně odlišných:

- klasické hry – hry určené k instalaci na osobní počítač
- flashové hry – hry bez nutnosti instalace, často jednodušší oddychový charakter
- webové hry – hry, které lze hrát pouze prostřednictvím webového prohlížeče

Často se také můžeme setkat s kombinací zmíněných typů, např. webová flash hra. Všechny výše uvedené typy mohou (ale nemusí) umožňovat hru jednoho nebo i více hráčů. U posledních dvou typů se také často setkáme s nutností hry online.

Při kategorizaci her, je však pro hráče podstatnější informace o obsahu a tématu hry. K tomu jsou užívány ustálené pojmy, které však také nejsou pevně stanoveny, proto se můžeme setkat i s jinými než uvedenými typy:

- Strategie (RTS - Real Time Strategy) - viz. kapitola 2.1.2
- Střílečka (FPS - First person shooter) - akční hra z pohledu ovládané postavy
- Dobrodružné hry (Adventure) - hra založená na plnění úkolů
- Sportovní hry - hráč ovládá jednu postavu družstva
- Závodní hry - typicky automobilové závody
- Hry na hrdiny (RPG – Role Playing Game) - hráč se stává postavou a ovlivňuje děj
- Simulace - letecké a dopravní simulátory

Každá z uvedených kategorií má své zástupce i fanoušky. Ve světě online her se však nejčastěji setkáváme s RTS, FPS a RPG hrami.

### 2.1.1 Dělení online her

Dle zdroje [2] jsou do on-line kategorie her řazeny takové počítačové hry, které umožňují hraní po internetu. Hra je buď pro tento styl hraní přímo navržena, nebo nabízí tento způsob hry jako druhou možnost vedle hry pro jednoho hráče bez potřeby připojení k síti.

Hry vyvíjené přímo pro hraní po internetu (někdy označované zkratkou *OMG: online multiplayer game*) jsou děleny do několika kategorií:

- MUD (Multi-User Dungeon – tradiční hra pro více hráčů hraná prostřednictvím textového rozhraní)
- MMOG (Massively-Multiplayer Online Game – modernější obdoba s grafickým rozhraním; podle žánru se dělí na:
  - MMORPG (Massively-Multiplayer Online Role Playing Game)
  - MMORTS (Massively-Multiplayer Online Real-Time Strategy)
  - MMOFPS (Massively-Multiplayer Online First Person Shooter)
- Webové hry – zvláštní skupina her ovládaných pomocí webového prohlížeče

## **2.1.2 Výběr vhodné hry k zadanému projektu**

Jak již bylo řečeno v úvodu, tématem práce se měla stát online webová hra pro více hráčů. Tomuto záměru dobře odpovídá kategorie MMORTS ve skupině webových her. Pro bližší představu ještě uvedme přibližnou definici strategické hry.

Mezi strategie jsou řazeny takové počítačové hry, u kterých je ovládána větší skupina objektů a je s nimi manipulováno po hrací ploše tak, aby byla utrpěna co nejmenší újma na vitalitě, bojeschopnosti, údržbě objektů, zdrojů, území, jednotek, sil atp. Zároveň je tato újma působena protivníkům. Strategie jsou kategoricky děleny do několika sfér podle obsahu a cíle, jenž je dán. Jednou podkategorií strategických her jsou budovatelské strategie, které představují takový podžánr, v němž se hráč soustředí na budování a ekonomiku místo na boj. Představitelé tohoto žánru jsou tvoření hrami, ve kterých hráč jako plánovač a vůdce řídí město nebo osadu a usiluje o její rozkvět, jak uvádí [3].

## **2.2 SPECIFICKÉ VLASTNOSTI MMO HER**

MMO hry se v poslední době teší velké oblibě. Jejich základní vlastností, již z názvu vyplývající, je to, že jsou záměrně tvořeny pro hru více hráčů současně po internetové síti. Hra svým tématem i systémem hodnocení společnou účast více lidí ve hře výrazně podporuje, a to zejména tím způsobem, že staví všechny hráče do role spoluhráčů nebo protihráčů. Velmi často se také u MMO her setkáme s tím, že nejsou určeny pro soustavnou několikahodinovou hru, ale snaží se přinutit hráče hrát v krátkých časových intervalech několikrát denně.

Základním kamenem každé MMO hry je podpora sdružování hráčů do aliancí, part nebo gild. Hra může sdruženým hráčům nabízet různé výhody v podobě snadnější organizace společných akcí, bezpoplatkové obchodování, snadnější způsob komunikace uvnitř aliance apod. Cílem hráčů je potom vytvoření co nejsilnější nebo nejbohatší skupiny.

Další velmi častou možností uvedeného typu her je možnost vzájemných útoků mezi hráči. Společně s první jmenovanou vlastností jde o velmi silnou kombinaci, při jejímž dobrém navržení se může hra stát velmi úspěšnou i přes minimum jiných herních možností. Motivace

hráčů spojit se s přáteli a společně porazit skupinu nepřátel je hnacím motorem většiny MMOG, jako je populární Travian, Divoké Kmeny a jiné.

I přes uvedené nesporné výhody výše popsaných dvou vlastností her se můžeme setkat se hrami, které ne zcela odpovídají standardnímu schématu. Proti proudu v tomto ohledu jde například hra Ragnarok, která je založena pouze na principu vzájemné pomoci hráčů při boji s virtuálními nepřáteli, společném plnění úkolů a vzájemném obchodování.

Na závěr ještě také uveďme že MMO hry jsou téměř vždy vybaveny komunikačními nástroji a to v podobě živého chatu s možností skupinového i privátního zasílání zpráv a také je k těmto hrám velmi často přidruženo fórum, ať už alianční nebo veřejně přístupné.

## **2.3 RIZIKA A VHODNOST HER**

Riziky a vhodností počítačových her se tato práce zabývá proto, že v případě uvedení webové hry jako prezentace katedry KIV by nebylo vhodné, aby hra vzbuzovala odpor a protesty hráčů, jejich rodičů nebo pedagogů z důvodu nevhodného herního obsahu. Při návrhu herního obsahu konkrétní hry se proto práce snaží dodržovat zásady a doporučení uvedená v této kapitole.

Při hledání vhodných zdrojů ohledně rizik a přínosu počítačových her nebyla nalezena žádná vhodná publikace, která by se počítačovým hrám věnovala z pedagogického hlediska. Názory na tuto problematiku se tak velmi často formulují pouze jako příspěvky do různých článků na toto téma. Předložme proto alespoň hlavní myšlenky příspěvků [4] dvou magistrů psychologie do článku na serveru 21století.cz:

### **Waldo Šarý, magistr psychologie, Praha**

#### *Klady*

Násilí ve hrách poskytuje příležitost pro ventilaci napětí a je dobré pro odreagování nakumulované agresivity. (...) Výzkumy prokázaly, že odreagování kumulované agresivity je potřebné nejen pro psychické zdraví jedince, ale má i příznivý vliv na komunikaci s okolím. (...) Počítačové hry v tomto směru nabízejí nejlepší řešení. Dostatečné reálné zobrazení, přímo reagující na podněty hráčů, poskytuje jiným způsobem neuskutečnitelný pocit opravdovosti. A přitom masakrem hromady jedniček a nul v počítači nikomu (a ničemu) neublíží.



### *Zápory*

Samozřejmě je nutné věnovat pozornost věku hráče. Jsou hry, které jsou pro daný věk jednoznačně nevhodné. (...) Podstatný rozpoznávací faktor vhodnosti je to, jestli si hráč uvědomuje, že to, co se před ním odehrává na obrazovce, je pouze hra a nikoli skutečnost.

### **Monika Otmarová, magistra psychologie, Praha**

#### *Klady:*

Dítě se seznámením s konfliktními situacemi ve hře může v reálném životě vyvarovat traumatu, které by si mohlo přivodit naivním chováním. Někdy si lze násilnou hrou vybit i nahromaděný vztek, nicméně jestli je rozumná míra virtuální agrese přínosem, nebo hrozbou závisí případ od případu, v žádném případě nelze paušalizovat.

### *Zápory*

Dospělých zdravých osobností se týká nejvíce problém tzv. habitualizace, kdy se opakovaným dlouhodobým stykem s reálně vypadajícím násilím snižuje citlivost na násilí a agresi. Ty pak mohou být vnímány jako běžně akceptovatelné chování.

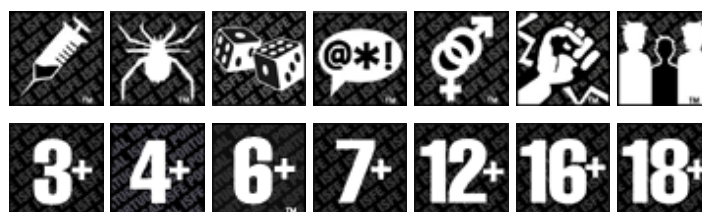
Násilí ve hrách však nejvíce ohrožuje děti do 12 let, které ještě nedovedou odlišit symbolickou a reálnou rovinu nabízeného obsahu. (...) Zkušenosti ze hry pak považují za rovnocenné i jinak nabytým zkušenostem, a nemusí proto někdy pozorovat rozdíl mezi fikcí a skutečností.

Roli při přejímání agresivního chování hraje i mechanismus identifikace, kdy se dítě ztotožňuje s postavou počítačové hry. V kůži svého idolu pak může alespoň symbolicky dosáhnout toho, co je pro něj ve skutečnosti nedostupné. Uspokojuje tak své vnitřní touhy např. po uznání, moci, síle... Toto prvotně symbolické uspokojení však někdy může vyvolat touhu aplikovat způsob chování ze hry i ve skutečném životě.(...)

### 2.3.1 Kvalifikační systém PEGI

Jedná se o systém celoevropské informace o hrách klasifikující vhodnost interaktivních her. Účelem systému [5] je zabezpečení přístupu mládeže pouze ke hrám, které jsou vhodné pro danou věkovou kategorii. Systém má podporu nejen všech hlavních výrobců hracích konzol, ale rovněž autorů her a vývojových programátorů interaktivních herních systémů po celé Evropě. Systém byl vyvinut Evropskou federací pro interaktivní software (Interactive Software Federation of Europe – ISFE). Cílem Evropské komise je nasazení tohoto systému jako modelu evropské harmonizace pro ochranu dětí a nezletilých osob na úseku společenských her. Systém PEGI byl uveden na trh v roce 2003 a nahrazuje předcházející národnostní klasifikační systémy vhodnosti her pro mládež. Klasifikace vhodnosti je uvedena na přední i zadní straně obalu interaktivních her.

Systém PEGI je složen ze dvou oddělených, vzájemně se doplňujících prvků. První z nich klasifikuje vhodnost pro věkové kategorie 3+, 7+ 12+, 16+ nebo 18+. Druhý z nich vyjadřuje jednu nebo více charakteristik hry. Ikony na zadní straně obalu pak tyto charakteristiky znázorňují. Pro vyjádření žánru hry obsahuje systém PEGI až šest ikon. Působnost charakteristiky hry poté odpovídá klasifikaci vhodnosti pro určitou kategorii mládeže. Kombinováním klasifikace věkové vhodnosti a charakteristiky hry je umožněno rodičům a osobám vybírajícím hry dosáhnout ujištění, že určitá hra je přiměřeně vhodná věku uživatele. PEGI je navržen tak, aby odpovídal měnícím se kulturním normám a postojům napříč světovými zeměmi. Je také podporován většinou vládních agentur příslušných států a všemi organizacemi obchodujícími s interaktivním softwarem.



Obrázek 2.1 - Piktogramy PEGI

(Význam piktogramů – horní řada: Drogy, Strach, Hazard, Vulgární výrazy, Sex, Násilí, Diskriminace;  
spodní řada: označuje spodní doporučený limit věku hráče)

Klasifikace PEGI je dobrovolná. Hodnocení je prováděno přímo členy herního průmyslu. Děje se tak prostřednictvím formuláře vlastního hodnocení. Ten je vyplněn vnitropodnikovým programátorem a na jeho základě hra získává hodnocení vhodnosti. Pro každou obsahovou kategorii je na základě odpovědí programátora stanoven přiměřený věk.

Hodnocení navržené vydavatelem je prověřeno nizozemským institutem pro hodnocení audiovizuálních médií (Netherlands Institute For The Classification of Audiovisual Media – NICAM). Hry, u kterých je navrženo hodnocení 16+ nebo 18+, jsou před potvrzením ohodnocení prověřeny. Hry s hodnocením 12+ a namátkově vybrané hry 3+ a 7+ jsou prověřeny po potvrzení ohodnocení. Na konci tohoto procesu udělí NICAM v zastoupení evropské federace ISFE licenci k užívání loga a možných charakteristik.

### 2.3.2 PEGI Online

PEGI Online je novým doplňkem výše uvedeného systému PEGI. Klade si za účel poskytnout mládeži větší ochranu před nevhodným herním obsahem [6]. Licenci k vyobrazování loga PEGI Online uděluje správce tohoto systému jakémukoli poskytovateli online her, jenž splňuje požadavky stanovené kodexem bezpečnosti PEGI Online (PEGI Online Safety Code – POSC). Požadavky se týkají povinnosti chránit webové stránky před nelegálním a nepřístojným obsahem vytvářeným uživateli a nežádoucími odkazy, jakož i zajištění opatření na ochranu mladých lidí a jejich soukromí v průběhu hraní online her.



*Obrázek 2.2 Logo PEGI online*

Kodex bezpečnosti POSC je zpracován za účelem podpory minimální úrovně ochrany, jež by měla být mladistvým poskytována v prostředí online her. Správci systému, kteří přijmou tento kodex, se zavazují k odstranění nevhodného materiálu ze svých stránek a k zajištění patřičného chování uživatelů. V důsledku toho jim je povoleno, po registraci v systému PEGI, zobrazovat na stránkách svých webových aplikací logo PEGI Online (obrázek 2.2).

Aby systém zobrazující logo PEGI Online splnil hlavní ustanovení kodexu, musí obsahovat pouze hry, jejichž obsah byl patřičně ohodnocen regulérním systémem PEGI. Hráčům musí být umožněno ohlásit existenci nevhodného obsahu na jakýchkoliv souvisejících webových stránkách. Držitelé licence jsou dále povinni zajistit, aby spravované online služby byly chráněny před nelegálním, urážlivým, rasistickým, ponižujícím, korupčním, výhružným a obscénním obsahem. Jakýkoliv držitel licence PEGI Online je povinen shromažďované osobní informace od uživatelů chránit v souladu se zákony Evropské unie. Dále jsou držitelé povinni při reklamní politice uplatňovat zásady tak, že veškeré reklamy musí přesně odrážet povahu a obsah prezentovaného produktu, případně i jeho hodnocení. Reklamy musí být vytvářeny se smyslem pro zodpovědnost vůči veřejnosti. Nesmí obsahovat žádnou skutečnost, jež by mohla být příčinou závažného či rozsáhlého deliktu. Reklamy s hodnocením 16+ nebo 18+ nesmí být zaměřeny na ty, kteří jsou mladší. Další doplňkové či samostatné produkty, jež jsou propagovány spolu s hlavním produktem, musejí být vhodné pro uživatele, pro něž je určen hlavní produkt. Držitelům licence PEGI Online je dále doporučeno, aby prostřednictvím všeobecného prohlášení veřejnost informovali o sponzorování a/nebo reklamě související s jakoukoliv online službou.

## **2.4 NÁVRH HER**

K navržení zábavné a pro hráče zajímavé hry neexistuje žádný univerzální návod, stejně jako je tomu u návrhu jiného softwarového produktu. Bude-li existovat takový návod, budou všechny aplikace a hry dokonalé a nebude nutné se jejich vývojem dále zabývat.

Návrhem kvalitních her se velmi podrobně zabývá Jesse Schell v knize *Art of GameDesign* [7]. Prvotním předpokladem úspěchu je podle Schella abstrakce hry od následné implementace a to dokonce v tak silném významu, že použité téma, motiv a pravidla hry bude možné použít jak pro realizaci počítačové, tak například stolní společenské hry. O účinnosti této zásady se můžeme přesvědčit například na dvojici her *Dungeon and Dragons* (v české mutaci *Dračí doupe*) a *Baldur's Gate*, kde první jmenovaná je úspěšná stolní společenská hra a druhá neméně úspěšná hra počítačová.

Časté testování prototypů, jako jeden ze základních nástrojů ke zlepšení výsledné hry, bylo uvedeno již v předcházející kapitole. Postupujeme-li ovšem v návrhu tak, že abstrahujeme

od implementace, je naším nástrojem ke zlepšení také testování různých realizací hry, a proto byla v rámci našeho projektu testována také stolní varianta hry.

Kniha Art of Gamedesign podává velmi podrobné informace a doporučení k sestavení dobrého herního systému a využívá k tomu sto různých úhlů pohledu na konkrétní herní záměr. Tyto pohledy jsou velmi cenným námětem k zamyšlení nad vznikající hrou a pomáhají herní design posunovat kupředu. Pro názornost uvedme úhel pohledu #2 – Pohled překvapení a úhel pohledu #4 – Pohled zvědavosti.

## **#2 – Pohled překvapení**

Překvapení je základ, ale často na něj lze zapomenout. Použij tento pohled aby sis připomněl, že tvá hra má být plná zajímavých překvapení.

- Co překvapí hráče při hraní této hry?
- Je v herním příběhu překvapení? Je v pravidlech? Ve zpracování? V technologii?
- Umožňují pravidla hráčům překvapit jiné hráče?
- Může být hráč sám ze sebe překvapen?

Překvapení je základním kamenem zábavy – je to kořen humoru, strategie a řešení problémů. Lidský mozek je určen k tomu, aby si užil překvapivé situace, což je vědecky dokázáno. Pomocí mozkového scanu bylo také dokázáno, že pokud se děje nějaké nemilé překvapení, mozková centra „rozkoše“ jsou aktivní.

## **#4 – Pohled zvědavosti**

Zamysli se nad skutečnou motivací hráče. Ne nad cílem hry, ale nad tím, co hráče nutí těchto cílů dosáhnout.

- Jaké otázky vkládá moje hra do myšlenek hráče?
- Co můžu udělat pro to, aby se o tu myšlenku musel více zajímat?
- Jak mohu hráčům nabídnout více takových otázek?

Příklad: Videohra má časový limit na level. Hráč se tedy sebe ptá: Jak najdu cestu tímhle bludištěm za 30 sekund? Tuto otázku můžeme v jeho mysli upevnit třeba animací na konci každého levelu, takže v hráči probudíme otázku: Jaká bude asi animace na dalším levelu?

Při návrhu her se dále řídíme osvědčenými zásadami, mezi než patří využít zažité zkušenosti hráče z jiných her, aby si hráči na naši novou hru rychleji navykali. Tato zásada je opět dvojnásobně důležitá u webových her, protože nejen hráči onlinových her, ale i běžní uživatelé internetu jsou podvědomě zvyklí hledat určité navigační prvky na konkrétních místech.

Přestože při návrhu her konfrontujeme své nápady s již existujícími hrami, je velmi žádoucí přidat do hry něco nového, čím hru odlišíme od jejích konkurentů. O úspěšnosti tohoto kroku se můžeme opět přesvědčit na příkladu z herní minulosti. Při uvedení hry Istar v roce 1992 byly vůbec poprvé použity vztahy mezi jednotlivými postavami ve hře. Poprvé bylo možné se setkat se situací, kdy se do sebe dvě postavy ve hře zamilují nebo se začnou nesnášet a odmítnou společně plnit nějaký úkol. Dnes patří tyto vztahy postav ve hře k základu každé počítačové hry na hrdiny (RPG).

Dále, vzhledem k tomu že úspěch hry je založen na velkém počtu hráčů, je nutné již při návrhu myslet na speciální promo akce a bonusy určené aktivním šířitelům povědomí o hře. Známými způsoby tohoto odměňování jsou například bonusy za přivedení nového aktivního hráče, což je způsob pravděpodobně nejpoužívanější a setkáme se s ním v téměř každé online hře. Jiná situace by poté nastala u jiných než webových implementací. Zde by úlohu co největšího rozšíření hry převzala marketingová strategie realizačního týmu nebo firmy.

## **2.5 SOFTWAREVĚ-INŽENÝRSKÁ SPECIFIKA HER**

Na první pohled by se mohlo zdát, že hry jsou softwarový produkt jako každý jiný a není tedy nijak specifický ani z hlediska softwarově inženýrského. Opak je ovšem pravdou. Klasické hry určené k instalaci na stolní počítač, které ještě navíc umožňují současnou hru více hráčů, se lehce blíží realtimeové aplikaci. U klasických her tedy nelze použít třívrstvou architekturu, nejčastěji používanou při realizaci běžných programů. Hra si nemůže dovolit jakékoli zpoždění, načítání apod. Všechny operace musí probíhat hladce a především co nejrychleji. Proto je její architektura podřízena požadavku maximalizace rychlosti.

