

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta



Návrh herní aplikace pro zjišťování osobnostních profilů uživatelů

Bakalářská práce

Jan Schupke

Vedoucí práce: Ing. Jiří Jelínek, CSc.

České Budějovice 2016

ZADÁVACÍ PROTOKOL BAKALÁŘSKÉ PRÁCE

Student: Jan Schupke
(jméno, příjmení, tituly)

Obor – zaměření studia: Aplikovaná informatika
Katedra: Ústav aplikované informatiky

Školitel: Ing. Jiří Jelínek, CSc.
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Garant z PřF: dtto
(jméno, příjmení, tituly, katedra – jen v případě externího školitele)

Školitel – specialista, konzultant:
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

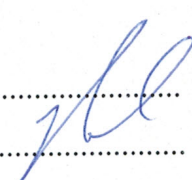
Téma bakalářské práce: Návrh herní aplikace pro zjišťování osobnostních profilů uživatelů

Cíle práce:

1. Shromáždění informací, příkladů a materiálů týkajících se užití počítačových her pro analýzu chování uživatelů a uživatelského osobnostního profilu.
2. Návrh herní aplikace v jazyce Java pro zjišťování uživatelských osobnostních charakteristik s použitím UML pro návrh SW. Návrh základní metodiky zpracování získaných dat.
3. Realizace aplikace v rozsahu konzultovaném s vedoucím práce podle zásad vývoje SW produktů.
4. Provedení praktických testů herní aplikace a příkladu analýzy získaných dat.
Vyhodnocení testů.

Základní doporučená literatura:

- Davidson, Andrew. *Killer game programming in Java*. ISBN 0-596-00730-2
- Blatný, Marek, kol. *Psychologie osobnosti : hlavní témata, současné přístupy*. ISBN 978-80-247-3434-7
- Smékal, Vladimír. *Pozvání do psychologie osobnosti : člověk v zrcadle vědomí a jednání*. ISBN 80-86598-65-9
- Aktuální materiály a články z Internetu a digitálních knihoven ACM a INSTICC.

Financování práce :.....
Vedoucí práce :.....podpis : 

U externích vedoucích fakultní garant práce.....podpis :

Garant oboru bak.. studia (nepožaduje se u zaměření „příprava na mag. studium biologie)

..... podpis :

Vedoucí katedry (V. Z. JAN FESL) podpis *Jan Fesl*

Případný souhlas vedoucího ústavu AV podpis :

V Českých Budějovicích dne 21.5.2014

Převzal/a dne..... 27. 05. 2014 podpis : *Schupke*

Bibliografické údaje

Schupke J., 2016: Návrh herní aplikace pro zjišťování osobnostních profilů uživatelů. [Design of a Game Application for Profiling Personality Characteristics of Its Users. Bc. Thesis, in Czech.] – 65 p., Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Tématem práce je návrh a implementace počítačové hry, schopné profilovat osobnost hráčů na základě jejich herních akcí. První část práce je věnovaná teorii osobnosti a způsobu profilování obecně. Následuje specifikace požadavků na software a jeho návrh. V závěru je přiložena ukázka výsledků fungování aplikace. Vzhledem k rozsahu tématu je výsledek ve fázi konceptu, a je předpokládán jeho další zpracování.

Summary

The topic of this thesis is design and implementation of a video game, capable of profiling its players based on their in-game actions. The first part of the thesis is focused on the theory of personality, and methods of its profiling in general. The following chapters focus on the specification of the software requirements and its design. At the end, there is an example of results, demonstrating the application's functionality. Given the vastness of the topic, the research is currently in its conceptual state, and future development is expected.

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně, pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce a implementovaného software. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V dne

Podpis autora

Poděkování

Rád bych poděkoval Ing. Jiřímu Jelínkovi, CSc. za vedení této práce,
a prof. PaedDr. Ivě Stuchlíkové, CSc. za cennou konzultaci ohledně psychologie
osobnosti.