Dalším specifikem práce na vývoji her je velké množství prací, které musí ze své podstaty proběhnout striktně sekvenčně. Tato situace nás později limituje při výběru vhodné metodiky, dle které budeme projekt řídit. Příklad takovýchto činností znázorňujeme na obrázku 2.3, který byl převzat ze zdroje [8]. Tentýž zdroj také podává doporučení, jak modifikovat známé

metodiky vývoje tak, aby byl čas vývoje co nejkratší a minimalizoval kód, který nebude použit. Pro vývoj her je zde doporučováno rozvržení práce do různých úrovní (například dle zaměření práce nebo dle levelu hry) a postupné posouvání těchto úrovní jednotlivými „pracovišti“. Na práce, které zabírají výrazně více času, se doporučuje přidělit více lidí, kteří se o práci podělí. Tím je zajištěn plynulý průchod úrovní jednotlivými fázemi zpracování a uspořený čas se odhaduje až na 46%.



Obrázek 2.3 – Znárodnění sekvenčnosti tvorby počítačových her

Opustíme-li výše uvedenou kategorii realizace her a zaměříme se na hry flashové nebo webové, nedostaneme se už do tak velkých problémů. Zde můžeme hovořit o podobném, ne-li zcela stejném systému, jako je jakýkoli jiný běžný software. Ovšem více než kde jinde se u návrhu a implementaci her setkáme s nutností častého a důsledného testování, jehož výsledky často mění plány příštích prací. Výsledky testování však neznamenají pouhý seznam bugů, ale vyžadují se od testerů také návrhy a připomínky k hratelnosti a zábavnosti hry, které jsou klíčem ke zdokonalení vytvářené aplikace. O hratelnost a zábavu jde především a proto potřebujeme především hráče, nikoli programátory a architekty k otestování vznikající aplikace.

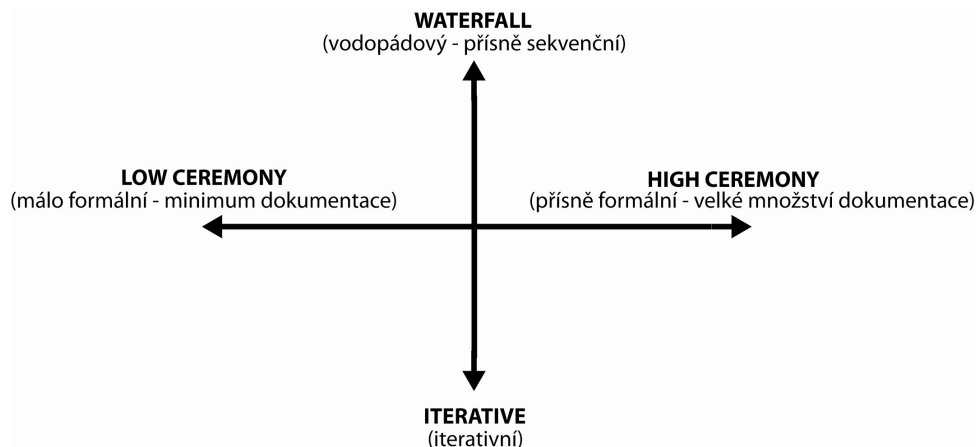
Poslední lehce specifickou záležitostí je uživatelská přístupnost finální aplikace. O to nám samozřejmě jde při vývoji jakéhokoli softwarového díla. Ovšem pracujeme-li například na interním informačním systému pro středně velkou firmu, lze některé nepřilíš uživatelsky přívětivé operace uživatele naučit, protože firma po zaplacení produktu nebude ihned shánět uživatelsky příjemnější alternativu. Opačná situace je ovšem u počítačových her. Pokud se hráč ve hře neorientuje nebo pro něj práce s aplikací není příjemná, aplikaci po krátké chvíli opustí a hledá jinou alternativu. Dvojnásobně toto pravidlo platí ve hrách hratelných zdarma.





### 3. PŘÍSTUPY K VEDENÍ SOFTWAREVÝCH PROJEKTŮ

Cílem této kapitoly je představení různých přístupů k vedení softwarových projektů. Proces vývoje softwarového projektu se může v zásadě ubírat různými cestami k dosažení stejného cíle. Variabilitu přístupu k vedení projektů názorně ukazuje obrázek 3.1 převzatý ze zdroje [9].



Obrázek 3.1 – Mapa druhů přístupu k vývoji softwarového projektu (Process Map)

Z obrázku 3.1 je zřetelné, jaké vlastnosti vybrané metodiky nás nejvíce zajímají. Jedná se především o vodopádový (zcela sekvenční) nebo iterativní proces vývoje a o množství dokumentace, které je v průběhu projektu produkováno.

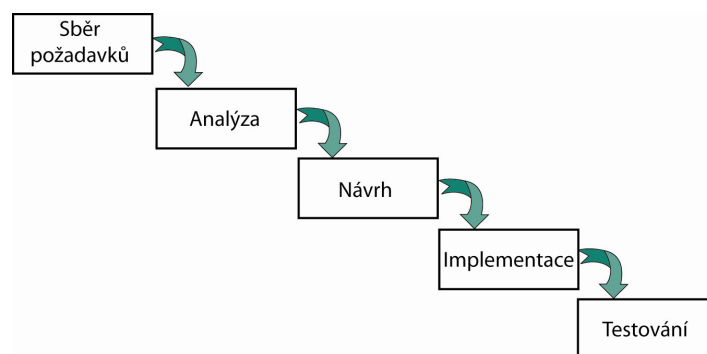
Jakékoli dále popisované metody vedení vždy pracují s pěti základními činnostmi, kterými jsou:

- Sběr požadavků - zjišťujeme, co zákazník chce
- Analýza - na základě požadavků analyzujeme chování programu
- Návrh - na základě analýzy navrhujeme možnou implementaci
- Implementace - tvorba zdrojového kódu a stavba software
- Testování - ověření funkčnosti vytvořeného kódu

### 3.1 Klasické vedení projektů a používané modely

V rámci předloženého textu rozumíme klasickým vedením takový způsob vedení projektu, který se blíží vodopádovému modelu, nepracuje přírůstkově (iterativně) a nepřijímá žádné změny během projektu. Nejčastěji používanými modely pro takto vedené projekty jsou vodopádový model a evoluční model.

Vodopádový model, tak je chápán a představován dnes, je přesně vystižen níže na obrázku 3.2, ze kterého je zřejmé, že při použití vodopádového modelu jsou všechny realizační práce prováděné striktně sekvenčně pro celý vytvářený systém najednou<sup>1)</sup>. V případě tvorby softwarové aplikace to znamená, že je celý systém nejprve analyzován a navržen, poté je v jednom kuse implementován a celý hotový systém je teprve testován. Od začátku projektu tedy musí být přesně specifikován předpokládaný cíl projektu a jeho vlastnosti, přičemž nelze v žádném případě akceptovat jakoukoli změnu plánu počínaje okamžikem startu práce na projektu.



Obrázek 3.2 – Vodopádový model procesu vývoje

Z uvedených vlastností vodopádového modelu vyplývá, že se hodí pro malé projekty, kde můžeme dosáhnout přesné specifikace produktu již na začátku projektu. Za výhodu tohoto přístupu můžeme považovat fakt, že preimplementační fáze vyžaduje kompletní dokonalou dokumentaci, což vede k výhodám při personálních změnách realizačního týmu a eliminuje chybné kroky v implementaci.

---

<sup>1)</sup> vodopádový model původně představený Winstonem Roycem [10] nepracoval s celým systémem najednou, ale pracoval vždy na určité části a byl tak ve své podstatě iterativní.

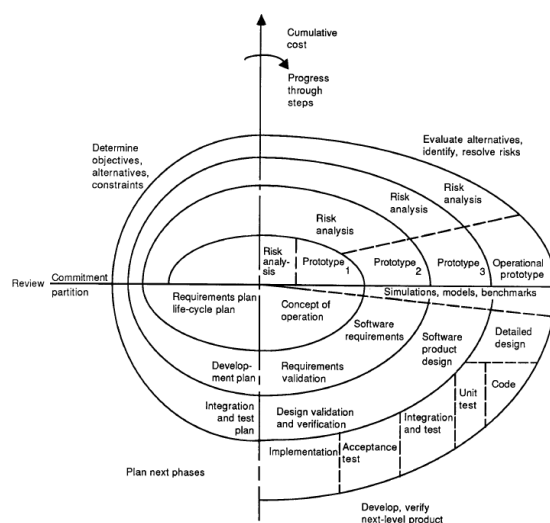
Při personálních změnách se noví členové snadněji začlení do projektu prostudováním předchozích dokumentací a v jakékoli následující fázi projektu se můžeme spolehnout na bezchybné provedení předchozí fáze a to právě proto, že bez toho nemůže být vodopádový model funkční. Vodopádový model se proto nachází v pravém horním kvadrantu mapy uvedené na obrázku 3.1.

Popsané výhody však můžeme jen s těží použít při vytváření středně velkých a velkých projektů. Téměř vždy se totiž dostaneme do situace, kdy zjistíme, že by bylo lépe některou část předchozího stupně vývoje přepracovat. Je potom snadnější změnit například návrh implementovaného systému než trvat na realizaci obtížně řešitelného návrhu.

Drobnou změnou k lepšímu je poté model sašimi, u kterého se jednotlivé fáze vodopádového modelu zčásti překrývají. Dalším krokem ke zlepšení je použití evolučního modelu, u kterého dochází k aplikaci vodopádového modelu na prototypy výsledného systému, který je zahozen. Teto prototyp slouží k lepšímu pochopení požadavků zákazníka tak, aby mohl být v příštím kroku použit vodopádový model.

### 3.2 Iterativní přístup k vývoji softwaru

Iterativní procesem vývoje softwaru se rozumí takový postup vývoje při kterém jsou postupně tvořeny stále větší části systému dokud není systém kompletní. Nejkomplexnějším příkladem takového přístupu je spirálový model popisovaný Barry Boehmem [11], znázorněný na obrázku 3.3.



Obrázek 3.3 – Schéma spirálového modelu

Při aplikaci spirálového modelu tedy několikrát zopakujeme základní činnosti pro vývoj softwaru, jako je analýza rizik, požadavků, implementace prototypu a jeho testování, dokud nedosáhneme zákazníkem očekávaného a smluveného výsledku.

Zvláštním případem iterativního přístupu k vedení projektu jsou agilní metodiky, které by se na obrázku 3.1 nacházely v levém spodním kvadrantu. Agilnímu přístupu k vývoji softwaru se práce věnuje v následujících kapitolách.

### **3.3 Agilní přístup k vedení projektů**

Agilní metodiky vývoje [12] se začaly objevovat na konci devadesátých let minulého století v mnoha rozličných formách jako reakce na to, že většina softwarových projektů nekončila včas, v rámci rozpočtu, případně nedodala zákazníkovi to, co očekával.

Při překladu výrazu agilní metodiky se setkáváme s termínem pružná nebo hbitá metodika, tedy taková, která přivítá změnu. Tento kostrbatý překlad však přesně vystihuje podstatu agilního přístupu. Agilní přístup k vývoji softwaru si neklade za cíl analyzovat celý vytvářený systém na počátku práce, ale právě naopak se soustřeďuje jen na relativně malé jádro se základní funkčností. Po jeho úspěšné implementaci je vytvořená část systému testována, a tím jsou sbírány další požadavky, které se stávají počátečním artefaktem nové iterace.

Tento přístup je velmi užitečný už jen z toho důvodu, že zákazník velmi často neví, jaký produkt vlastně chce. Postupným vytvářením systému a konzultacemi se zákazníkem získáme postupně přesnou představu o jeho požadavcích a můžeme tak dodat softwarové řešení plně odpovídající jeho představám.

Pro srovnání agilních metodik [13] a klasických přístupů nám nejlépe poslouží tři faktory, které ovlivňují výsledný produkt. Těmito faktory jsou funkcionality, čas (termín dodání) a zdroje (finanční i lidské). Zatímco klasické metodiky, definující celý vytvářený systém na počátku, mají funkcionality fixní a zdroje a čas variabilní, agilní metodiky jdou přesně opačnou cestou. Za fixní jsou zde považovány zdroje a čas, přičemž očekávaná funkcionality se v průběhu celého projektu mění. Namísto dokumentace je kladen největší důraz na komunikaci se zákazníkem, aby se předešlo situaci, kdy zákazník ve výsledku obdrží systém, který nechtěl. Častá komunikace a diskuze nad dodanou částí dokáže rychle reagovat na změnu v požadavcích, a tato změna téměř jistě v každém větším projektu přijde.

Průběhy jednotlivých iterací při agilním vývoji probíhají v principu podobně bez ohledu na konkrétní použitou metodiku. Na počátku je vždy proveden sběr požadavků libovolnou formou, a to od papírových štítků po speciální software. Dále je provedena analýza těchto požadavků a jsou tak stanoveny úkoly, které je nutné řešit. Opět lze použít celou řadu nástrojů z nichž nejznámějším je tzv. backlog. Do něj jsou zapsány všechny případy použití systému, jsou jim přiřazeny priority a náročnosti a řešitelé projektu z něj jednotlivé úkoly vybírají a plní je. Současně lze do backlogu zapsat problémy, které brání vyřešení některého úkolu. Tímto způsobem pomáhá tento nástroj i při následujících fázích iterace, tedy implementaci a testování. Jednotlivé základní činnosti, tak jak jsou uvedeny na počátku kapitoly 3 neprobíhají za sebou sekvenčně, ale probíhají současně a vzájemně se ovlivňují a doplňují.

Pro plánování průběhu projektů a iterací při agilním vývoji se často využívá pomůcka zvaná Burndown chart. Jedná se o prosté grafické znázornění, kolik úkolů je třeba zvládnout v jakém čase a jak by měly vyřešené úkoly postupně přibývat v průběhu iterace. Současně se do toho samého grafu znázorňuje, kolik úkolů je v daný den (nebo jiná časová jednotka) hotovo a jaká je rychlost týmu. Znalost rychlosti týmu je využívána při plánování budoucích prací a iterací.

### **3.3.1 Metodiky agilního vývoje software**

Pro agilní přístup k vedení projektu můžeme využít několik existujících metodik, které mají společné hlavní myšlenky uvedené v nadřazené kapitole. Mezi nejznámější metodiky patří XP, SCRUM a DSDM, dále bude popsána i metodika FDD.

#### **Extrémní programování - XP**

Název metodiky [14] zde opět vcelku přesně popisuje styl, jakým je výsledný produkt vytvářen. Nízké nároky na dokumentaci a vysoké nároky na funkční zdrojový kód jsou zde dotaženy do skutečného extrému. Při extrémním programování je nositel všech informací samotný zdrojový kód, dokumentace prakticky neexistuje. Dobrá komunikace mezi členy řešitelského týmu a mezi řešiteli a zákazníkem je jedním z naprosto klíčových faktorů úspěchu. Dodávky nových funkcí systému pro zákazníka probíhají prakticky denně, stejně jako testování. Programátoři v jednom pracovním dnu nejprve napíší testy pro plánovanou funkčnost,

poté ji implementují. Projde li systém testem je celý proces opakován. Výsledkem je perfektní kód a 100% odladěná funkčnost.

## **SCRUM**

Jakožto pravděpodobně nejpoužívanější metodika agilního přístupu, Scrum [13] rozděluje práci na projektu do několika kratších intervalů (sprinty), které trvají zpravidla okolo třech týdnů. Na začátku sprintu jsou stanoveny úkoly a cíle, kterých je třeba v dané iteraci dosáhnout. Dále probíhají každý den krátké schůzky, na kterých si jednotliví členové řešitelského týmu (programátoři) vybírají úkoly, které budou následující den řešit. Na začátku dalšího dne se výběr úkolů opakuje a vždy jsou také předneseny překážky v implementaci, které byly předchozí den objeveny.

Cíle každé iterace musejí být odsouhlaseny všemi členy řešitelského týmu. Tým se v podstatě řídí sám a odpadá zde role vedoucího. Na první pohled se může zdát tato záležitost šokující, ale praxe ukazuje, že přenechat zodpovědnost za projekt programátorům může být vhodným prostředkem, jak efektivně dosáhnout chtěného výsledku.

## **Dynamic Software Development Method - DSDM**

Za vývojem metodiky DSDM [15] stojí celé konsorcium šestnácti organizací. Metodika je zajímavá tím, že autoři k ní dodávají i *podpůrné vývojové prostředí (framework)*. Metodika obsahuje řadu propracovaných technik spolu s doporučeními o jejich použitelnosti a zacílení. Metodika se vyznačuje propracovaným vývojovým cyklem, který probíhá iterativně jak na úrovni hlavních fází, tak i uvnitř fází. Mezi hlavními fázemi je možné se i vracet, takže práce jsou vždy směřovány do místa, které je v dané chvíli klíčové. Nevýhodou je, že metodika není zcela „zdarma“ – je třeba zaplatit členský příspěvek konsorciu.

## **Feature Driven Development - FDD**

V čistém překladu vývoj řízený užitnými vlastnostmi daného software. Narozdíl od metodiky XP zachovává tato metodika fázi modelování systému [16]. Za základní model je zde považován doménový model na vysoké úrovni abstrakce, který popisuje celý systém a slouží tak k minimalizaci problémů integrace a kolaborace jednotlivých částí vytvořených různými programátory. Práce na projektu je poté rozdělena do zpravidla

dvoutýdenních iterací, přičemž v každé iteraci jsou navrhovány a implementovány konkrétní užité vlastnosti systému (features). Dalším rozdílem od metodiky XP případně SCRUM je fakt, že jednotliví programátoři si sami nevybírají práci, kterou budou řešit. Každému z nich je přidělena zodpovědnost za část doménového modelu – za jeden objekt. V případě, že řešená vlastnost vyžaduje spolupráci více objektů s rozdílnými vlastníky, je vytvořen tým pro řešení dané vlastnosti – feature team.

### **3.4 Nástroje pro agilní vedení projektů**

Z výše uvedených vlastností agilního přístupu k vedení projektů vyplývá, že pro úspěšné vedení projektů agilní metodikou je vhodné zajistit dostatečnou softwarovou podporu především třech základních úkonů, jimiž jsou správa verzí, zápis chyb/zápis úkolů (bug reporting/task reporting) a vzájemná komunikace. Jednou možností realizace této podpory je společné používání několika freewarových nástrojů nebo užití některého z placených nástrojů, které se snaží poskytnout komfortní podporu všech zmíněných úkolů.

Nyní krátce představíme nejnámější nástroje poskytované zdarma a podáme základní informace o placeném nástroji Team Foundation Server společnosti Microsoft. Ve zvláštní kapitole se poté budeme věnovat platformě Jazz a nástroji IBM Rational Team Concert, který byl vybrán jako nejvhodnější pro náš projekt z toho důvodu, že se jedná o profesionální placený nástroj, který je ovšem pro malé projekty použitelný zdarma.

#### **Subversion (SVN)**

Základní nástroj pro správu verzí zdrojového kódu vyvíjený firmou CollabNet, Inc. poskytovaný zdarma. Systém [17] se skládá z klientské a serverové části, přičemž klientská část poskytuje nástroje pro práci s verzemi v pracovní složce a komunikaci se serverovou částí. Serverová část má na starost centrální úložiště dat (repository). Do centrálního úložiště lze přistupovat mnoha způsoby, přístupem přes příkazovou řádku počínaje a integrovaným GUI klientem konče.

Po nahrání zdrojového souboru do repository je tento soubor nasdílen všem zúčastněným uživatelům. Ti poté mohou v práci pokračovat na své vlastní pracovní kopii nasdíleného souboru. Po ukončení úprav je proveden tzv. commit a změněný soubor je uložen do repositáře. V případě

paralelní práce dvou programátorů je zahlášen konflikt, který si mezi sebou commitující programátoři vyřeší ručně, například komunikací přes Instant Messenger (IM).

K úspoře dat při komunikaci klienta se serverem slouží tzv. change-sets. Při změně se totiž nezasílají cele zdrojové soubory ale pouze informace o změně od poslední verze.

## **Bugzilla**

Jedná se o systém [18] pro sledování chyb, požadavků na rozšíření a jiných úkolů napsaných v Pearlu. Je produktem společnosti Mozilla.org, která ho používá při vývoji svých produktů. Dnes Bugzillu používají stovky komerčních i nekomerčních projektů po celém světě. Systém je dostupný zdarma i pro komerční využití. Systém potřebuje ke své činnosti aplikační server a databázový systém. Chybu může vložit kdokoliv a každá chyba je přiřazena konkrétnímu vývojáři. U každé chyby je možné evidovat řadu parametrů, jako například popis, stav, závažnost a podobně. Bugzilla je primárně určena pro provozování na Linuxu, ale lze ji nainstalovat i na Windows.

Mezi hlavní výhody systému Bugzilla patří propracované fulltextové vyhledávání, snadno přizpůsobitelný vzhled pomocí šablon, lokalizované uživatelské rozhraní a sledování změn prostřednictvím elektronické pošty.

## **Flyspray**

Alternativní produkt [19] k výše uvedené Bugzille napsaný v jazyce PHP s moderním uživatelským rozhraním řešeným částečně v AJAXu. Dalším rozdílem je pak možnost sledování změn přes Jabber.

## **Stránky Wiki**

Jedna ze základních možností výměny informací v rámci řešitelského týmu. Wiki umožňuje [20] vytvářet dokumenty kolektivně pomocí jednoduchého značkovacího jazyka za použití webového prohlížeče. Jednou z definičních charakteristik wiki technologie je snadné vytváření a aktualizace stránek. Obecně není před přijetím stránky prováděna žádná kontrola a většina systémů wiki je otevřena veřejnosti, nebo přinejmenším komukoli, kdo má přístup na wiki server. Tím může být právě omezená skupina vývojářů projektu.



## **Team Foundation Server**

Produkt společnosti Microsoft určený pro vývojové prostředí Visual Studio. (Plný název je Microsoft Visual Studio Team Foundation Server 2010.) Cílem aplikace [21] je usnadnění řízení životního cyklu softwarového díla tak, že umožňuje provádět veškeré související činnosti, jako je modelování, kódování a testování v rámci jedné aplikace. Stejně tak se Team Foundation Server stará o správu centralizovaného úložiště dat a sdílení zdrojového kódu mezi jednotlivými programátory. Verze softwaru 2010 je optimalizována pro iterativní postupy práce včetně populární metodiky SCRUM.

### **3.4.1. IBM Rational Team Concert a platforma Jazz**

Rational Team Concert (RTC) společnosti IBM je aplikace [22] integrující systémy pro oznamování pracovních položek a chyb, pro správu verzí, pro prototypování, iterační plánování a pro konfiguraci pracovních procesů tak, aby zajistila vývojářům, architektům a projektovým manažerům co nejefektivnější spolupráci. Je distribuována ve třech verzích, přičemž verze Express-C je do počtu tří licencí poskytována zdarma, jak již bylo uvedeno dříve.

Celá aplikace je součástí platformy Jazz, která je postavena na open-source technologii Eclipse. Tím je jednak umožněno užívání dalších produktů určených pro tuto platformu (Rational Quality Manager, Rational Requirements Composer), ale také používání stovek užitečných modulů Eclipse. Cílem platformy Jazz je zvýšení efektivity dodávky softwaru pro vývojářské týmy právě cestou usnadnění týmové práce.



## 4. SCÉNÁŘ WEBOVÉ HRY PRO KIV

V předcházejících kapitolách již bylo řečeno, že pro prezentační účely Katedry informatiky a výpočetní techniky byla zvolena forma online webové hry pro více hráčů. Při stanovování tématu, herních možností a omezení a svým způsobem i vize celého projektu byl využit dokument gamedesign, který je při tvorbě her základním dokumentem projektu. V dokumentu jsou popsány všechny důležité informace ohledně hratelnosti, nejsou řešeny detaily implementační části projektu. V zásadě se jedná o odborný článek, ze kterého vychází analýza a návrh hry.

### 4.1 Space Traffic – Gamedesign

#### Úvod

Tento dokument si klade za cíl podat kompletní informace o online webové hře Space Traffic z hlediska hratelnosti a herních možností. Nenalézají se zde žádné podrobnější informace o technickém zpracování hry, návrhové ani analytické detaily.

Nejprve bude nastíněna základní myšlenka hry včetně gameflow. Poté budou představeny další důležité prvky hry.

#### Základní rysy a princip hry

Popisovaná hra patří do kategorie MMORTS (viz kap. 2.1.1) s tématem obchodování ve vesmíru. Je určena pro cílovou skupinu mládeže od 15ti do 19ti let. Ve hře budou také existovat principy umožňující hráčům znalým programování vytvoření vlastních technologií, které mohou využívat k co nejvyšší prosperitě své společnosti. Hráč manipuluje se sadou objektů (lodě, budovy, technologie) tak, aby si zajistil co největší výdělek.

Herním plánem jsou dynamické mapy slunečních soustav, z nichž některé jsou propojeny červími děrami. V tomto smyšleném vesmíru bude operovat hráč ovládaný počítačem (NPC - Non Player Character) s vlastní společností pro doručování zásilek, což zaručí, že i první hráč bude mít protihráče.

Hra bude v zásadě umožňovat hráčům vydělávat různými způsoby. Prvním a hlavním z nich je doručování obchodních zásilek mezi jednotlivými planetami. Po vytvoření hráčského

úctu je náhodně vybrána planeta, na kterou je umístěna hlavní základna hráče. Hráč získá do vlastnictví standardní vesmírnou loď a získá přístup k obchodním nabídkám v dané sluneční soustavě.

Nové obchodní zásilky vytvářejí herní systém automaticky. Nebude ovšem možné do každé lodi naložit každou zásilku. Ve hře existuje více druhů lodí. Generování obchodních zásilek na planetách se také bude řídit předem definovanými pravidly (jaké typy, jaké cílové destinace apod.), což zpestří a znesnadní hráčům výběr vhodné zásilky pro jejich loď.

Vesmírná plavidla budou také čelit náletům pirátských stíhačů, kteří mohou sestřelit některé ze strážných lodí, poškodit loď nákladní nebo celou flotilu úplně zničit. Pirátská skupina je částečným NPC ve hře. Její vznik ovlivní sami hráči tím, že některou ze svých lodí pirátům věnují. Útoky pirátů na hráčovu flotilu jsou ovšem tím méně časté, čím jsou dary pirátům štědrější. Vztahy k pirátům jsou ovlivněny dary, vzájemnými souboji, ale také svévolí pirátů, které mohou i štědré dárcy vydírat a nutit k dalšímu darování lodí.

Lodě lze pochopitelně libovolně dokupovat v závislosti na solventnosti hráče. Pokud chce hráč doručit zásilku do jiné sluneční soustavy, než v jaké zásilku nakládá, cestu vesmírem si musí sám naplánovat výběrem soustav přes které se má letět. Flotila se poté pohybuje přímočaře z jedné soustavy do druhé a v cílové soustavě letí k cílové lodi. Navigování lodí ve hře je možné dvojím způsobem. Každá loď je vybavena základní navigací, která dokáže sledovat aktuální polohu cílové planety a dle ní upravit kurz lodi. Hráčům bude také umožněno naprogramovat vlastní navigaci, kterou mohou po schválení systémem jejich loď využívat.

Nakládání zásilek do lodí provádí hráč „ručně“ výběrem zásilky v seznamu dostupných zásilek. Tuto činnost lze také automatizovat naprogramováním vlastního robota, který může být po schválení systémem hráčem používán.

Obě výše uvedené technologie (navigace a roboti) mohou být po schválení používány výhradně hráčem, který je vymyslel, nebo mohou být nabídnuty ke koupi ostatním hráčům. I toto je způsob, jak zabezpečit vlastní společnosti co největší obrat.

Létání vesmírných lodí s sebou samozřejmě ponese i náklady. Náklady budou určeny dle počáteční a cílové planety letu a počtu změn kurzu během letu, eventuálně dle počtu lodí, které jsou ve flotile. Dalšími náklady jsou poplatky za použití přistávacího doku. Tím se dostáváme k dalším možnostem hry, v zásadě odlišným od výše uvedených.

Hráči mají ve hře také možnost vybudovat vlastní dokovací stanice, ve kterých neplatí přistávací poplatky. Vybudováním sítě doků mohou naopak vydělávat, a to tak, že nabídnou jiným hráčům nižší poplatky za přistání než NPC nebo ostatní hráči. Tato možnost s sebou nese nutnost vybrat přistávací dok před zahájením letu. Planety ovšem nepojmou nekonečný počet přistávacích stanic. Počet možných doků na konkrétní planetě je předem definován a nelze jej nikdy překročit. Se stavbou doků tedy není radno otálet.

Vzhledem k tomu, že se jedná o multiplayerovou hru, je hráčům umožněno sdružovat se do aliancí. Zapojení do aliance s sebou přináší následující změny. V dokovací stanici lze zvlášť stanovit cenu pronájmu pro členy aliance a ostatní hráče. Vztah k pirátům mají všichni členové v alianci shodný, členové aliance spolu mohou komunikovat mimo standardní zasílání zpráv také na aliančním fóru. Nutností pro šíření hry je stanovení systému pozvánek pro nové hráče a vyplácení bonusů za přivedeného aktivního hráče.

Hra je hodnocena formou několika hodnotících žebříčků a je také pravidelně vyhodnocována. Hodnotíme zvlášť hráče s největším obratem, s nejpočetnější flotilou, s nejrozsáhlejší infrastrukturou staveb, hráče s největším nárůstem staveb a lodí flotily od minulého hodnocení a zvláště můžeme také hodnotit hráče s nejvyšším obratem v poměru k času, který hráč strávil hraním hry (nejlepší manažer hraje hru minimálně a přesto nejvíce prosperuje).

## **Gameflow**

Při vstupu do hry je nejprve nutná registrace. Po vyplnění všech potřebných údajů je hráči zaslána mailová zpráva obsahující odkaz k aktivaci účtu. Po provedení aktivace je možné se přihlásit do hry.

Po přihlášení je hráči zobrazena mapa systému Tíras, jedné z mnoha slunečních soustav, ve které je označena jeho domovská planeta. Na této planetě má hráč základu a svoji první loď. Hráč se rozhodne ihned začít vydělávat ve velkém, a nalezne tak nejcennější zásilku v celé soustavě, která ovšem není k vyzvednutí na jeho domovské planetě. K přepravě pak určí svoji jedinou loď. Jeho vesmírné plavidlo se v tom okamžiku vznese a začne po mapě cestovat směrem k planetě, kde se nachází zásilka k vyzvednutí. Po uplynutí určité doby loď dorazí na planetu a automaticky vyzvedne zásilku. Poté se opět zcela automaticky uvede do letu a cestuje napříč vesmírem k cílové planetě. Po přiletu je zásilka automaticky vyložena a hráči přibudou na herní

účet stanovené peníze. Z logu akcí je ovšem také patrné, že hráč zaplatil jisté částky za přistání na planetách, kde nevládnul žádné budovy. Proto se rozhodne další zásilku vyzvednout na planetě, kde jeho loď právě dokuje a vybere takovou zásilku, jejímž cílem je jeho domovská planeta. Doručování probíhá stejně jako v minulém případě, jen od zisku není odečten poplatek za přistávání. Uvedeným způsobem hráč pokračuje dokud nemá dostatek finančních prostředků ke koupi nákladní lodi.

Jakmile hráč získá dostatek finančních prostředků rozhodne se ke koupi nákladní lodi. Napřed je ovšem nutné zakoupit licenci, která k provozu takové lodi opravňuje. Hráč souhlasí s uvedenou cenou a přechází do menu *Koupit loď*. Vybírá nákladní loď a současně zakupuje dvě bojové lodě k ochraně lodi nákladní. V menu *Management lodí* poté tyto lodě k sobě vzájemně přiřadí. Protože s nákladní lodí lze létat i do jiných slunečních soustav a převážet více zásilek najednou, nalezne hráč na svojí domovské planetě (kde byla nákladní loď koupena) všechny zásilky se stejnou cílovou planetou v jiné vzdálené sluneční soustavě resp. v systému Sineár. Po naložení lodi pěti zásilkami a vyčerpání kapacity lodi hráč volí možnost dokončit doručení. V tu chvíli je mu zobrazena mapa galaxie, aby vyhledal cestu, po které lze do cíle doletět. Nejkratší cesta ze systému Tíras do systému Sineár vede přes další dva systémy. Tuto cestu hráč v mapě nalezne a po potvrzení se jeho nákladní loď včetně dvou svých hlídačů začne přímočaře pohybovat k nejbližší červí díře a tak stále dokola, dokud nedorazí do cílového systému, kde zamíří k cílové planetě. Hráči je vyplacena velká částka za velkého množství zboží na velkou vzdálenost. Současně je mu také odečtena vypočtená částka jako náklady na jeho let.

Během takto dlouhé cesty však došlo k tomu, že byla jeho flotila napadena pirátskými loděmi. O této skutečnosti byla hráči zaslána zpráva na herní účet (ne na e-mail). Hráč v souboji nepřišel o žádnou loď a piráty porazil. Jeho vztahy s piráty ale díky tomu klesly na nepřátelskou úroveň. Aby hráč zajistil relativně klidné průběhy letů svých lodí, rozhodne se zakoupit jednu bojovou loď a pirátům ji věnovat. Tím se jeho vztah s piráty znovu vrátil na neutrální úroveň.

Vzhledem k tomu, že se jedná o hru pro více hráčů, rozhodne se hráč pro stavbu ambasády a založení aliance. Všem hráčům se stejnou domovskou planetou rozešle pozvánky a čeká na jejich odpovědi. Vstupem dalších hráčů do aliance se všem otevírají další herní možnosti. Hráči mají jednotný vztah k pirátům, mohou mezi sebou snadněji komunikovat, podporovat se po finanční a materiálové stránce, nebo podnikat společné lety při kterých poté získávají převahu oproti případným pirátským útokům.

## **Lodě a budovy**

Ve hře budou existovat standardní lodě, lodě na nebezpečné náklady a nákladní lodě. Také bude existovat více druhů zásilek. Do standardní lodě lze naložit pouze jednu zásilku odpovídajícího typu, stejně tak jako do lodě na nebezpečný náklad. Do nákladní lodě lze nakládat zásilky se stejným místem doručení do vyčerpání kapacity lodi a jejich kapacitu lze zvětšovat.

Posledním typem lodi, která se objeví ve hře je strážní, bojová loď (guard). Její účel je následující: Lodě standardní a lodě na nebezpečný náklad mohou létat pouze v jedné sluneční soustavě. Ne tak lodě nákladní. Ty mohou cestovat i do jiných slunečních soustav a je nutné přiřadit k nim ozbrojený doprovod v podobě strážných lodí. Počet potřebných strážců je závislý na levelu nákladní lodi. Bojové lodi mají také za úkol střežit nákladní loď, ke které jsou přiřazeny při útoku pirátů.

Ve hře se dále objevují budovy – doky a ambasáda. Vystavěním doku hráč zajistí sobě a svým spojencům bezplatné přistávání na planetě, kde byl dok vybudován. Stavba ambasády pak umožňuje získat spojence a založit alianci.

## **Herní prostředí**

Celá hra se bude odehrávat ve smyšleném sektoru galaxie s mnoha slunečními soustavami, neboli systémy (například 25). Každá soustava má vlastní slunce a tři nebo čtyři planety, které obíhají kolem slunce. V každé soustavě je také předem určený počet červích děr, kterými lze proletět do jiných systémů. Polohy těchto děr jsou všem hráčům známé od začátku hry. Zobrazovaná herní mapa bude čtvercová aby bylo možné zajistit, že obíhající planety kolem slunce budou na mapě viditelné ve kterékoli fázi oběhu.

Po stranách mapy budou zobrazeny dva panely. Levý panel bude sloužit jako menu, v pravém panelu budou zobrazovány relevantní informace o konkrétních objektech, které hráč aktivuje. Nad mapou budou zobrazeny nejdůležitější informace o hráčově společnosti, jako jsou počty lodí, stav financí, náležitost k alianci apod., dále zde bude umístěn odkaz na nastavení a profil a změnu jazyka aplikace. Pod herní mapu bude zobrazováno několik posledních hráčových akcí – log. Po přepnutí zde bude možné zobrazit živý pokec – live chat.

Aplikace bude ovládána výhradně přes webový prohlížeč s použitím klávesnice a počítačové myši.

## **Grafika**

Grafika bude omezena na jednoduchý styl prezentovaný pomocí jazyka HTML a kaskádových stylů (CSS). Základní mapa vesmíru bude černá s drobnou dekorací připomínající vesmír. Na ní budou zobrazována jednotlivá tělesa ve 2D. Veškeré grafické prvky dodá určený grafik – ilustrátor dle vlastního vkusu nebo předložených schématických návrhů.

Grafika hry bude přizpůsobena zobrazení, resp. rozlišení více typů monitorů. Pro každou výšku monitoru se vždy vybere největší možné zobrazení tak, aby se nejméně spodní hrana mapy vtěsnila na monitor bez nutnosti skrolování. Velikost zobrazovaného herního rozhraní má hráč možnost změnit ve svém nastavení.

## **Zvuky a hudba**

Aplikace v zásadě nepočítá s žádným výrazným ozvučením. Přehrávání hudby na pozadí ve webovém prohlížeči působí rušivě. Pro oživení hry je však možné jednoduchým způsobem ozvučit následující akce:

- Aktivace položky menu
- Změna stavu finančního konta
- Změna vztahu s piráty
- Nová příchozí zpráva
- Výběr objektu na mapě (dle typu objektu – loď, slunce, planeta ...)

Ozvučení akcí je závislé na přání hráče. Ten může ve svém nastavení vybrat libovolné zvuky, které budou přehrávány, nebo všechny zvuky povolit/zakázat.

## **NPC**

Hra obsahuj v zásadě dvě formy NPC (non-player character tj. nehráčská postava). První formou je doručovatelská společnost, která ve hře operuje od jejího spuštění a není závislá na počtu hráčů. Tato společnost dle předem definovaných algoritmů doručuje zásilky a nakupuje nové lodi a tvoří tak přirozenou konkurenci i minimálnímu počtu hráčů.



Druhým NPC ve hře jsou piráti, mající za úkol napadat nepřátelské flotily, znesnadňovat doručení zásilek a ničit hráčům lodě. Chování pirátů je ovlivněno vzájemným vztahem pirátů a hráče, resp. pirátů a aliance. S klesající vzájemnou náklonností roste šance napadení hráčových lodí. Vztahy jsou upravovány směrem k lepšímu vždy po darování lodě pirátům, směrem k horšímu pak vždy, když hráč porazí pirátskou letku v boji, nebo se pro toto zhoršení vztahů piráti svévolně rozhodnou.

## **Fyzika**

Rychlosti pohonu vesmírných těles i všech ostatních objektů se zakládají na částečně reálných hodnotách. Poměry rychlostí jsou zachovány v reálném stavu, běh času je však oproti realitě několikanásobně zrychlen. Tělesa ve hře jsou nekolizní a nevykazují žádné deformace a mohou procházet jedno druhým. V reálných měřítkách ke kolizi dojde velmi zřídka a proto se touto možností zobrazení objektů nezabývá.

Planety a slunce však vykazují vlastní gravitační pole, které znesnadňuje létání v blízkosti těchto těles. Při vletu lodě do gravitačního pole některého objektu má gravitační síla vliv na kurz letící lodě případně jiného objektu.

## **Návrh řešitelského týmu a odhad doby realizace**

Pro realizaci výše uvedeného herního konceptu bude vhodné sestavit následující tým řešitelů:

Analýza a návrh vlastností systému	-	2 lidé
Programování vlastní aplikace	-	4 lidé
Tvorba databáze	-	2 lidé
Grafické práce	-	1 osoba

V případě účasti vysokoškolských studentů, u kterých nelze garantovat každodenní osmi hodinovou pracovní dobu, se doba na realizaci projektu odhaduje na jeden akademický rok.



## 5. VEDENÍ TÝMU PŘI REALIZACI PROJEKTU

Před zahájením tvorby hry bylo nutné určit používanou metodiku, předpokládaný průběh iterací a sestavit řešitelský tým. Proto se v této kapitole nejprve seznámíme s metodikou Rational Unified Process (RUP) a uvedeme způsob jejího použití v projektu společně s metodikou FDD. Dále pak bude představen řešitelský tým a předpokládaný průběh jednotlivých iterací.