Obsah

1 Úvod	1
1.1 Cíle práce	1
1.2 Existující řešení	1
1.3 Praktické využití	2
1.3.1 Využití v rámci hry	2
1.3.2 Využití mimo hru	2
1.3.3 Kombinace obou předešlých	2
1.4 Struktura práce	3
2 OCEAN model	4
2.1 Popis osobnostních charakteristik	4
2.2 Alternativní modely	4
2.3 Způsob měření	5
2.3.1 Soubory měřících otázek	5
2.3.2 Vybraný soubor	6
2.4 Kritika	6
3 Požadovaná funkcionalita	7
3.1 Popis hry	7
3.2 Požadavky na hru	7
3.2.1 Open world	7
3.2.2 Dialog	7
3.2.3 Questy	8
3.2.4 Nelineárnost	8
3.2.5 Single player	8
3.2.6 Modulárnost a extensibilita	8
3.2.7 Logování herních akcí	9
3.3 Grafické a zvukové assety	9
3.4 Metodika zpracování dat	9
3.5 Požadavky na analyzér	10
4 Implementace řešení	11
4.1 Technologický stack	11
4.1.1 Java	11
4.1.2 LibGDX	11

4.1.3	Box2D	11
4.1.4	Tiled	11
4.1.5	JUnit	12
4.1.6	Git	12
4.1.7	Gradle	12
4.1.8	Apache Maven	12
4.2	Metodika vývoje	12
4.3	Struktura projektu podle LibGDX	13
4.4	Logické moduly projektu	14
4.4.1	Base	14
4.4.2	Content	21
4.4.3	Logging	21
4.5	Management získaných dat	21
4.6	Game design a development	22
4.7	Technický design analyzáru	22
4.7.1	Konfigurační rozhraní	23
4.7.2	Produkční rozhraní	24
5	Testování	26
5.1	Testování technické funkcionality	26
5.1.1	Unit testing	26
5.1.2	Scenatio testing	26
5.2	Testování osobnostního profilování	27
5.2.1	Skupiny typů událostí	28
5.2.2	Závěr osobnostního profilování	30
6	Možnosti rozšíření	31
6.1	Návrh frameworku hry	31
6.1.1	Přidání mechanik	31
6.2	Produkční kvalita	31
6.3	Komplexnější analýza dat	32
6.4	Postup přidání obsahu do hry	32
7	Závěr	34
	Seznam použité literatury	35
	Seznam tabulek	38

Seznam obrázků	39
Seznam použitých zkratk	40
A Manuál hry	41
B Manuál analyzéro	42
C Event model	43
D BFI model	44
E Herní log	45
F Testovní scéráře	46
G Případy užití analyzéro	53
H Obsah DVD	56

1. Úvod

1.1 Cíle práce

Tato práce si klade za cíl popsat, navrhnout a implementovat jednoduše rozšířitelnou počítačovou hru, kterou je možno využít pro zjištění osobnostních charakteristik hráčů na základě jejich rozhodování a akcí v rámci hry. Na této hře jsou ukázány minimální požadavky, které musí každá hra splňovat, aby pomoci ní bylo možno osobnostní charakteristiky profilovat.

Vzhledem k rozsahu problematiky je kladen důraz zejména na demonstraci konceptu na úkor obsahu hry takovým způsobem, aby bylo možno hru jednoduše rozšířit v případě potřeby.

Logika pro sběr relevantních dat pro profilování osobnosti je zapouzdřená a oddělená od samotné hry, aby ji bylo případně možné použít i v jiné hře bez nutnosti velké modifikace implementace.

Pro testování funkčnosti a zpracování získaných dat je implementovaná samostatná aplikace.

Narozdíl od řešení zmíněných v sekci 1.2 je zde kladen důraz na samotnou extrakci charakteristik pomocí analýzy chování hráče, ne na implementaci realtime profilování a dynamického generování obsahu na základě výsledků profilování přímo ve hře. V tomto ohledu lze na cíl práce nahlížet jako na snahu o poskytnutí alternativního média ke klasickým dotazníkům, což je podrobněji vysvětleno v následujících kapitolách.