### 5.1 Použité metodiky a nástroje

#### Rational Unified Process (RUP)

Metodika RUP [23] se zrodila jako vyspělý a otevřený standard tvorby softwarového vybavení vyvinutý autory jazyka UML (viz. kap. 6.1). Usiluje o přírůstkovou tvorbu robustní architektury navrhovaného systému a obsahuje pět základních pracovních postupů. Těmito postupy jsou:

Požadavky – Analýza – Návrh – Implementace - Testování

Přírůstková tvorba vytvářeného systému může v každém iteračním kroku obsahovat až všech pět pracovních postupů v závislosti na aktuálních požadavcích. Množství práce potřebné vykonat v jednotlivých iteracích je ale poplatné fázi vývoje softwarového díla.

Metodika UP také definuje 4 základní fáze vývoje softwarového díla, a to:

Zahájení – Rozpracování – Konstrukce – Zavedení

Každá fáze vývoje definuje své vlastní cíle, kterými jsou ve fázi Zahájení začátek práce na projektu, ve fázi Rozpracování je cílem vytvoření první spustitelné verze, ve fázi Konstrukce převedení první verze na kompletní funkční systém a ve fázi Zavedení nasazení vytvořeného systému na provoz u uživatele - zákazníka .

### **Konkrétní použitá metodika, průběh iterací**

Pro účely tohoto projektu byla zvolena metodika kombinující univerzálnost metodiky RUP a klíčovou vlastnost metodiky FDD. V každé iteraci byly nejprve stanoveny její cíle a to zejména prostřednictvím seznamu užitečných vlastností (features). Úspěšná implementace těchto vlastností se stala základním hodnotícím kritériem iterace. Poté bylo v zásadě postupováno dle metodiky RUP. V každé iteraci byly provedeny všechny pracovní postupy, ve kterých bylo odvedeno takové množství práce, které odpovídalo fázi vývoje softwaru. Výraznou změnou prošel pracovní postup návrh, který byl silně minimalizován. Návrhová rozhodnutí byla přenechána programátorovi a vzhledem ke složení týmu nebylo nutné přesným návrhem koordinovat práci více lidí, jak bude uvedeno v následující kapitole. Stejně tak nebyly ve všech postupech použity veškeré doporučené UML modely, užívané v metodice RUP. Projekt produkoval pouze takové množství dokumentace, které postačovalo pro bezproblémovou programátorskou činnost.

### **Práce s nástrojem RTC**

Jako podpůrný nástroj pro vedení týmu a organizaci prací byl zvolen software Rational Team Concert, krátce představený v kapitole 3.4.1. Serverová část byla nainstalována na server <http://jazz.kiv.zcu.cz> a administrační práce byly provedeny Bc. Janem Boháčem, který se administrací toto systému zabýval v rámci své diplomové práce. Členové řešitelského týmu byly poté zodpovědny za instalaci klientské části včetně rozšíření, které umožňovalo vývoj aplikace v jazyce PHP.

Vzhledem k možnosti lokalizace nástroje do českého prostředí, je práce po překonání prvotního překvapení ze softwaru, který se nám snaží nabídnout vše, poměrně příjemná. Pomineme-li instalaci klientské a serverové části, sestává počátek práce na projektu vytvořením oblasti projektu s nutnou definicí metodiky, která bude použita, a založením repositáře, neboli centrálního úložiště dat (podobnost s popsáním systémem Subversion). Dále je nutné stanovit tým řešitelů, přidělit jim role v projektu a přizvat je do vytvořené projektové oblasti. Tuto pozvánku lze zaslat jednoduchým způsobem formou e-mailu a dotýčný pozvaný vývojář celou záležitost uzavře pouhým překopírováním obsahu pozvánky do svého klienta RTC.

Máme-li definovanou oblast projektu a vytvořený tým řešitelů, můžeme dle přidělených práv stanovovat plány jednotlivých iterací, vytvářet a přiřazovat nové pracovní položky,

psát komentáře a diskuse k jednotlivým řešeným problémům, vytvářet a zálohovat subverze výsledného programu a v neposlední řadě řešit přidělené úkoly (pracovní položky).

Po vyřešení pracovní položky je nutné nasdílet vytvořené zdrojové soubory nebo změny těchto souborů do centrálního úložiště dat. Tuto operaci lze provést pouze, pokud je odesílanému setu změn přidružen komentář a především alespoň jedna pracovní položka (závislé na konkrétním konfiguračním nastavení). Tato vlastnost umožňuje týmu vývojářů plně se soustředit na řešené problémy, místo vyváření nepotřebných částí systému a také lze díky tomu vždy dohledat, za kterou změnu zdrojových souborů je kdo zodpovědný a jaké byly důvody této změny. Samozřejmostí je poté provázání pracovních položek s některou z iterací v časovém harmonogramu projektu. Tímto celkovým vzájemným provázáním můžeme vytvářet přehledné statistiky o projektu a v každé fázi projektu říci, které funkcionality a vlastnosti vytvářeného softwarového produktu byly kdy a v jaké iteraci řešeny.

Mezi další možnosti nástroje RTC patří velice pokročilá filtrace pracovních položek. Každý uživatel má možnost kromě standardně nastavených filtrů definovat libovolný počet vlastních filtrů, které mu umožní rychlou orientaci v přidělených i vyřešených pracovních položkách (viz obrázek). Dále je možné nastavit informování o změnách v projektu e-mailovými zprávami, také s velkou variabilitou nastavených parametrů. K dispozici je i zabudovaný jabber komunikátor, který ještě více usnadňuje komunikaci uvnitř.

Systém nám také umožňuje plánovat průběhy iterací v závislosti na prioritách a časové náročnosti úkolů s ohledem na časové možnosti členů řešitelského týmu. Tím se nástroj stává vhodný i pro nekomerční nebo studentské projekty, kde nemůžeme předpokládat plné pracovní nasazení vývojářů.

I	Stav	P	S	Souhrn	Vlastník	Vytvořen(a) kým
267	Vyřešeno	🚫	🔴	<b>Oprava registrace na třetí iteraci</b>	Petr Kellhofer	Zbyněk Neudert
266	Vyřešeno	🟡	🟡	Předat výsledek práce 3 betatesterům	Zbyněk Neudert	Zbyněk Neudert
265	Vyřešeno	🟡	🟡	Odeslat žádosti o review	Zbyněk Neudert	Zbyněk Neudert
264	Vyřešeno	🟡	🟡	Kontrola aplikace, vyřízení reklamací	Zbyněk Neudert	Zbyněk Neudert
258	Vyřešeno	🟡	🟡	<b>P: Zobrazení infrastruktury společnosti</b>	Petr Kellhofer	Zbyněk Neudert
257	Vyřešeno	🟡	🟡	<b>P: Smazání účtu</b>	Petr Kellhofer	Zbyněk Neudert
256	Vyřešeno	🟡	🟡	<b>P: Změnit heslo</b>	Petr Kellhofer	Zbyněk Neudert
253	Vyřešeno	🟡	🟡	P: Platby za přistání	Petr Kellhofer	Zbyněk Neudert
244	Vyřešeno	🚫	🔴	P: Postavit budovu - dokovací stanice	Petr Kellhofer	Zbyněk Neudert
220	Vyřešeno	🟡	🟡	Vytvořit zadání pro DB2	Zbyněk Neudert	Zbyněk Neudert
215	Vyřešeno	🟡	🟡	P: Koupit loď	Petr Kellhofer	Zbyněk Neudert
214	Vyřešeno	🟡	🟡	P: Hodnotící tabulky	Petr Kellhofer	Zbyněk Neudert
212	Vyřešeno	🟡	🟡	<b>P: Čtení zpráv</b>	Petr Kellhofer	Zbyněk Neudert
211	Vyřešeno	🟡	🟡	<b>P: Zaslání zpráv</b>	Petr Kellhofer	Zbyněk Neudert
153	Vyřešeno	🟡	🟡	Napsat reflexi na desig review 091228	Petr Kellhofer	Zbyněk Neudert
152	Vyřešeno	🟡	🟡	P: Aktualizace zásilek na plnětách	Petr Kellhofer	Zbyněk Neudert
151	Vyřešeno	🟡	🟡	Definovat pracovní položky pro programátora	Zbyněk Neudert	Zbyněk Neudert
150	Vyřešeno	🟡	🟡	P: Úprava po registraci	Petr Kellhofer	Zbyněk Neudert
149	Vyřešeno	🟡	🟡	P: Reset hesla při zapomenutí	Petr Kellhofer	Zbyněk Neudert

Obrázek 5.1 – Filtrace pracovních položek v prostředí nástroje IBM RTC

## **5.2 Sestavení a představení řešitelského týmu**

Z dokumentu gamedesign uvedeného v kapitole 4.1 vyplývá, že pro realizaci uvedeného záměru by bylo vhodné sestavit řešitelský tým o počtu osmi lidí. Zajištění dostatečného počtu řešitelů bylo věnováno velké úsilí. Byly použity následující metody:

### **Osobní oslovení potencionálních členů, zprostředkované oslovení**

Osobní oslovení probíhalo formou krátké prezentace na přednáškách vybraných předmětů. Tato forma zaznamenala mizivý úspěch. Zájem o práci na projektu projevil pouze jeden student předmětu Databáze 1, jehož úkolem se mělo stát vytvoření datového modelu pro výslednou aplikaci. Je nutné podotknout, že zmíněný student odvedl práci na hranici akceptovatelnosti a na projektu po velmi krátké době přestal pracovat.

Díky zprostředkovanému oslovení byl dalším členem projektu student předmětu Webové aplikace. Úkolem byly implementační práce ve vybrané technologii.

### **Oslovení studentů formou informativních letáků**

Vzhledem k minimálnímu úspěchu osobního pozvání, byly v prostorách katedry informatiky a výpočetní techniky vyvěšeny letáky obsahující informaci o chystaném projektu a nabídku spolupráce při implementaci webové hry. Na tuto formu upoutávky nikdo nereagoval.

### **Nabídnutí participace na projektu v rámci semestrální práce přes aplikaci Courseware**

S ohledem na stále nedostatečný počet řešitelů a absenci grafika byly některé části práce nabídnuty studentům ke zpracování přímo prostřednictvím aplikace courseware, sloužící ke zveřejňování zásadních informací o předmětech vyučujícími. Bylo nutné formalizovat zadání pro jednotlivé předměty a zajistit dostatečnou informovanost jednotlivých vyučujících. Tyto práce zabrali nemalé množství času a ani přes námahu tomu věnovanou, nereagoval na tyto nabídky žádný student.

### **Vypsání stipendia za provedení grafických návrhů pro projektu katedrou KIV**

Posledním způsobem zajištění alespoň minimální velikosti řešitelského týmu bylo v závěrečných fázích projektu, po vyčerpání všech jiných variant, vytvoření poptávky na obsazení pozice grafika-ilustrátora s výhledem na finanční odměnu. Díky tomu se podařilo

personálně obsadit všechny předpokládané práce, které jsou uvedeny na závěr dokumentu gamedesign.

### Řešitelský tým

Práce si nyní dovoluje předložit k porovnání předpokládaný a skutečný stav řešitelů projektu:

		Předpokládaný stav	Skutečný stav
Analýza a návrh vlastností systému	-	2 lidé	1 osoba
Programování vlastní aplikace	-	4 lidé	1 osoba
Tvorba databáze	-	2 lidé	0,2 <sup>2)</sup> osob
Grafické práce	-	1 osoba	0,2 <sup>2)</sup> osob

### Shrnutí

Průběh prací na projektu byl dále ovlivněn tím, že se jedná o univerzitní prostředí. Toto prostředí nám totiž neumožňuje efektivně řešit tyto základní problémy (mimo problému sestavení řešitelského týmu):

- Zajištění včasného odevzdání zadané práce
- Zajištění kvalitního zpracování zadané práce

Ve firemním prostředí je mocnou motivací systém finančních benefitů nebo srážek z platu. V univerzitním prostředí neexistuje žádná podobně silná alternativa.

Na základě výše uvedených skutečností je zřejmé, že nebylo možné realizovat celou hru tak, jak je nastíněna v dokumentu gamedesign. Veškeré práce a průběh iterací byly podřízeny vytvoření nasaditelné a hratelné betaverze.

---

<sup>2)</sup> Tímto způsobem se práce snaží alespoň přibližně vystihnout podíl doby, po kterou se jednotliví členové podíleli na projektu vzhledem k jeho celkové době trvání. Tedy 1 osoba = osoba účastnící se projektu od jeho zahájení do jeho ukončení.

### 5.3 Plánování, tvorba a rozdělování pracovních úkolů

Vzhledem k pevně danému termínu odevzdání práce a fixním zdrojům byl projekt rozplánován do pěti iterací po dvou nebo třech týdnech. Volitelným prvkem se tak stala výsledná funkčnost, což přesně odpovídá agilním principům. Na konci téměř každé iterace bylo žádáno o sepsání posudku na aktuální prototyp práce. Všechny tyto posudky (design reviews) jsou uvedeny v příloze této práce. Dále byly uskutečněny konzultace s vedoucím práce a konzultantem projektu. Na základě těchto aktivit byly stanoveny nové plány následujících iterací, nebo byly předchozí plány změněny dle aktuálních potřeb projektu.

Po celou dobu trvání projektu, na jeho realizaci pracovali pouze dva lidé. Dělbba prací proto byla snadná. V počátečních fázích každé iterace byly prováděny především analytické práce a po jejich vyřešení byly stanoveny programátorské pracovní položky (viz. obrázek 5.1) řešené druhým členem týmu.

Každá pracovní položka našeho projektu si s sebou vždy nesla odkaz na požadavek, který řeší, případně na diagram, který upřesňuje způsob řešení. Modelování probíhalo v nástroji Enterprise Architect, proto byly tyto odkazy do pracovních položek zadávány ručně. Dále byly stanovovány testy, kterými se po splnění položky testovalo, zda byla položka splněna správně a úplně. Během závěrečných prací byla také využívána diskuse přidružená ke každé položce, ve které si členové týmu vyměňovali informace přímo ovlivňující plnění položky.

The screenshot shows the IBM RTC project plan interface. At the top, it displays the project name 'Plán' and team information: 'Oblast týmu: Tým Webgame KIV | Iterace: Iterace V (20.4.10 - 11.5.10) | Uzavřeno 12 | Otevřeno 12'. The progress bar shows 'Průběh: 54,5/127,5 | -3,25 h' and 'Vytížení: 63 / 3 | -60 h' with an 'Odhad: 100%' indicator.

Below the team information, the tasks are listed for two team members:

**Petr Kellnhofer**  
Uzavřené položky: 8 | Otevřené položky: 10  
Vytížení: 63 / 3 | -60 h | Odhad: 100%

Task	Priority	Complexity	Duration	Start	End	Estimate
P: Létání lodí do jiných slunečních soustav	Normální	Vysoká	1 hodina	..	..	482
P: Bonus za přivedené nového hráče	Normální	Střední	3 hodiny	..	..	517
Nasazení nové grafiky	Normální	Střední	1 týden	..	..	479
P: Možnost přijmou zásilku za letu	Normální	Střední	2 hodiny	..	..	471
P: Live chat	Normální	Střední	2 hodiny	..	..	468
Společní létání členů aliance	Normální	Nízká	1 den	..	..	478
P: Sežazení lodí při výběru lodě pro doručení	Normální	Nízká	2 hodiny	..	..	472
Doplnění možnosti "nalož a dokonč" u nákladní lodí	Normální	Nepřřazeno	2 hodiny	..	..	541
Změna formy platby za přistání	Normální	Nepřřazeno	1 hodina	..	..	540
P: Pirátské útoky na ambasády	Normální	Nepřřazeno	2 hodiny	..	..	477

**Zbyněk Neudert**  
Uzavřené položky: 4 | Otevřené položky: 2  
Vytížení: 10 / 5 | -5 h | Odhad: 100%

Task	Priority	Complexity	Duration	Start	End	Estimate
Parsování grafického návrhu	Normální	Vysoká	1 den	..	..	488
Kontrola vyřešených položek	Normální	Střední	2 hodiny	..	..	485

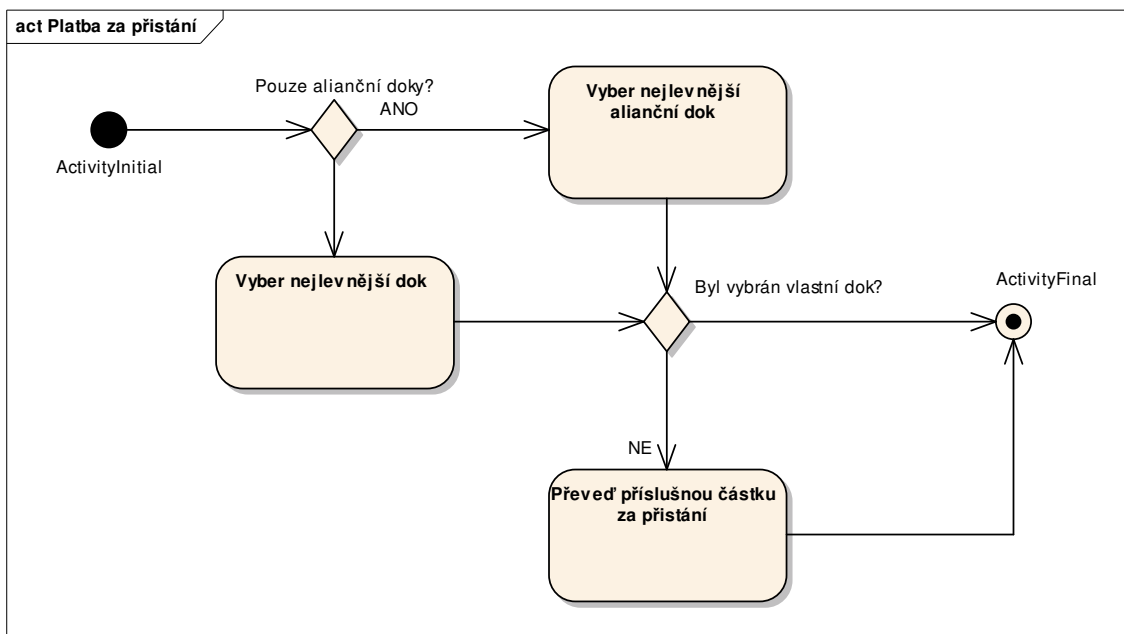
At the bottom, there is a section for 'Nepřřazeno' (Unassigned) with 0 closed and 0 open tasks.

Obrázek 5.2 – Některé pracovní položky páté iterace vývoje projektu – prostředí IBM RTC



### Příklad průchodu konkrétního požadavku jednou iterací

Následující část textu podává přímý popis vyřešení požadavku 8.1 – Přistávací poplatky. Tento požadavek byl řešen v průběhu třetí iterace. Byl zaveden do kalatogu požadavků v modelovacím nástroji Enterprise Architect. Protože se nejednalo o záležitost použití systému hráčem, ale o vnitřní pravidlo hry, nebyl k tomuto požadavku přidělen digram případů užití, ale diagram aktivit, který popisuje jaké akce musí systém vykonat v případě, že hráč přistane se svojí přepravní lodí na některé planetě. Jednotlivé stavy diagramu uvedeného na obrázku 5.2 jsou v programu EA dále specifikovány upřesňujícím popisem tak, aby se diagram stal konkrétním zadáním pro programátora a nebyl možný víceznačný výklad modelovaného pravidla (i když tomu při použití běžného jazyka nelze nikdy 100% zabránit).



Obrázek 5.3 – Diagram aktivit pro vyřešení požadavku na zavedení plateb za přistávání lodí

Takto zpracovaný požadavek byl poté zaveden do systému IBM RTC jako pracovní položka 253 naplánovaná pro iteraci 3:

## **253: P: Platby za přistání**

### **Popis**

Položka řeší povinnou část třetí iterace. Hráčům je zpoplatněno přistávání na planetách.

### **Řeší**

požadavek 8.1 - diagram aktivit Platby za přistání

### **Test**

Hráč platí definovanou částku na planetě kde nemá dok

Hráč neplatí nic na planetě kde má dok

Jakmile byla pracovní položka splněna, proběhly dva krátké uvedené testy k ověření správné funkčnosti vyřešeného požadavku ve hře.

### **Plány iterací**

Každý iterační plán vždy obsahoval informace o termínu zahájení prací a termínu jejich dokončení. Déle zde byl uveden stručný popis iteračních cílů, hrubé rozdělení úkolů, kritéria úspěšnosti iterace a seznam užitečných vlastností (features), které byly pro danou iteraci důležité a měly být vyřešeny. Práce nyní předkládá harmonogramy a cíle všech těchto iterací. Nejsou v nich ovšem zavedeny features, které iteraci ovlivňují ani klíčové úkoly a kritéria úspěšnosti. Kompletní iterační plány jsou přílohou práce. Přidělení klíčových vlastností aplikace k jednotlivým iteracím je také zřejmé z kapitoly 6.3.

## **Plán iterace I**

**Zahájení iterace:** 17.11.2009

**Ukončení iterace:** 20.12.2009

### **Cíle iterace:**

Hlavním cílem je představení vize celého projektu a analyzování základních částí hry. V rámci iterace vznikne nehratelná aplikace s dynamickou sluneční soustavou a systémovým doručováním vygenerovaných zásilek.

## **Plán iterace II**

**Zahájení iterace:** 04.01.2010

**Ukončení iterace:** 1.3.2010

### **Cíle iterace:**

Cílem iterace je vytvoření první verze hratelné aplikace, která umožňuje základní správu herních účtů a doručování zásilek hráčem. Jednotlivé objekty ve hře budou mít od této iterace více typů (lodě, zásilky apod.) Pro doručování zásilek je třeba analyzovat způsob letu a navigace vesmírných lodí, pravidla přepravování a doručování zásilek.

## **Plán iterace III**

**Zahájení iterace:** 2.3.2010

**Ukončení iterace:** 22.3.2010

### **Cíle iterace:**

Jako hlavní cíl je stanoveno zdokonalení aplikace vytvořené v předešlé iteraci. Hra obsahuje hodnotící systém a lze již svou doručovatelskou společnost rozšiřovat nákupem lodí a stavbou objektů. Přistávání na planetách je zpoplatněno. Hráči mohou měnit své profily, zasílat si mezi sebou zprávy.

## **Plán iterace IV**

**Zahájení iterace:** 23.3.2010

**Ukončení iterace:** 12.4.2010

### **Cíle iterace:**

Hlavním cílem zdokonalení aplikace vytvořené v předešlé iteraci. Při létání vesmírných lodí vznikají variabilní náklady rozdílné pro každou cestu. Hráči mají nově možnost nakupovat, vylepšovat a spravovat své nákladní lodě. Ve hře se objevují piráti, která napadají letící nákladní lodě jednotlivých hráčů.

## **Plán iterace V**

**Zahájení iterace:** 20.4.2010

**Ukončení iterace:** 11.5.2010

### **Cíle iterace:**

Hlavním cílem je vytvoření nasaditelné betaverze celého projektu. Hráčům je umožněno přepravovat zásilky mezi více slunečními systémy a mají také možnost sdružovat se do aliancí, což jim přináší výhody. Ve hře je alespoň 20 slunečních soustav. Do hry bude nasazen vytvořený grafický návrh.

## 6. TECHNICKÉ ZPRACOVÁNÍ PŘEDLOŽENÉHO KONCEPTU

Tato kapitola se věnuje všem technickým disciplínám, které byly v průběhu práce na projektu prováděny. V úvodu kapitoly seznamujeme s použitými nástroji, dále představujeme jednotlivé disciplíny tak, jak se promítaly do jednotlivých iterací a v závěru kapitoly podáváme přehledné shrnutí objemu prací v rámci celého projektu.

### 6.1 Použití nástrojů pro definici vlastností projektu

Modelování systému v jazyce Unified Modeling Language (UML) lze s výhodou použít při tvorbě softwarových projektů při použití jakékoli metodiky. Uvádíme proto nejprve shrnutí vlastností jazyka UML a následně definujeme způsob jeho použití při tvorbě online webové hry pro KIV. Krátce je také představen modelovací nástroj Enterprise Architekt (EA).

#### Unified Modeling Language (UML)

UML je univerzální jazyk [23] pro vizuální modelování systémů a lze jej využít nejen při modelování objektově orientovaných softwarových projektů. Tento jazyk byl navržen tak, aby spojil nejlepší existující postupy modelovacích technik a softwarového inženýrství, ale jako takový nenabízí žádný druh metodiky modelování. Také proto není jazyk UML vázán na žádnou specifickou metodiku nebo životní cyklus systému. Přesto je ovšem jazyk UML přednostně využíván ve spojení s metodikou RUP, protože cílem jazyka UML a metodiky RUP je od jejich vzniku podpora nejlepších postupů používaných v softwarovém inženýrství, jak uvádí zdroj.

Objektové modelování považuje svět nebo systém za skupinu vzájemně se ovlivňujících objektů. Objekty pak obsahují určité informace a vykonávají různé funkce. Skupina modelů v jazyce UML tedy obsahuje statickou strukturu systému, ve které stanoví, jaké typy objektů jsou podstatné a jaká je mezi nimi souvislost, a dále pak dynamickou strukturu, ve které zachycuje spolupráci jednotlivých objektů při zajišťování funkcí daného systému. K tomu jazyk UML využívá tři základní stavební bloky, a to Předměty, Relace spojující tyto objekty a Diagramy, znázorňující zajímavé pohledy na systém.

## **Použití UML v našem projektu**

V rámci předložené práce nebyly využity všechny možnosti jazyka UML, ale byly využity vždy takové digramy, které nejvíce napomohly růstu projektu. Nejvíce byly použity diagramy pro zachycení požadavků a diagramy případů užití. Dále byly v potřebných případech zpracovány scénáře jednotlivých případů užití a diagramy aktivit, včetně podrobných popisů jednotlivých akcí.

## **Modelovací nástroj Enterprise Architekt (EA)**

Pro tvorbu diagramů potřebných k vývoji aplikace byl použit nástroj Enterprise Architekt společnosti Sparx Systems. Software posloužil jednak pro zachycování požadavků, tak i pro jejich další rozpracování v podobě různých diagramů. Dále byla využita také matice vzájemných vztahů a to především pro vztah mezi funkčními požadavky a diagramy užití. Díky tomu bylo možné přehledně sledovat, které požadavky na systém jsou již namodelovány a které nikoli. Výsledky modelování byly předávány ostatním členům týmu přes repositář nástroje RTC.

## **6.2 Požadavky**

Získat přehled o požadavcích kladených na systém je vždy zásadní záležitost pro celý zbytek projektu. V našem případě byl základním zdrojem těchto požadavků dokument gamedesign. Díky němu byl sestaven seznam všech užitečných vlastností vytvářeného systému a tyto vlastnosti pak byly postupně přiřazovány k jednotlivým iteracím. Přiřazení je zřetelné z čísla uvedeného za každou vlastností.

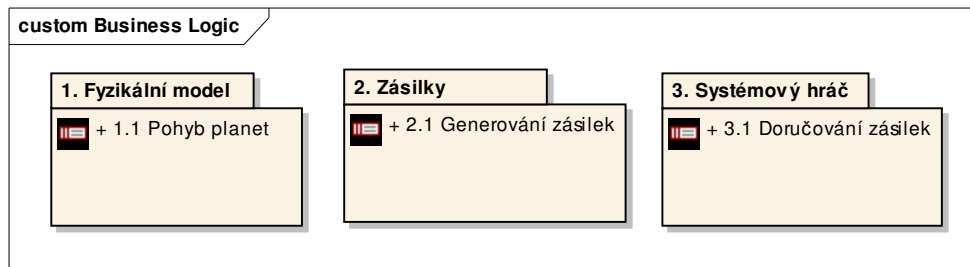
### **Užitečné vlastnosti hry – Game features**

- Systém obsahuje jednu sluneční soustavu s pohyblivými planetami (1)
- Systém obsahuje tři sluneční soustavy – jsou propojené červími děrami jako vláček (4)
- Ve hře je 25 slunečních soustav – není propojena každá s každou (5)
- Systém generuje zásilky na planetách (1)
- Systém generuje více typů zásilek (2)
- NPC ve hře může doručovat zásilky v jedné soustavě (1)

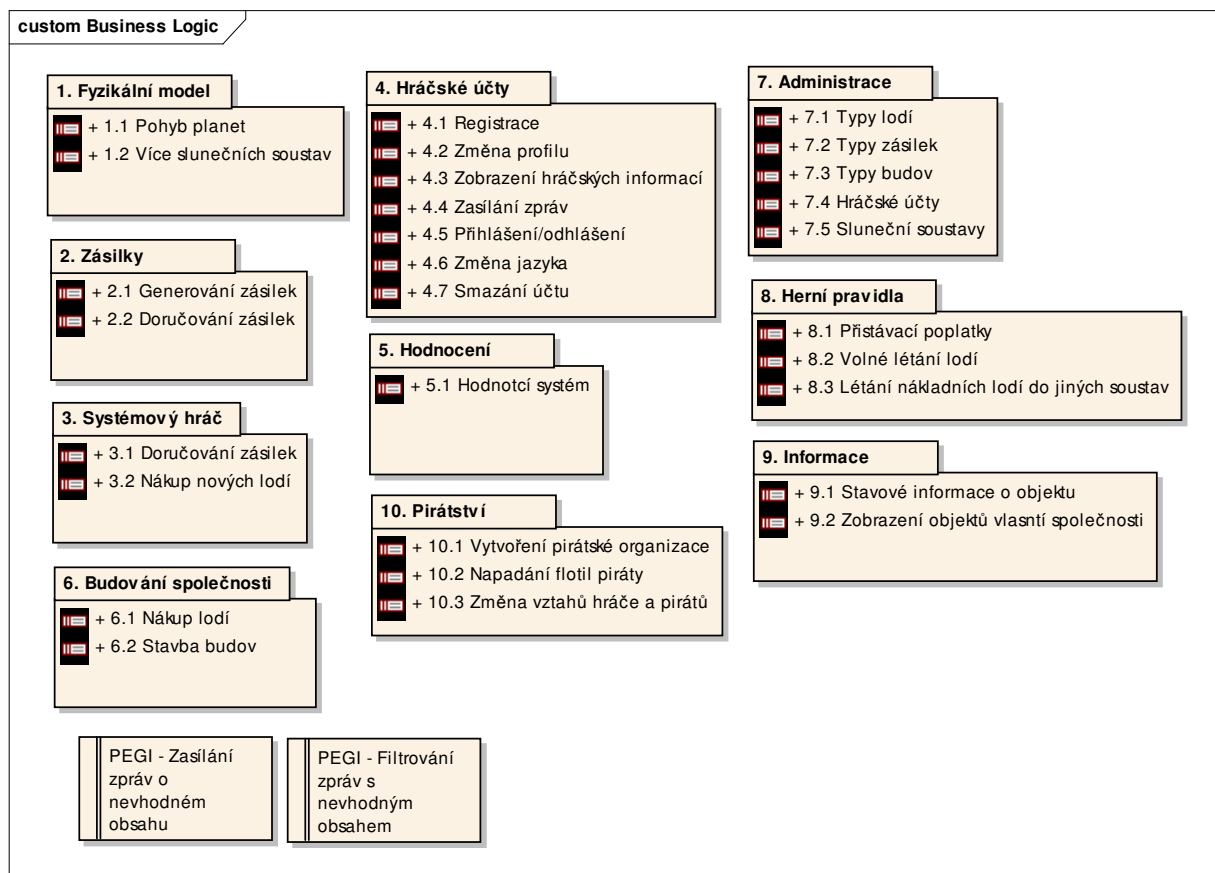
- NPC ve hře může doručovat zásilky mezi planetami v jiných soustavách (5)
- Lze založit hráčský účet, přihlásit se a odhlásit se ze hry (2)
- Uživatelé systému si mezi sebou mohou posílat zprávy (3)
- Hráči se mohou sdružovat do aliancí a přináší jim to výhody (5)
- Hra obsahuje více typů lodí (2)
- Ve hře se objevuje bojová loď bránící nákladní lodi proti pirátům (4)
- Hráč může darovat loď pirátům (4)
- Piráti provádějí útoky na letící nákladní lodě (4)
- Hráč definuje přiřazení bojových lodí ke konkrétním nákladním lodím (4)
- Mezi piráty a hráčem je vzájemný vztah, který ovlivňuje riziko pirátského útoku (4)
- Hráči mohou vydělávat doručováním zásilek – lodě létají mezi planetami (2)
- Do nákladních lodí lze nakládat různé zásilky do vyčerpání kapacity (2)
- Ve hře lze kupovat nové lodě (3)
- Hráč může rozšířit společnost stavbou doků (3)
- Systém umožňuje hráčům doručovat zásilky mezi dvěma různými solárními systémy a plánovat dráhu letu nákladní lodi (5)
- Hráčům je zpoplatněno přistávání na planetách (3)
- Hráčům je účtována variabilní částka za létání lodí dle počtů změn kurzu (4)
- Hráči neplatí přistání ve vlastním doku (3)
- Na každé planetě lze postavit jen určitý počet doků (3)
- Hra obsahuje systém hodnocení (3)

Dále bylo v této fázi iterace vždy nutné definovat vícevýznamové pojmy a slovní spojení tak, aby byla zajištěna jednoznačnost těchto výrazů v rámci týmové komunikace. Slovníček pojmů (Project vocabulary) je uveden v elektronické příloze ve své finální podobě.

Nejpodstatnější částí práce tohoto pracovního postupu se však v každé iteraci stalo stanovení, případně doplnění strukturovaného katalogu požadavků, který odrážel cíle iterace a jí přidělených features. Práce nyní předkládá dva diagramy vytvořené v nástroji EA (obrázek 6.1 a 6.2), které znázorňují stav katalogu funkčních požadavků v první a čtvrté iteraci. Celý vývoj a růst tohoto katalogu požadavků je zachycen na CD, které je přílohou této práce.



Obrázek 6.1 – Stav katalogu požadavků pro první iteraci práce na projektu



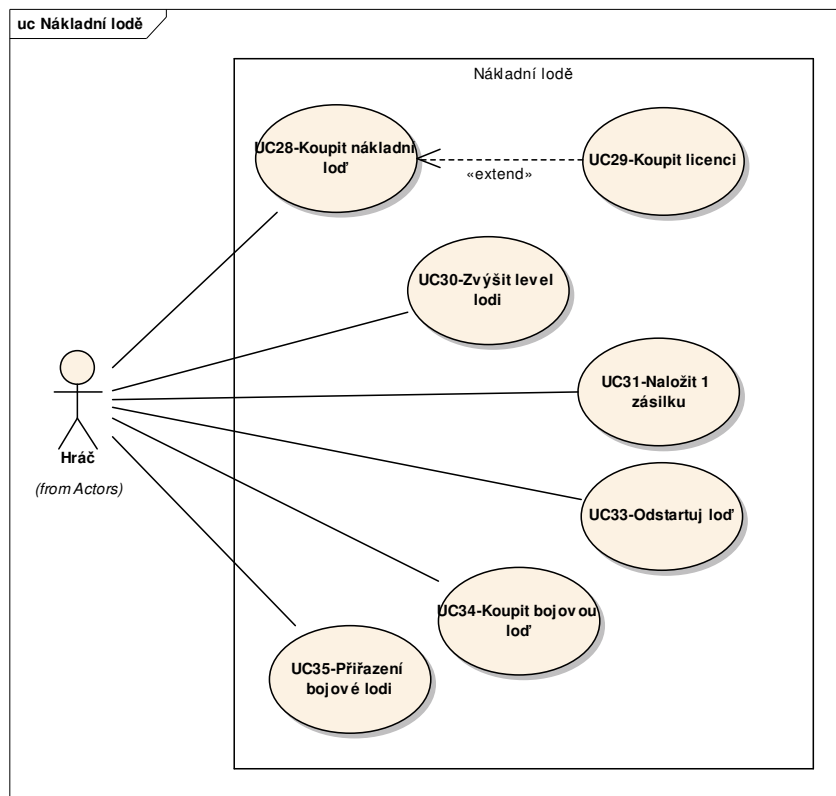
Obrázek 6.2 – Stav strukturovaného katalogu požadavků během iterace 4

V každém projektu se také dále setkáváme s nefunkčními požadavky, které vyjadřují omezení a nároky kladené na vyvíjený software. Při realizaci našeho projektu byly tyto požadavky následující:



- Trvalost hry - Hra je nekonečná  
 - Hra je pravidelně vyhodnocována
- Bezpečnost - Osobní data hráčů jsou chráněna heslem  
 - Hesla jsou kódována
- Rozšiřitelnost - Systém má vícevrstvou architekturu  
 - Objektový přístup k programování
- Stabilita - Aplikace je stabilní do počtu min. 1000 uživatelů

Při sběru a formalizaci požadavků byly také využity diagramy aktivit (příklad na obrázku 6.3), znázorňující možná použití vytvářeného softwaru.



Obrázek 6.3 – Diagram případů užití – hranice systému Nákladní loď

Podobných případů evidujeme v projektu více než čtyřicet. Většina z těchto případů (pokud se nejedná o zcela triviální záležitosti) je doplněna o scénář, který přesně popisuje očekávané chování systému, bude-li jej uživatel ve smyslu daného případu používat.

V předložené práci (příložené CD) se také nacházejí případy užití s více než jedním scénářem. V zásadě by se tyto případy měly rozložit na dva oddělené případy užití. V zájmu snadnější orientace v modelu projektu bylo rozhodnuto zavedené případy užití dále neštěpit. V případě, že u některých případů užití není scénář uveden, jedná se o use-case, který je velice podobný některému, který byl již v práci až na úroveň scénáře propracován. V takové situaci je u daného případu užití pouze upřesňující popis, definující rozdíly od případu již zpracovaného. K této situaci docházelo častěji v pozdějších iteracích, protože práce z pochopitelných důvodů budovala nové herní možnosti na principech již dříve objasněných.

### **Příklad řešení konkrétního požadavku**

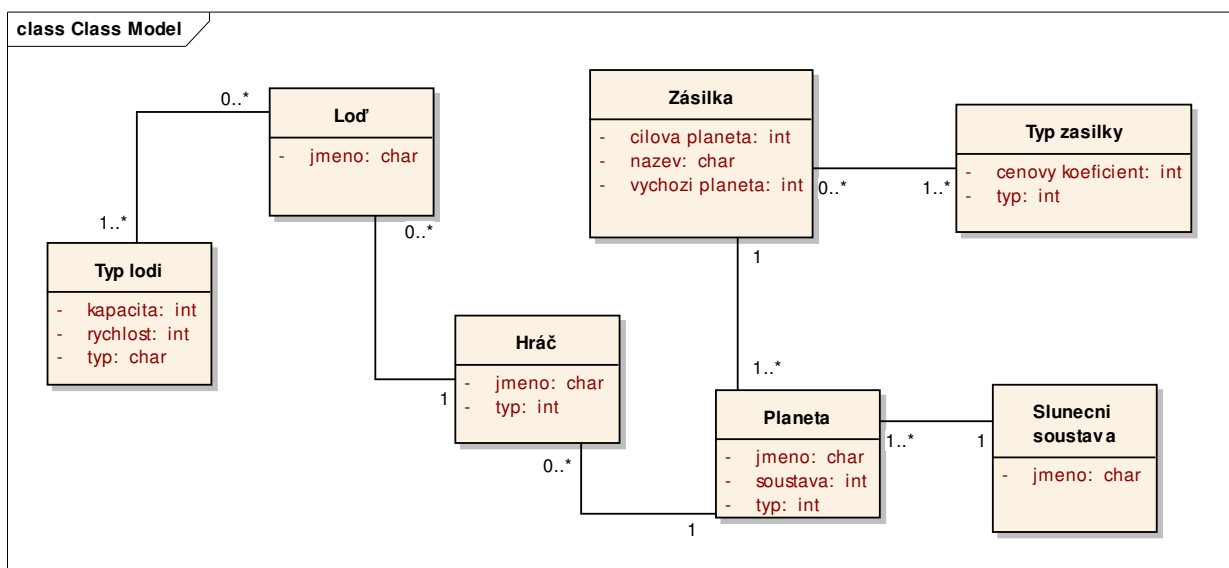
Jedním z řešených požadavků byl v tomto projektu požadavek číslo 6.1 – Nákup lodí, který patří do skupiny požadavků 6 – Budování společnosti. Požadavek byl zpracován do diagramu případů užití a v následujícím kroku byl vytvořen scénář tento požadavek upřesňující. Pro psaní scénářů neexistuje žádná bližší specifikace a to z toho důvodu, že je scénář určen pro malou skupinu zainteresovaných osob a podstatnější důraz než na formu, je kladen na obsah. Formální stránku scénářů se však doporučuje během projektu neměnit.