1.2 Existující řešení

Studium osobnosti a počítačových her se často zaměřuje na volbu žánru hry v závislosti na osobnosti hráče, např. [1]. Co se týká samotných her, které by umožňovaly profilování, za zmínku stojí např. studie [2], která (1) definuje cíle profilování (poskytnout schopnější AI, personalizovat obsah) a způsoby jejich získání (profilování hráče, modelování jeho akcí, taktik a strategií), a (2) zmiňuje některé konkrétní případy existujících her (Civilization, World of Warcraft, apod.), které využívají určitý aspekt profilování hráče na základě jeho chování.

Dále např. výzkum [3] demonstruje zkoumání chování hráče v existující hře, upravené za účelem zjištění B5 (*Big Five* model, popsáný v kapitole 2). Oproti zmíněnému výzkumu se tato práce snaží implementovat vlastní hru, která je od

základu navržená pro zjišťování osobnostních charakteristik v relativně krátkém čase.

Existují také například hry typu [4], které ale pouze prezentují klasické dotazníkové otázky interaktivní formou, proto nejsou pro účely této práce relevantní.

1.3 Praktické využití

Osobnostní profilování je tradičně prováděno formou dotazníku [5]. Tato práce by měla poskytnout alternativní přístup k získání těchto dat.

Důvodů pro získávání osobnostních profilů je několik. Dají se rozdělit do dvou skupin: *informace využitelné v rámci hry*, např. [6][2], a *informace využitelné mimo samotnou hru*, např. [7][8].

1.3.1 Využití v rámci hry

Ukazuje se, že podskupiny jednotlivých osobnostních charakteristik lze přímo aplikovat na konkrétní herní aspekty, což je využitelné při návrhu hry z hlediska zvýšení zájmu (nebo udržení pozornosti) hráče [9]. Pokud hra implementuje nástroj pro tvorbu osobnostního profilu, může získaná data interně využít pro generování obsahu určeného pro konkrétního hráče (adaptabilní obtížnost hry, preference určitého druhu obsahu, apod.) [6][2].

1.3.2 Využití mimo hru

Osobnost hraje roli i při volbě povolání, případně role v pracovní skupině. Profilování zaměstnanců ze strany managementu organizace může vést ke zvýšení pracovní efektivity skupiny a zlepšit pracovní vztahy [7][8].

Takové profilování je možno využít i při pracovních pohovorech s potenciálním zaměstnancem. Je na zvážení, zda je formát hry pro tuto konkrétní situaci vhodný, nicméně příklad obecně demonstruje možný způsob využití.

1.3.3 Kombinace obou předešlých

Zmíněné je možno zkombinovat v souvislosti s napojením konkrétní hry na rozsáhlejší systém (např. sociální síť), který může v závislosti na profilu uživatele cíleně doporučit obsah.

1.4 Struktura práce

Práce se skládá z teoretické části, která se zabývá problematikou měření charakteristik, a implementační části, jejíž výstupem bude funkční hra, demonstrující v omezené míře vztah mezi herními akcemi a osobnostními charakteristikami. Omezenou mírou se zde myslí výpuštění některých charakteristik z analýzy, případně nezohlednění některých dílčích podcharakteristik z důvodu velkého rozsahu tématu, které zahrnuje několik disciplín (psychologie osobnosti, software engineering, game design), a omezeného časového rozsahu pro tvorbu práce. Proto je práce koncipovaná jako návrh, na kterém bude možné v budoucnu dále stavět.

Do implementační části patří i vývoj analyzáru, což je jednoúčelová samostatná aplikace, jejímž úkolem bude zpracování a interpretace zíschaných dat.

Na konci práce, v kapitole 5, jsou zmíněny konkrétní výstupy ze hry a jejich interpretace.