### **UC14 – Koupit loď**

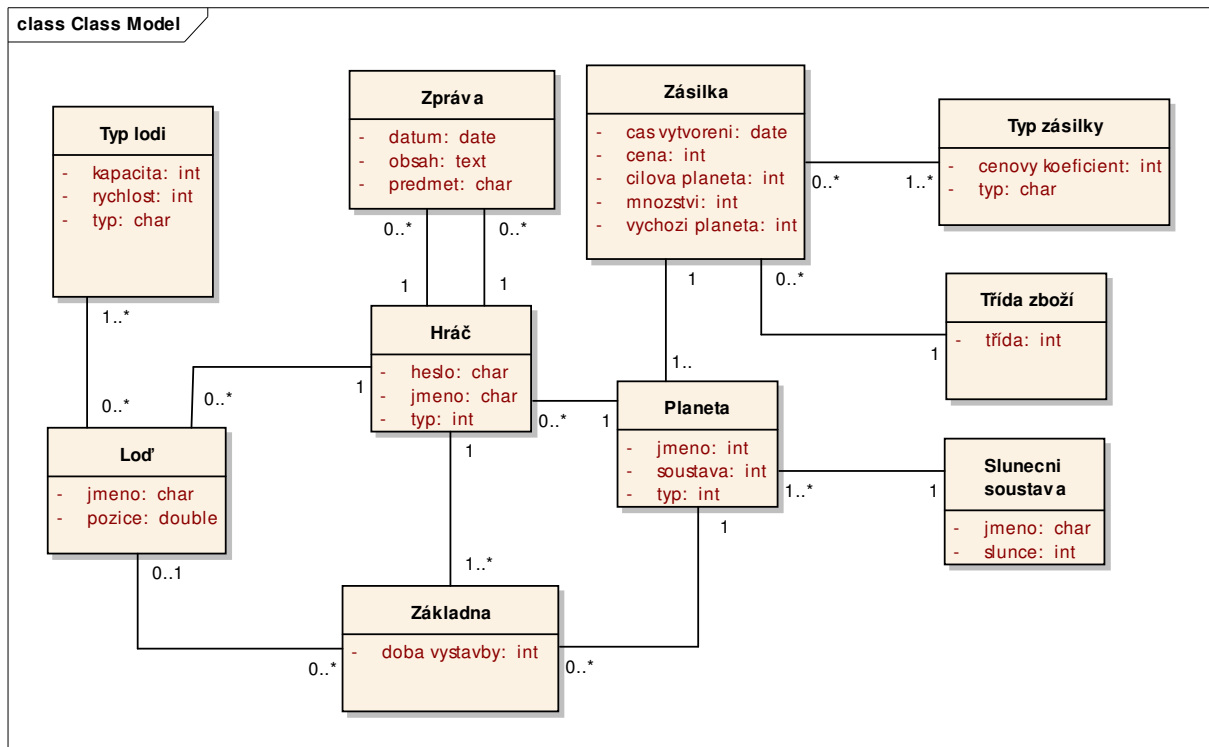
1. Případ spuštěn požadavkem hráče - hráč aktivuje odkaz nákupu lodí
2. Systém zobrazí všechny lodě dostupné ve hře, zvýrazní lodě, které hráč nemůže koupit
3. Hráč vybere loď a aktivuje odkaz koupit
4. Systém se zeptá na název lodi
5. Hráč vyplní název lodi a potvrdí jej
6. KDYŽ „VYPLNĚNO A POTRVZENO“
- 6.1 Systém provede aktualizaci DB, hráč získá danou loď a ta se objeví na domovské planetě
7. KDYŽ „VYPLNĚNO A STORNOVÁNO“
- 7.1 . Systém zobrazí všechny lodě dostupné ve hře, zvýrazní lodě, které hráč nemůže koupit

### 6.3 Analýza

V pracovním postupu analýza byly jako nejdůležitější prvky použity diagramy analytických tříd a diagramy aktivit. Vzhledem k objektovému přístupu k programování byly diagramy analytických tříd vzorem, jak vytvářet objekty při implementaci aplikace. Do tohoto diagramu nezahrnujeme žádná bližší implementační rozhodnutí. Práce opět předkládá k porovnání dva tyto diagramy. První z nich je výsledkem analýzy první iterace, druhý z nich pak výsledkem iterace třetí. Diagramy pro ostatní iterace jsou uvedeny v elektronické příloze práce. Datové typy jednotlivých atributů nejsou konkrétními datovými typy programovacího jazyka, ale slouží především k lepšímu pochopení významu daného atributu. Při práci také nebyly hledány zodpovědnosti tříd, které byly přenechány v kompetenci programátora.



Obrázek 6.4 – Model analytických tříd pro první iteraci

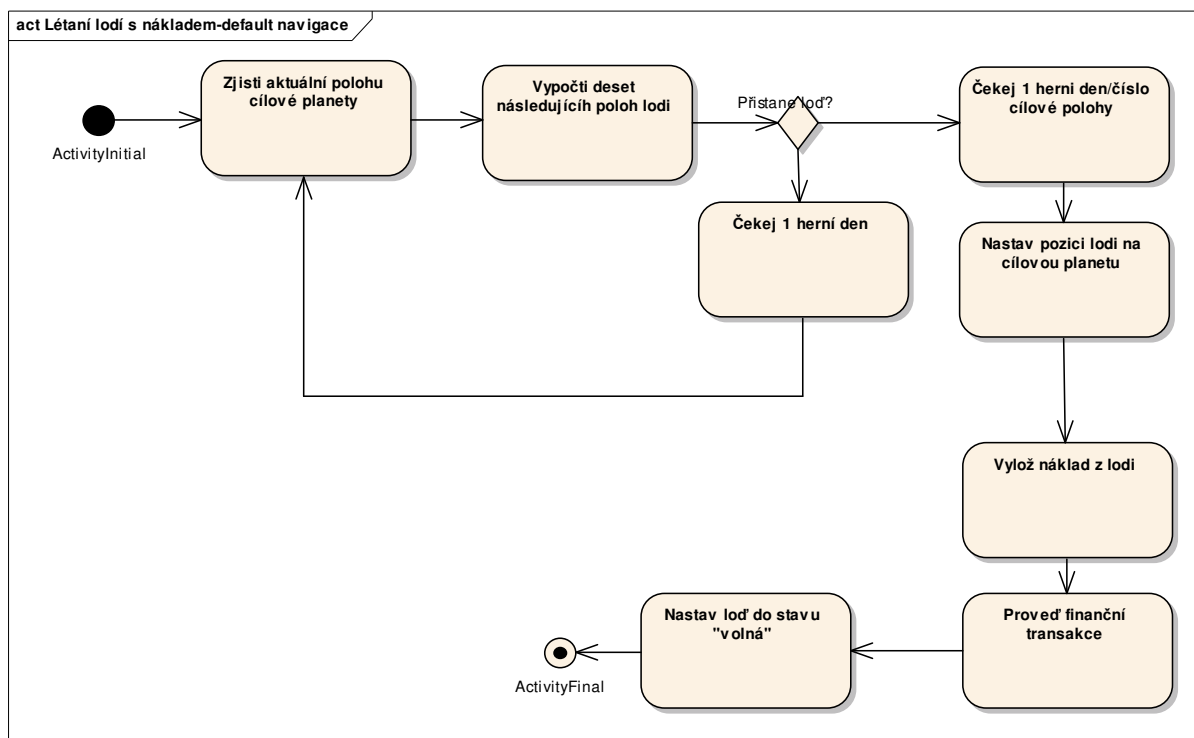


Obrázek 6.5 – Diagram analytických tříd pro třetí iteraci

K zachycení akcí, které systém provádí bez vědomí uživatele byly využity diagramy aktivit. Jeden z těchto diagramů byl již uveden v kapitole 5.3 jako příklad průchodu jednoho požadavku systémem. Mezi akce, zachycené v diagramech aktivit v naší aplikaci patří:

- Generování zásilek
- Létání lodí dle standardní navigace
- Nakládání zásilek NPC hráčem
- Napadání flotil piráty
- Pirátské vydírání
- Platby za přistání
- Darování lodí pirátům a zlepšení vzájemných vztahů

Jako příklad uvádíme diagram aktivit Létání lodí dle standardní navigace, která je klíčovým podkladem pro pohyb jednotlivých lodí všech hráčů.



Obrázek 6.6 – Diagram aktivit: Létání lodí s nákladem dle standardní navigace

## 6.4 Výběr technologie a ověření vhodného výběru

Výběr technologie pochopitelně proběhl v první iteraci a vícekrát se neopakoval. Práce se výběru technologie na tomto místě věnuje proto, že před tímto výběrem bylo nutné prvotní sběr požadavků a analýzu provést. Mezi vhodné technologie, kterými lze náš projekt realizovat patří JSF, PHP, ASP. Práce nyní předkládá stručné informace [24] o těchto technologiích a diskusi výběru vhodné technologie. Je popsán také způsob ověření, zda je vybraná technologie pro realizaci projektu použitelná.

### Servlety a JSP

Spojení servletů a JSP (Java Server Pages) je technologie pro vývoj profesionálních webových aplikací od firmy Sun. Hlavním rysem je oddělení definice uživatelského rozhraní od programování aplikační logiky (vrstva servisní), a tím docílení vývoje podle vícevrstvé architektury. Prostředí je silně objektové. Uživatelské rozhraní se programuje pomocí servletů nebo JSP stránek, ze kterých se pomocí speciálních odkazů přistupuje do Java-Bean, které jsou

součástí aplikační logiky. Servlety jsou ve své podstatě programy, které běží na webovém serveru a působí jako střední vrstva mezi požadavkem přicházejícím z webového prohlížeče nebo jiného HTTP klienta a databází či aplikací na serveru HTTP.

Technologie JSP umožňuje směřovat běžné, statické HTML stránky s dynamicky generovaným obsahem ze servletů. Výsledný kód JSP stránky připomíná svojí strukturou samotný HTML soubor, na pozadí se ale tento soubor převede na servlet, který se tiskne do statické HTML stránky (stejně jako tomu je např. u PHP).

## **PHP**

PHP je volně šířeným skriptovacím programovacím jazykem, je dosti rozšířeným a výkonným nástrojem pro tvorbu dynamických internetových stránek. Nejčastěji je vnořený do HTML stránky a svojí strukturou je velmi podobný jazyku C++. Textové soubory s PHP kódem jsou uloženy na serveru a mají příponu .php. PHP skript se nejprve dle zadaných kritérií zpracuje na serveru a následně se odešle prohlížeči pouze výsledek stejným způsobem jako klasická statická HTML stránka. Výhodou PHP je snadná spolupráce s databázovými systémy – nejčastěji s MySQL.

## **ASP.Net**

ASP.NET je nástupcem ASP, ale obě technologie jsou velmi vzdálené. ASP.NET je založen na CLR (Common Language Runtime – společné programovací prostředí), které je sdíleno všemi aplikacemi postavenými na .NET Frameworku. Stránka v ASP.NET aplikaci obsahuje HTML speciální značky, tzv. serverové komponenty (např. button), které mají své vlastnosti, metody a události. Zároveň má stránka kód na pozadí, který ošetřuje události komponent a stará se o funkcionality celé stránky.

## Diskuse výběru

Práce nyní předkládá přehledný seznam výhod a nevýhod jednotlivých technologií. Poté jsou uvedeny nejdůležitější argumenty, které vedly k rozhodnutí, která technologie bude při realizaci použita.

<b>Servety a JSP</b>	<b>PHP</b>	<b>ASP.Net</b>
+ podpora vícevrstevných aplikací	+ snadnější učení jazyka	+ nezávislost na program. jazyce
+ malé zatížení OS	+ velká flexibilita	+ podpora vícevrstevných aplikací
+ nezávislost na platformě	+ levný hosting	+ obsahuje serverové ovládací prvky
+ generuje validní HTML	+ snadná komunikace s databázemi	+ malý rozdíl web. a jiných aplikací
+ univerzálnost servletů	+ velká komunita vývojářů	+ generuje validní XHTML
		+ pouze jedna kompilace aplikace
- nízké zastoupení hostingu	- chybí zabudované komponenty	- nízké zastoupení hostingu
- nutnost znalosti jazyka JAVA	- nedostupné pokročilé knihovny	- velké znalosti pro malé aplikace
- kompilace při každém spuštění	- málo kvalitních vývoj. prostředí	- závislost na firmě Microsoft

Pro projekt, který tato práce popisuje, bylo využito programovacího jazyka PHP. Pro pohyby objektů byl využit Javascript. Tato práce zaujímá stanovisko, dle kterého se technologie spojení servletů a JSP hodí především pro větší projekty, které se budou dále spravovat nebo rozšiřovat a vývoj je oproti vývoji v jazyce PHP pomalejší. Proti technologii ASP.Net hovoří fakt, že katedra, pro kterou je výsledný software určen, neprovozuje standardně žádný .Net server. U technologie ASP.Net a JSP + servlety byla navíc předpokládána malá množina vývojářů v univerzitním prostředí na rozdíl od skupiny vývojářů, kteří mají zkušenost s využitím PHP.

## Způsob ověření vybrané technologie

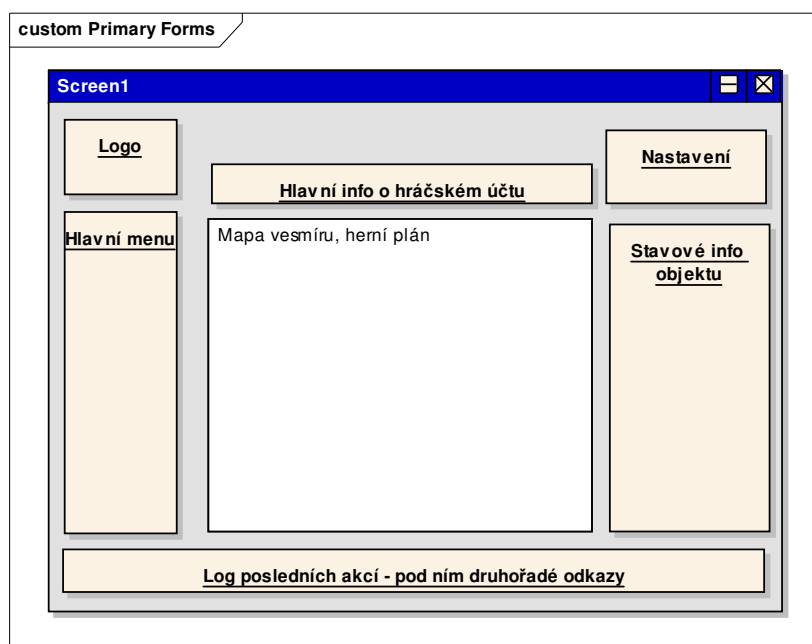
Pro ověření vhodnosti technologie PHP + Javascript byla vytvořena testovací aplikace s dynamickou mapou jedné sluneční soustavy. V zobrazeném okně se také nalézal objekt představující vesmírnou loď a po zadání příkazu tento objekt doletěl na vybranou planetu. Aplikace bez problémů plnila svůj úkol a byla později začleněna do celé výsledné aplikace. Jediným problémem, který při ověřování vyvstal byl neplynulý pohyb objektů při některých rychlostech běhu testovací aplikace. Přestože se objekty pohybovaly po jednotlivých pixelech,

byl tento pohyb velice rušivý. V tuto chvíli bylo ovšem přihlédnuto k tomu, že rychlost běhu výsledné aplikace bude několikanásobně nižší a k pohybu objektů bude docházet ve výrazně kratších intervalech. Proto nebyl shledán tento nedostatek jako důvod k přehodnocení vybrané technologie.

## 6.5 Návrh

Je-li projekt nebo jeho část ve fázi návrhu, je úkolem řešitelského týmu učinit konkrétní implementační rozhodnutí, které upřesní analyzovanou část systému. V našem projektu byla tato rozhodnutí přenechána přímo programátorovi. Jak již bylo uvedeno dříve, nebylo nutné návrhovými modely koordinovat práci více programátorů. Ve fázi návrhu tak proto byl definován pouze uživatelský interface a způsob nasazení aplikace. K těmto účelům posloužily diagram Uživatelského rozhraní – User Interface diagram (UI) a diagram nasazení.

Diagram UI zobrazuje rozložení hlavních ovládacích a informativních prvků na hlavní herní obrazovce. Tomuto rozhraní předchází rozhraní přihlašovací. Představený model není pro implementaci závazný, ale slouží k upřesnění představy budoucího konkrétního grafického návrhu a použitého rozložení prvků na stránce – page layout.



Obrázek 6.7 – Diagram uživatelského rozhraní (User Interface)





- Control                      výpočetní logika aplikace
- Data                            datové podklady pro databázi
- Images                        ilustrace a grafické prvky
- Scripts                        logika klientské části – Javascripty

Při práci bylo s výhodou použito serverového démona CRON, který provádí některé automatické operace bez vědomí uživatelů v pravidelných časových informacích. Tímto způsobem jsou řešeny například veškeré akce NPC hráčů, kterými jsou v naší aplikaci piráti a jedna doručovací společnost, tvořící přirozenou konkurenci zúčastněným hráčům.

Datový model aplikace byl vytvořen v databázi MySQL. Vzhledem k absenci osoby pověřené vývojem datového modelu v realizačním týmu, tvoří tento model v podstatě jen datové entity s potřebnými atributy. V průběhu práce nebyl vytvořen model s vazbami mezi těmito entitami, ani žádná forma aktivní vrstvy tohoto datového modelu.

## **6.7 Testování**

Jak bylo již v práci uvedeno, většina pracovních položek měla stanovenou sadu testů ověřující správnost jejího vyřešení. Testování vyřešených položek probíhalo ručním průchodem sadou testů. Nebyly použité žádné automatické formy testování.

Byla-li v průběhu práce některá z položek označena jako vyřešená, autor práce otestoval aplikaci dle předepsané sady testů. V případě, že byla nalezena odchylka od zadání, byla položka znovu otevřena s přidaným komentářem popisujícím chybu. V případě, že se při testování objevila závada v aplikaci nepřímo související s některým z úkolů aktuální iterace, byla vytvořena nová pracovní položka, která byla vizuálně i svým popisem identifikována jako chyba v aplikaci. Programátor tedy ihned po zobrazení svých pracovních položek tuto chybu našel a mohl pracovat na jejím odstranění.

Vzhledem k tomu, že aplikace byla implementována pouze do stavu betaverze, nebylo možné zapojit do testování velké množství testerů a aplikace tak byla testována převážně tvůrci, konzultantem a několika oslovenými studenty. Návrhy a připomínky těchto osob byly autorům práce zasílány formou krátkých mailových zpráv nebo osobních sdělení. V případě chyby v aplikaci byla konkrétní připomínka zapracována do iteračního plánu. Ostatní podněty se staly

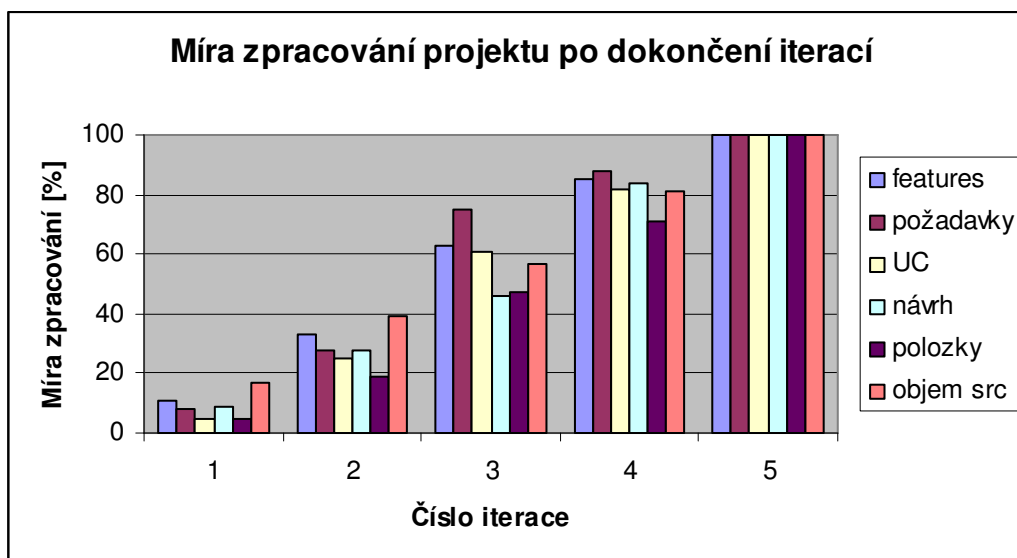
jedním z určujících kritérií při stanovování plánů nové iterace. Vzhledem k omezeným lidským zdrojům a fixní době trvání projektu byla většina připomínek uchována jako seznam možných rozšíření. O nejvýznamnějších možných rozšířeních pojednává zvláštní kapitola této práce. Mimo to jsou tyto připomínky a návrhy na rozšíření hry uvedeny v příloze práce.