2. OCEAN model

Big Five, neboli *Five Factor Model* (FFM), je soubor pěti osobnostních charakteristik, kterými je možno obecně popsat osobnost [5] [10]. Je považovaný za zatím nejlepší dostupnou reprezentaci osobnosti [11]. V některých zdrojích se rozlišuje mezi B5 a FFM, nicméně z hlediska výsledků, které oba modely přinášejí, jsou rozdíly hlavně v použitém názvosloví charakteristik a je možné je ignorovat [12] [13]. V této práci oba názvy znamenají totéž.

Jedná se o velmi obecný model, což je nutné pro zachování jeho univerzality napříč jazyky a kulturami [5] [10]. Každá osobnostní charakteristika v sobě zahrnuje množinu konkrétnějších podcharakteristik.

Akronym OCEAN je tvořen počátečními písmeny měřených charakteristik (*Openness to experience, Conscientiousness, Extraversion, Agreeableness, Neuroticism*) [11]. Názvosloví původního výzkumu se mírně liší (*Emotional stability* místo *Neuroticism, Culture* místo *Openness*) [5], nicméně se z hlediska výsledků jedná o totéž.

2.1 Popis osobnostních charakteristik

Podle [11] [9] je možno popsat jednotlivé charakteristiky následovně:

- *Otevřenost* — *Openness to experience* vyjadřuje tendenci zkoušet nové a nekonvenční věci, umělecké cítění. Opakem je konzervativnost.
- *Svědomitost* — *Conscientiousness* vyjadřuje tendenci plnit svěřené úkoly systematicky do konce, míru zodpovědnosti. Opakem je nezodpovědnost.
- *Extraverze* — *Extraversion* vyjadřuje přátelskost a míru vyhledávání sociálních kontaktů. Opakem je introverze.
- *Přívětivost* — *Agreeableness* je schopnost dohodnout se, ochota spolupracovat. Opakem je skepticismus, podezřívavost apod.
- *Neuroticismus* — *Neuroticism* je míra prožívání negativních emocí, nervozita, pesimistický přístup. Opakem je psychická vyrovnanost.

2.2 Alternativní modely

Množina osobnostních charakteristik se v průběhu vývoje modelu měnila. Z původních tří charakteristik, přes 7, až 16 [5] [11]. Stejně tak existuje více jmen pro

jednotlivé charakteristiky, což je vidět například na původních názvech, které se mírně lišily od stávajících [5]. Model *Big Five* se ukázal jako nejvhodnější zčásti proto, že těchto 5 charakteristik je možno přeložit z původní angličtiny do jiných jazyků při zachování konzistentních výsledků [11]. V této práci bude stavěno na využití FFM.

2.3 Způsob měření

Osobnostní charakteristiky není možno měřit přímo, ale je nutno využít metod sebehodnocení, a hodnocení osoby ostatními (*peer assessment*). Proto se obvykle používají dotazníky [11]. Prvotní měření bývalo prováděno pomocí analýzy adjektiv, kterými se tázaný charakterizoval [5], což ale přináší nebezpečí různých interpretací smyslu těchto adjektiv při hodnocení, případně během překladu adjektiv do jiných jazyků. Proto je vhodnější použít fráze, pomocí kterých je možno sdělit myšlenku, a význam se tak jednoduše nezkrslí [11]. Dalším argumentem pro užití popisných frází místo samostatných adjektiv je to, že ne každá část osobnosti je popsatelná jedním adjektivem přirozeného jazyka, resp. shodná napříč jazyky [11].

2.3.1 Soubory měřících otázek

Existuje mnoho měřících nástrojů a nemá smysl zde zmiňovat všechny. Projekt IPIP [14] se snaží o jejich rozvoj. Zde jsou uvedeny jen některé, historicky významné, metriky:

- *NEO-PI-R — Personality Inventory Revised* — je aktualizovaná verze podrobného inventáře 240 otázek [11]. Tento soubor každou charakteristiku ještě dělí na 6 skupin, čímž dosahuje vyšší vypovídající hodnoty. Jedná se o komerční produkt.
- *BFI — Big Five Inventory* — je soubor 44 frází, které tvoří položky dotazníku, na které tázaný odpovídá formou volby, do jaké míry s danou frází souhlasí. Každá z nich se váže na určitou charakteristiku a modifikuje ji s určitou vahou (podle zvolené odpovědi). I přes relativně nižší počet otázek mají výsledky získané pomocí BFI velmi vysokou korelaci jak s výsledky NEO testů, tak i při porovnání s *peer assessmentem* [5].