## 6.8 Shrnutí objemu prací během iterací

Graf 6.8 názorně ukazuje, jakým způsobem a tempem postupovaly jednotlivé práce během celé doby trvání projektu. Zachycuje tak stav znalosti systému formou požadavků, ukazuje rostoucí objem analyzované části systému, počty řešených pracovních položek a celkový objem zdrojového kódu po uplynutí každé iterace.

Metrika jednotlivých položek uvedených v grafu je následující:

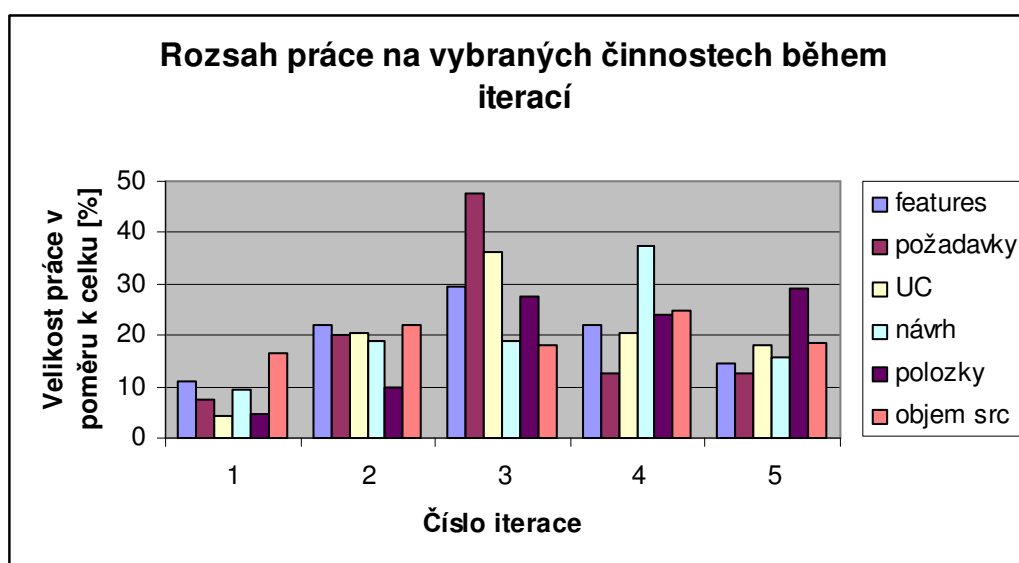
- Užité vlastnosti (features) - počet features
- Požadavky - počet požadavků v katalogu
- Use Cases - počet zpracovaných diagramů
- Návrh - počet diagramů aktivit a zpracovaných scénářů UC
- Pracovní položky - počet řešených pracovních položek
- Objem zdrojového kódu - velikost zdrojových souborů skriptů v kilobytech



Graf 6.1 – Velikost procentuálních dokončených částí po jednotlivých iteracích

Při protnutí vrcholů jednotlivých barevných sloupců pomyslnou křivkou vidíme téměř lineární růst jednotlivých sloupců. Z toho je zřejmé, že rychlost týmu byla téměř konstantní a práce probíhaly vyrovnaně.

Další neméně zajímavou statistiku předkládá graf 6.2, na němž je zachycen graf objemu řešených problémů v každé iteraci zvlášť a dává tak přehled o procentuálním příspěvku dané iterace do finálního celku, jímž je rozuměna vytvořená betaverze webové hry.



Graf 6.2 – Procentuální rozsah práce na vybraných činnostech v jednotlivých iteracích

Pro kompletní celkovou statistiku projektu uvádíme dále absolutní čísla některých vybraných atributů projektu:

- Počet zpracovaných užitečných vlastností: 27
- Známé požadavky systému: 40
- Počet případů užití: 44
- Počet návrhových scénářů a diagramů: 32
- Počet řešených pracovních položek: 62
- Velikost zdrojového kódu skriptů: 420kB

## 7. DALŠÍ PRÁCE BĚHEM PROJEKTU

Během práce byly také zpracovávány další úkoly, které měly zajistit úspěšnou implementaci celé práce. V zásadě se jednalo o tyto činnosti (uvádíme i osobu zodpovědnou za provedení příslušné činnosti):

- Tvorba zadání pro grafiku a databáze (Zbyněk Neudert)
- Koordinace grafických prací (Zbyněk Neudert)
- Tvorba grafických podkladů (Alexi Goubarev)
- Tvorba informativních textů ke hře (Zbyněk Neudert)
- Tvorba tutoriálu (Zbyněk Neudert)

O nutnosti vytvoření zadání pro implementační, databázové i grafické práce hovoří již kapitola 5.2 pojednávající o sestavení řešitelského týmu. Tato zadání jsou uvedena v příloze práce.

Pro potřeby grafického designu byl sestaven seznam všech komponent, které bylo nutno navrhnout a jednotlivé komponenty byly označeny prioritou, dle které se grafik-ilustrátor řídil a tím byly tyto práce koordinovány. Seznam těchto komponent je uveden v příloze práce včetně priorit. Pro úkolování a koordinaci ilustrátorských prací byly použity zprávy, zasílané prostřednictvím elektronické pošty nebo přes Instant Messenger (IM). Předání úkolů prostřednictvím nástroje RTC nebylo na tomto místě vhodné vzhledem k potřebným zkušenostem s instalací a obsluhou tohoto nástroje. Instalace RTC klienta pro potřeby řízení grafických prací byla shledána jako nežádoucí zatížení řešitelů práce.

V neposlední řadě bylo nutné vytvořit informativní texty, které jsou v aplikaci dostupné jako nápověda a usnadňují tak hráčům orientaci ve hře. Pro nové hráče se zájmem o hru bylo třeba dále vytvořit stručný a výstižný tutoriál, popisující základní myšlenku hry a podávající nejzákladnější instrukce, díky kterým bude hráč na začátku hry vědět, co je jeho úkolem a jak má ve hře a s hrou pracovat. Text tutoriálu je také uveden v příloze.



## 8. ZÁVĚR

Tato diplomová práce, věnující se předložení konceptu a týmové implementaci online webové hry pro středoškolské studenty, sloužící jako část webové prezentace Katedry informatiky a výpočetní techniky Západočeské univerzity v Plzni, splňuje zadání ve všech bodech. Začátek práce je věnován problematice her a jejich návrhu, jsou předloženy informace o specifických vlastnostech her a o nárocích, které jsou na hry kladeny. Dále jsou uvedeny informace o specifikách návrhu her a o zvláštích her, které jsou určeny pro online hru velkého počtu hráčů (Massively-Multiplayer Online Game – MMOG) . Práce se také věnuje možným metodikám vedení softwarových projektů. Zvláště se pak věnuje projektům s agilními prvky a předkládá možné nástroje, které usnadňují vedení a práci v týmu při takovýchto projektech. Práce dále předkládá vizi výsledného projektu v dokumentu gamedesign, který dostatečně popisuje hru a umožňuje tak její kompletní realizaci. Poslední část práce je poté věnována realizačním pracím z pohledu vedoucího člena týmu. Práce tak dokumentuje celý projekt od stanovení vize až po vytvoření nasaditelné betaverze. Věnujeme se zde všem technickým disciplinám v průběhu projektu a představujeme jejich náplň napříč jednotlivými iteracemi.

### Možnosti rozšíření

Vzhledem k tomu, že při implementaci aplikace bylo dosaženo stavu betaverze, je vhodné výslednou hru dále rozšiřovat. Jedná se především o zpracování připomínek konzultanta práce, ale také o nabídnutí nových herních možností uživatelům. Z tohoto hlediska je stále dostačující vytvořený dokument gamedesign, který popisuje původně plánovaný rozsah projektu. Za klíčové téma považujeme rozšíření herních možností pro hráče, kteří se sdruží do aliance. Jednou z možných rozšíření se tak může stát možnost plánování společných letů, což dá hráčům silovou převahu nad pirátskými loděmi, které budou útočit na jejich flotily. Další neméně zajímavou vlastností hry by se měla stát možnost naprogramování vlastní lodní navigace, která hráčům umožní dosahovat cílových planet v kratším čase a s menšími náklady. Tyto navigace by potom mohli hráči ve hře prodávat, čímž by získali velké finanční prostředky pro rozvoj své společnosti, nebo by je užívali sami, čím by si zajistili konkurenční výhodu před společnostmi jiných hráčů. Z dalších možností ještě zdůrazněme nabídku vydělávání jinými způsoby než pouhým

doručováním zásilek. Hra by mohla umožňovat obchod s technologiemi, jak bylo nastíněno výše, nebo vydělávat převážením zboží, které má v každé části galaxie, nebo na každé planetě jinou hodnotu. Stejně zajímavou možností pro hráče může být plnění úkolů, nebo hledání různých předmětů a objektů v galaxii. Dále je možné hru rozšířit o možnost pronájmu doků. Dokovací stanice na různé úrovni je schopna obsloužit jen určitý počet lodí. Hráči by se poté mohli rozhodnout vybudovat co největší síť těchto stanic a za vhodnou cenu je pronajímat ostatním hráčům.

### **Zhodnocení projektu**

Z předloženého textu je zřejmé, že se projekt dotýká problematiky realizace středně rozsáhlých projektů v univerzitním prostředí. Proto si na tomto místě práce dovoluji předložit některé postřehy, které s touto oblastí úzce souvisí. V univerzitním prostředí nelze beze změn využívat osvědčené agilní metodiky, používané v prostředí komerčním. Těžiště problému nacházíme v situaci, kdy nelze žádného člena realizačního týmu přinutit odvést kvalitní práci a jsme tak odkázáni pouze na osobní zájem a přesvědčení každé zúčastněné osoby. Jediným možným zlepšením této situace je stanovení realizačních nebo i návrhových prací jako povinných prací některých vhodných předmětů. Stejně tak není možné přejmout bez modifikace některou z metodik, které jsou úspěšně používány v komerční praxi. Jedním z důvodů je samozřejmě výše uvedený problém s motivací a výkonem členů týmu, dalším problémem je nemožnost stanovení přesného časového harmonogramu, protože časové možnosti studentů nejsou předem odhadnutelné a nelze s nimi ve větší míře kalkulovat. Dovolujeme si vyslovit doporučení, dle kterého lze pro projekty v univerzitním prostředí využít částečně modifikovanou metodiku FDD tak, jak bylo představeno v této práci a doporučujeme zajistit mohutnost realizačního týmu podporou v jednotlivých vyučovaných předmětech a to i formou paralelně pracujících týmů.

Hlavní nedostatky projektu vidíme v pouze částečné implementaci v porovnání s původní vizí projektu a ne vždy ideální uživatelskou přístupností funkcí ve výsledné aplikaci. Dosažená úroveň zpracování nemůže být pro webovou prezentaci katedry bez dalších úprav použita a bude proto nutné na projektu dále pracovat. Vytvořenou hru ovšem lze dle našeho názoru uveřejnit jako betaverzi plánovaného herního systému, což umožní získat více hráčů pro finální verzi a také tím bude zajištěno několikanásobně hodnotnější testování.



Přes výše uvedené nedostatky považujeme práci za úspěšně dokončenou. Stanovený koncept je dostačujícím podkladem pro další realizační práce a vytvoření použitelné betaverze hry je vzhledem k počtu lidí, kteří na aplikaci pracovali, dobrým výsledkem projektu.



## 9. PŘEHLED POUŽITÝCH ZKRATEK

<b>DSDM</b>	Dynamic Systems Development Method
<b>EA</b>	Enterprise Architekt
<b>FDD</b>	Feature Driven Development
<b>FPS</b>	First Person Shooter
<b>KIV</b>	Katedra informatiky a výpočetní techniky
<b>MMOG (MMO)</b>	Massively Multiplayer Online Game (Massively Multiplayer Online)
<b>MMOFPS</b>	Massively Multiplayer Online First Person Shooter
<b>MMORPG</b>	Massively Multiplayer Online Role Playing Game
<b>MMORTS</b>	Massively Multiplayer Online Real Time Strategy
<b>NPC</b>	Non Player Character
<b>OMG</b>	Online Multiplayer Game
<b>RPG</b>	Role Playing Game
<b>RTC</b>	Rational Team Concert
<b>RTS</b>	Real Time Strategy
<b>RUP</b>	Rational Unified Process
<b>UML</b>	Unified Modeling Language
<b>XP</b>	Extreme Programming



## 10. POUŽITÉ ZDROJE INFORMACÍ

- [1] Zbyněk Neudert. Návrh webové hry (Bakalářská práce). ČVUT Praha, Fakulta elektrotechnická, 2008
- [2] Online hra. (30.3.2008). Wikipedie: Otevřená encyklopedie  
[http://cs.wikipedia.org/w/index.php?title=Online\\_hra&oldid=2410163](http://cs.wikipedia.org/w/index.php?title=Online_hra&oldid=2410163)
- [3] Budovatelská strategie. (27. 12. 2007). Wikipedie: Otevřená encyklopedie.  
[http://cs.wikipedia.org/w/index.php?title=Budovatelsk%C3%A1\\_strategie&oldid=2099928](http://cs.wikipedia.org/w/index.php?title=Budovatelsk%C3%A1_strategie&oldid=2099928)
- [4] Petra Soukupová. POLEMIKA: Jsou počítačové hry nebezpečné?  
<http://www.21stoleti.cz/view.php?cislocianku=2006021722>
- [5] PEGI Pan European Game Information - ORGANIZACE  
<http://www.pegi.info/cs/index/id/60/>
- [6] PEGI Online – Co je PEGI Online?  
<http://www.pegionline.eu/cs/index/id/61/>
- [7] Jesse Schell. The Art of Game Design. Morgan Kaufmann Publishers, Burlington, USA, 2008
- [8] Keith Clinton. Beyond Scrum: Lean and Kanban for Game Developers  
[http://www.gamasutra.com/view/feature/3847/beyond\\_scrum\\_lean\\_and\\_kanban\\_for\\_.php](http://www.gamasutra.com/view/feature/3847/beyond_scrum_lean_and_kanban_for_.php)
- [9] Per Kroll, Philippe Kruchten. The Rational Unified Process Made easy. Addison Wesley, 2006
- [10] Winston Royce. Managing the Development of Large Software Systems. Proceedings of IEEE WESCON, 1970
- [11] Barry Boehm. A Spiral Model of Software Development and Enhancement. ACM SIGSOFT Software Engineering Notes, 1986
- [12] Co jsou agilní metodiky vývoje software? (25.4.2010)  
[http://www.reengine.cz/index/agilni\\_metodiky.do](http://www.reengine.cz/index/agilni_metodiky.do)
- [13] Tomáš Kotrla. Agilní metodiky vývoje software (Diplomová práce). MUNI Brno, Fakulta Informatiky, 2005
- [14] Kent Beck. Extrémní programování. Grada, 2002
- [15] Benjamin Voigt. Dynamic System Development Metod. Zurich, 2004

- [16] Alena Buchalceková. Metodika Feature-Driven Development neopouští modelování, a přesto přináší výhody agilního vývoje. KIT VŠE Praha, 2008
- [17] Subversion, Subversion Features  
<http://subversion.apache.org/features.html>
- [18] Bugzilla - Bug tracking pro váš projekt  
<http://www.czilla.cz/produkty/bugzilla/>
- [19] What is Flayspray?  
<http://flayspray.org/about>
- [20] Wiki Design Principles (6. 05. 2010)  
<http://c2.com/cgi/wiki?WikiDesignPrinciples>
- [21] Team Foundation Server: At Work  
<http://msdn.microsoft.com/en-us/library/ms364061.aspx>
- [22] Rational Team Concert, A collaborative software development environment  
<http://jazz.net/projects/rational-team-concert/>
- [23] Jim Arlow, Ila Neustadt. UML2 a unifikovaný proces vývoje aplikací. Computer Press, Brno, 2007
- [24] Michael Minář. Implementace prezenční vrstvy webové hry (Bakalářská práce). ČVUT Praha, Fakulta elektrotechnická, 2008

## A Plány jednotlivých iterací

### Plán iterace I

**Zahájení iterace:** 17.11.2009

**Ukončení iterace:** 20.12.2009

#### Cíle iterace:

Hlavním cílem je představení vize celého projektu a analyzování základních částí hry. V rámci iterace vznikne nehratelná aplikace s dynamickou sluneční soustavou a systémovým doručováním vygenerovaných zásilek.

#### Klíčové úkoly:

<i>Název</i>	<i>priorita</i>	<i>přiřazeno</i>	<i>pracnost (h)</i>
Dokument Gamedesign	vysoká	Zbyněk Neudert	20
Analýza a návrh s ohledem na cíle	normální	Zbyněk Neudert	10
Datový model s ohledem na cíle	normální	Václav Palík	5
Základní návrh grafického designu a UI	nízká	Dušan Záhoranský	20
Programování aplikace	normální	Petr Kellnhofer	30

#### Kritéria úspěšnosti:

- přesný popis hry v dokumentu gamedesignu
- úspěšné naprogramování aplikace v navržené grafické úpravě

## Plán iterace II

**Zahájení iterace:** 04.01.2010

**Ukončení iterace:** 1.3.2010

### Cíle iterace:

Cílem iterace je vytvoření první verze hratelné aplikace, která umožňuje základní správu herních účtů a doručování zásilek hráčem. Jednotlivé objekty ve hře budou mít od této iterace více typů (lodě, zásilky apod.) Pro doručování zásilek je třeba analyzovat způsob letu a navigace vesmírných lodí, pravidla přepravování a doručování zásilek.

### Klíčové úkoly:

<i>Název</i>	<i>priorita</i>	<i>přřazeno</i>	<i>pracnost (h)</i>
Analýza a návrh s ohledem na cíle	normální	Zbyněk Neudert	30
Úprava datového modelu	normální	Václav Palík	5
Finální návrh grafického designu a UI	normální	Dušan Záhoranský	40
Programování aplikace	normální	Petr Kellnhofer	40

### Kritéria úspěšnosti:

- hratelná aplikace dle cílů iterace



## Plán iterace III

**Zahájení iterace:** 2.3.2010

**Ukončení iterace:** 22.3.2010

### Cíle iterace:

Hlavním cílem zdokonalení aplikace vytvoření v předešlé iteraci. Hra obsahuje hodnotící systém a lze již svou doručovatelskou společnost rozšiřovat nákupem lodí a stavbou objektů. Přistávání na planetách je zpoplatněno. Hráči mohou měnit své profily, zasílat si mezi sebou zprávy.

### Klíčové úkoly:

<i>Název</i>	<i>priorita</i>	<i>přřazeno</i>	<i>pracnost (h)</i>
Analýza a návrh s ohledem na cíle	normální	Zbyněk Neudert	30
Úprava datového modelu	normální	Petr Kellnhofer	5
Programování aplikace	normální	Petr Kellnhofer	40

### Kritéria úspěšnosti:

- hratelná aplikace dle cílů iterace

## Plán iterace IV

**Zahájení iterace:** 23.3.2010

**Ukončení iterace:** 12.4.2010

### Cíle iterace:

Hlavním cílem zdokonalení aplikace vytvoření v předešlé iteraci. Při létání vesmírných lodí vznikají variabilní náklady rozdílné pro každou cestu. Hráči mají nově možnost nakupovat, vylepšovat a spravovat své nákladní lodě. Ve hře se objevují piráti, která napadají letící nákladní lodě jednotlivých hráčů

### Klíčové úkoly:

<i>Název</i>	<i>priorita</i>	<i>přřazeno</i>	<i>pracnost (h)</i>
Analýza a návrh s ohledem na cíle	normální	Zbyněk Neudert	30
Úpravy datového modelu	normální	Petr Kellnhofer	5
Programování aplikace	normální	Petr Kellnhofer	40

### Kritéria úspěšnosti:

- hratelná aplikace dle cílů iterace
- existence nového typu lodí a fungující NPC

## Plán iterace V

**Zahájení iterace:** 20.4.2010

**Ukončení iterace:** 11.5.2010

### Cíle iterace:

Hlavním cílem je vytvoření nasaditelné betaverze celého projektu. Hráčům je umožněno přepravovat zásilky mezi více slunečními systémy a mají také možnost sdružovat se do aliancí, což jim přináší výhody. Ve hře je alespoň 20 slunečních soustav. Do hry bude nasazen vytvořený grafický návrh.

### Klíčové úkoly:

<i>Název</i>	<i>priorita</i>	<i>přřazeno</i>	<i>pracnost (h)</i>
Analýza a návrh s ohledem na cíle	normální	Zbyněk Neudert	25
Úprava tutoriálů a příruček	normální	Zbyněk Neudert	10
Úpravy datového modelu	normální	Petr Kellnhofer	5
Programování aplikace	normální	Petr Kellnhofer	40

### Kritéria úspěšnosti:

- hratelná aplikace dle cílů iterace v nové grafické úpravě
- lze přepravovat zásilky ve všech soustavách, hráči mohou vytvořit aliance



## **B Design reviews**

### **DESIGN REVIEW - návrh úprav projektu**

Projekt: **SPACE BUSSINES – WEBOVÁ HRA PRO KIV**

Stav projektu k datu: 22.12.2009

#### **Popis úprav:**

V této fázi projektu bylo možné analyzovat pouze „view“ část hlavní webové stránky aplikace, kde má docházet k plánování zakázek uvnitř jednoho systému. Systém vypadal poměrně realisticky, zdál se sledovat fyzikální model, použitá metoda načtení hodnot z databáze se nezdála mít příliš velký dopad na plynulost pohybu. Přesto lze vytknout modelu jeden funkční a jeden estetický nedostatek:

1. při zneaktivnění stránky se někdy pohyb zastaví, takže při opětovném obnovení aktivity okna (gain focus) dojde k „přeskoku“ po načtení aktuální pozice. Je možné, že tento problém nastává pouze u některých prohlížečů nebo konkrétního systémového nastavení.
2. ikony představující objekty by bylo vhodné převést do formátu s transparentním pozadím odpovídajícím barvě pozadí hvězdného systému. Tak se odstraní bílá „aura“ a objekty lépe splynou s okolím. Tato úprava bude znatelná zejména u centrálního „Slunce“.

V Praze dne 28.12.2009

RNDr. Karel Neudert

## DESIGN REVIEW - návrh úprav projektu

Projekt: **SPACE BUSSINES – WEBOVÁ HRA PRO KIV**

Stav projektu k datu: 15.03.2010

### Popis úprav:

Cílem 2. iterace bylo vytvořit hratelné prostředí, tj. jednoduchou správu uživatelského účtu a základní herní funkčnost – doručování zásilek.

a) První část se celkem podařila splnit, formulář pro editaci herních údajů odesílá informace k uložení na server. Nepodařilo se mi však změnit heslo, což by určitě potenciální hráči využili. První návrh úpravy se tedy týká revize stránky pro spravování účtu, ošetření situací při vyplnění nevhodných údajů (prázdné, krátké nebo zase příliš dlouhé heslo z pohledu databázového pole,...).

b) Hratelnost aplikace. Tato část plánu se mi nepodařila otestovat, protože aplikace při pokusu o vybrání skončí s chybou v proceduře `ShipActionController.inc/ProcessActionLine($action)` – řádek 308, kdy se pokouší načíst prázdný objekt (null) místo objektu typu `Delivery`. Nepodařilo se mi nasimulovat žádnou situaci, kdy by aplikace nespada, takže jde o poměrně závažnou chybu.

c) V grafické části neproběhly žádné viditelné změny kromě přidání dalších objektů (lodí). Předpokládám, že jde o dočasné obrázky, nicméně by asi bylo v budoucnu vhodné nějak vyřešit jejich neustálé překrývání – velikostí, nějakým menu nebo jinak.

V Praze dne 15.3.2010

RNDr. Karel Neudert

## DESIGN REVIEW - návrh úprav projektu

Projekt: SPACE BUSSINES – WEBOVÁ HRA PRO KIV

Stav projektu k datu: 29.03.2010

### Popis úprav:

Registrace nového uživatele:

Po vyplnění všech polí proběhne registrace úspěšně, i když funkce CreateNewBase() vypíše chybějící argument 3. Registrace si vyžádá potvrzení přes e-mail (pro finální verzi bylo by dobré doplnit odesilatele zprávy a standardizovat html obsah – někteří klienti mají problém s kódováním nebo „pochopením“ obsahu jako hypertext), po kliknutí na odkaz se registrace dokončí (také tato stránka by mohla ve výsledku vypadat trochu reprezentativněji).

Hratelnost:

V této fázi je k dispozici jediná loď, kterou lze převážet různé druhy zboží za různou cenu. „Plutonium“ nelze převážet, ačkoliv to není z důvodu nedostatečné kapacity nebo doletu. V případě, že je loď na cestě, nelze ji využít pro jiný úkol.

Tyto 3 poznatky jsou zcela v pořádku a pravděpodobně naplňují smysl hry, ale z pohledu hratelnosti bych navrhnul několik úprav: přidat nějaký statistický přehled – tržby, počet a stav lodí, menu pro „obnovu vozového parku“ (nákup/prodej lodí). Grafika: současně se vznikem herního menu skrýt dokující lodě – obrázek lodí zcela překrývá planetu a bylo by velmi logické opouštět planetu rovnou s nákladem. To sice lze, ale jedině v případě, že si uživatel planetu „označí“ ještě před příletem. A poslední připomínka se týká algoritmu hledání cesty, která zcela ignoruje přírodní překážky. Pokud to není přímo úmyslem, bylo by dobré ošetřit tuto automatickou navigaci buď trestnými body (poškození lodí, mizející kapitál na opravu apod.) nebo zakázanou zónou pohybu – zejména pokud jde o centrální hvězdu.

Vzhledem k tomu, že se začíná rýsovat i nějaké herní pozadí nebo příběh, navrhnul bych také přidání nějakých pasivních textů, vysvětlivek, popisků a případně i nápovědy. Například mi chybí odůvodnění, proč s testovací lodí není možné převážet plutonium, i když to kapacitně zvládá.

Závěr hodnocení:

Tato programová iterace již vypadá hratelně a začíná ukazovat velký potenciál. Kromě zmíněných návrhů jsem nenašel žádný závažnější problém a myslím, že autoři odvedli dobrou práci.

V Praze dne 29.3.2010

RNDr. Karel Neudert



## **C Seznam grafických prvků hry a jejich priority**

### ***Nejvyšší priorita***

- Pozadí hrací plochy
- Slunce
- Planety
- Logo hry
- Layout stránek – user interface – rozložení menu a hrací plocha

### ***Střední priorita***

- Červí díra nebo jump gate
- Souhrnná mapa vesmíru (sektor galaxie s vyobrazením soustav a propojení mezi nimi)
- Profil, nastavení a nabídky – vzhled aktivovaných položek menu
- Ikony lodí
- Symbolické zobrazení vztahů s piráty – přátelské – neutrální - nepřátelské
- Piktogramy nebo malé ikony označující domovskou planetu a budovy, lodě které parkují

### ***Nízká priorita***

- Vesmírná loď typ standart
- Vesmírná loď typ „na nebezpečný náklad“
- Vesmírná loď typ nákladní
- Vesmírná loď typu guard – hlídač nákladní lodi
- Obyčejný obchodní balík
- Nebezpečná zásilka
- Zásilka do nákladní
- Vesmírná loď typ pirát
- Souboj lodí - koláž
- Zničené lodě – koláž
- Výhry a reklamní předměty

Veškeré ilustrace a grafické prvky jsou dílem Alexe Goubareho a jejich vyobrazení je k dispozici v elektronické příloze práce.



## **D Zadaní semestrální práce pro předmět KIV/DB2**

### **CÍL:**

Hlavním cílem je zdokonalení aplikace vytvořené v předešlých iteracích vývoje. Hra bude obsahovat hodnotící systém a bude možno svou doručovatelskou společnost rozšiřovat nákupem lodí a stavbou objektů. Přistávání na planetách bude zpoplatněno. Hráčům bude umožněno zasílat si vzájemně zprávy a smazat vlastní hráčský účet.

### **POŽADAVKY PRO TUTO ITERACI**

#### ***Bodový zisk hráče***

Hráčům jsou za jejich růst ve hře připisovány body, které slouží k sestavení hodnotícího žebříčku.

#### ***Mazání účtů***

Hráči mohou smazat svoje účty. Nechtěnému smazání brání 24hodinová lhůta ve které nesmí dojít k přihlášení do hry.

#### ***Zasílání zpráv mezi uživateli – práce se zprávami***

Hráči si mezi sebou mohou zasílat zprávy. Příchozí zprávy lze číst a mazat.

#### ***Koupě vesmírné lodě***

Každý hráč ve hře má možnost ze svého zisku pořídit novou vesmírnou loď.

#### ***Stavba objektu***

Každý hráč má možnost ze svého zisku postavit novou budovu. Pro účely této iterace se jedná pouze o dokovací stanici, která umožní danému hráči bezplatné přistání na příslušné planetě.

#### ***Poplatky za přistávání na planetě***

Při přistání na planetě, na které nemá hráč vlastní dokovací stanici, je z jeho účtu odečtena předem definovaná částka.

## **OBSAH PRÁCE V RÁMCI SEMESTRÁLNÍ PRÁCE DB2**

Vaším úkolem bude:

- upravit stávající DB, navrhnout její rozšíření pomocí nových atributů nebo tabulek
- navrhnout procedury a triggerly pro aktivizaci databáze aby nemuselo být vše řešeno na úrovni bussines logiky
- v případě zájmu o programování webového rozhraní pracovat na administrační části hry

## **MODEL**

Předchozí student DB1 vytvářející původní DB nijak nepropojil navržené tabulky - nelze tedy předložit relevantní ERA model. Databázi je tedy nutné v tomto kroku přepracovat ovšem zachovat stávající strukturu tabulek a atributů. Zájemcům o tuto práci bude zaslán SQL skript generující aktuální stav databáze.

**!!!!**

**Na hře se stále pracuje a s každým dalším krokem dochází k nejnnutnějším změnám v databázi. Proto berte uvedený seznam požadavků jako orientační. V případě zájmu o tuto sem. práci budete při prvním kontaktu seznámeni s aktuálním stavem projektu, databáze a aktuálně řešenými požadavky.**

## **E Zadání semestrální práce pro předmět KIV/KPG**

### **Generátor planet pro webgame**

Pro vznikající webgame, s tematikou obchodování ve vesmíru, vytvořte generátor 2D obrázků planet. Výstup generátoru ať je náhodný obrázek planety minimálně 300x300px. Obrázek bude pro použití na konkrétní herní ploše zmenšován. (Hrací plocha se formátuje pro každé rozlišení obrazovky zvlášť.) Není proto nutné věnovat se příliš detailům povrchu apod. Generátor by měl dokázat vygenerovat planety různých tvarů, barev, s různým počtem měsíců, různými prstenci a pod. Obrázky budou umístěné na průhledném pozadí. Barva, která bude průhledná nesmí být nastavena natvrdo, takže bude možné vytvořit například planety na černém pozadí, kde průhlednou barvou bude černá. Použité barvy planet musí být bezpečné pro web.

Odevzdání do 3.5.2010

### **Statické grafické prvky pro webgame**

Pro vznikající webgame nakreslete některé objekty používané ve hře. Velikost objektů bude 500px ve větším rozměru. (500x500 nebo 300x500 apod.) Vybrat si můžete jeden nebo více objektů z následující nabídky:

- Slunce - animovaný gif
- Vesmírná loď typ standart + ikona/piktogram
- Vesmírná loď typ „na nebezpečný náklad“ + ikona/piktogram
- Vesmírná loď typ express + ikona/piktogram
- Vesmírná loď nákladní + ikona/piktogram
- Zničená nákladní
- Zničená loď guard
- Vesmírná loď typu guard + ikona/piktogram

- Obyčejný obchodní balík
- Zásilka pro expresní doručení
- Objemná nákladní zásilka
- Nebezpečný náklad
- Vesmírná loď typ pirat + ikona/piktogram
- Červí díra (jump gate)
- Souhrnná mapa slunečních soustav
- Logo hry
- Layout stránek – rozvržení stránky bude dodáno.
- Ikona vztahů s piráty (tři stavy – přátelství, neutralita, nepřátelství)

Všechny obrázky musí být vhodné pro web – předpokládá se formát png s průhledným pozadím (černý alfa kanál) a vhodný výběr barev (barvy bezpečné pro web).

Termín odevzdání bude dohodnut se zájemci individuálně. Další informace mailem na [neudez1@students.zcu.cz](mailto:neudez1@students.zcu.cz).

## F Poznámky testerů

Tato příloha předkládá heslovité poznámky testerů, které nebyly doposud zpracovány. Největší podíl na těchto poznámkách má konzultant práce Ing. Petr Vaněček, Ph.D. Hra by dle těchto poznámek měla dále umožňovat:

- rezervování zásilky na planetě pro vesmírnou loď, která na tuto planetu právě letí.
- měnit trasu letu po odstartování lodí
- odložit start na výhodnější dobu s rizikem poklesu výtěžku za doručení
- systém rezervace zásilek s rezervačním poplatkem
- plánování delšího letového plánu pro nákladní loď než jen mezi dvěma planetami)
- zobrazení časových statistik vlastní společnosti (historie přepravy, příjmů, výdajů apod.)
- zobrazit proklikávací seznam vlastních lodí a doků
- prodávat již zakoupené lodě
- použití informačního systému, který sbírá informace o zakázkách z různých planet

Mezi nové herní vlastnosti by mohlo patřit zavedení:

- ekonomického modelu firem – režijní náklady
- akciového trhu – hráči mohou obchodovat s akciemi svých společností do výše 49%
- výroby produktů na planetách – některé suroviny se nacházejí jen na některých planetách, na jiných probíhá výroba mezi produktů, z těchto meziproductů se vyrábějí další produkty; převoz a prodej těchto výrobků a meziproductů

Pro uživení hry a zpříjemnění ovládání by bylo možné:

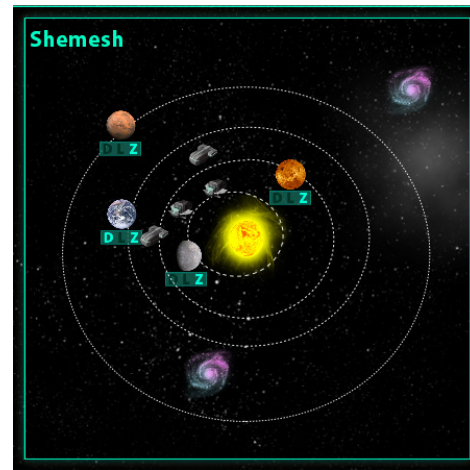
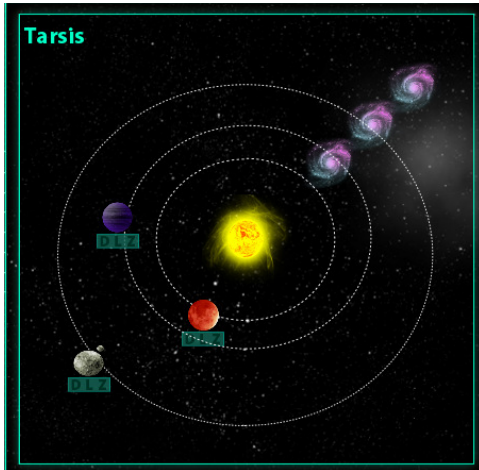
- Ozvučit hru
- Umožnit uživateli zobrazit názvy planet a cíl červích děr se seznamem planet, které jsou za nimi





## G Tutoriál ke hře

Ve hře začínáš jako majitel společnosti doručující zásilky mezi různými planetami v galaxii. Máš pouze jednu budovu a vlastníš jednu malou loď. Tvým cílem je budovat a rozvíjet společnost aby byla co nejvíce prosperující.



Na každé planetě nalezneš množství zásilek, které čekají na doručení. Ne všechny ale můžeš přepravit hned. Některé vyžadují speciální druhy lodí nebo jsou příliš velké na to abys je mohl do své lodi naložit.

Kochav Chama					
Zboží	Tun	Odměna	Cloná planeta	Vzdálenost	Zobraz lodě
Jaderné palivo	9	\$86043	Ma adim	Lokální	Vyber
Jaderné palivo	4	\$15963	Nogah	Lokální	Vyber
Drahé kovy	10	\$5736	Ma adim	Lokální	Vyber
Tkané látky	9	\$5623	Eretz	Lokální	Vyber
Tkané látky	1	\$807	Chiv	Mimo systém	Vyber

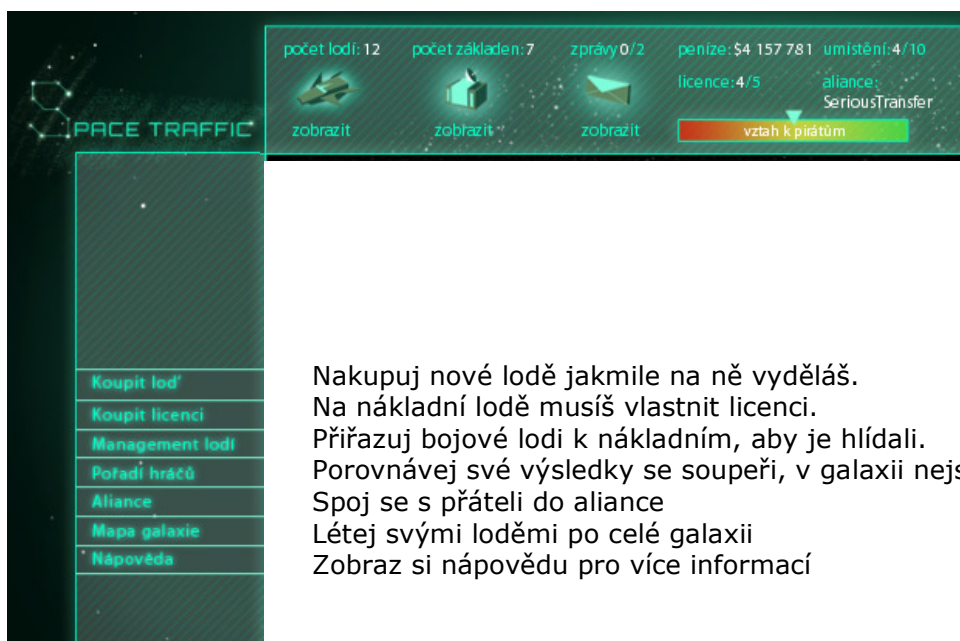
Každý tvůj let s sebou nese také náklady. Stejně tak při přistávání zaplatíš jistý poplatek. Buduj co nejvíce doků a plánuj lety abys zaplatil co nejméně.

```
18. 5. 2010 - 13:10:56 - Lod' Frighter01 zaplatila za svůj let $3950.  
18. 5. 2010 - 13:10:56 - Lod' Frighter01 přistála na planetě Jebusej za poplatek $2500.  
18. 5. 2010 - 13:10:18 - Lod' Frighter01 poslána na planetu Jebusej.  
18. 5. 2010 - 13:10:05 - Lod' Frighter01 proletěla k červí dírou do soustavy Misrajim.  
Zaplaceno clo $0.  
18. 5. 2010 - 13:10:05 - Lod' Frighter01 zaplatila za svůj let $750.  
Zobrazit vše
```

Jakmile vyděláš na nákladní loď, budeš moci doručovat i zásilky do jiných slunečních soustav. Do těch můžeš proletět červí dírou. Loď můžeš i vylepšovat a díky tomu můžeš nakládat větší zásilky. Čím lepší loď chceš mít, tím více bojových lodí potřebuješ aby ji hlídali. Můžeš se totiž setkat s piráty, kteří se tě pokusí zničit.



Ve hře se orientuj pomocí horního a bočního menu:



The screenshot shows the game interface for 'PACE TRAFFIC'. At the top, there is a status bar with the following information: počet lodí: 12, počet základen: 7, zprávy 0/2, peníze: \$4 157 781, umístění: 4/10, licence: 4/5, and alliance: SeriousTransfer. Below this, there are three 'zobrazit' buttons with icons of a ship, a base, and an envelope. A progress bar for 'vztah k pirátům' is also visible. On the left, a sidebar menu contains the following options: Koupit loď, Koupit licenci, Management lodí, Pořadí hráčů, Alliance, Mapa galaxie, and Náповěda.

Nakupuj nové lodě jakmile na ně vyděláš.  
Na nákladní lodě musíš vlastnit licenci.  
Přiřazuj bojové lodě k nákladním, aby je hlídali.  
Porovnávej své výsledky se soupeři, v galaxii nejsi sám.  
Spoj se s přáteli do aliance  
Létej svými loděmi po celé galaxii  
Zobraz si nápovědu pro více informací

Toto jsou všechny základní informace abys mohl začít hrát. Zaregistruj se, buď nejlepší a vyhráj některou z cen!