- Existují i jiné formy, například 10ti otázkový zkrácený BFI dotazník, vhodný pro situace s velmi omezeným časem. V takovém případě je ale nutné počítat s úměrnou ztrátou přesnosti.

2.3.2 Vybraný soubor

V této práci bude využit soubor BFI. Bude snaha inspirovat se těmito frázemi při tvorbě herních situací. Reakce hráčů na tyto situace budou sloužit jako zdroj dat pro pozdější analýzu. Výhoda BFI je nejen v jeho jednodušší implementaci z hlediska objemu dat, ale i proto, že je to volně dostupný soubor, narozdíl od komerčního NEO [15] [16].

Metodika zpracování naměřených dat je popsána v sekci 3.4.

2.4 Kritika

Big Five model je zaměřený na testování osobností v rámci moderní civilizace. Mohou se objevit specifické případy, kdy tento model nelze spolehlivě aplikovat pro velké civilizační rozdíly [17], nicméně je dokázané, že kultura a jazyk nemají na výsledek vliv [11]. Proto je možné říct, že model měření obecně funguje.

3. Požadovaná funkcionalita

Tato kapitola si klade za cíl popsat výslednou hru po uživatelské stránce, specifikovat funkcionalitu potřebnou k dosažení cílů práce, a rozhodnutí zdůvodnit. V této kapitole je i naznačený obecný záměr analýzy chování hráče ve hře.

3.1 Popis hry

Vzhledem k potřebě profilovat osobnost hráče je zvolen žánr *roleplaying game*, který umožňuje hráči vžít se do určité míry do hrané postavy, volnost pohybu v herním světě, a sledování rozhodnutí hráče.

Pro uspokojení požadavku poskytnout alternativní nástroj pro zjišťování osobnostního profilu k jiným metodám (dotazníky) je nutné umožnit dosažení cíle hry v relativně krátkém čase (do 20ti minut).

Detailní výčet funkcionalit a jejich návrh je popsán v kapitole 4. Zde bude zmíněna předpokládaná relevance aspektů hry ke konkrétním osobnostním charakteristikám.

3.2 Požadavky na hru

Následující text popisuje požadavky na jednotlivé části hry. Je formulovaný v rámci možností obecně, aby specifikoval hlavní myšlenky bez implementačních detailů.

3.2.1 Open world

Možnost volného pohybu hráče je důležitá pro monitorování jeho tendence prozkoumávat oblasti mimo hlavní dějovou linii, popřípadě ochotu odbočit z této linie na žádost NPC (*Non-Player Character*) za účelem vykonání nepovinné činnosti.

Je nad rámec práce designovat komplexní a zcela otevřený svět, nicméně je kladen důraz na umožnění co největší volnosti jak pohybu, tak rozhodnutí hráče.

3.2.2 Dialog

Dialogem se do určité míry dá přímo zamaskovat otázka klasického osobnostního dotazníku. Interakce s postavami umožňuje poskytnout hráči výběr z množiny konkrétních rozhodnutí, mezi kterými zvolí.

Na dialogu a volbě reakcí hráče na situace je postavená velká část hry.

3.2.3 Questy

Hráč se v průběhu hry setkává s úkoly, které může splnit, nebo je může ignorovat. Questy mohou být řazeny do linie, kdy splnění jednoho zpřístupní další. Každý quest se navíc skládá z jednoho nebo několika podúkolů (tasků). Charakter questů byl volen tak, aby pokryl co největší spektrum obosnostních charakteristik. Např.:

1. Předání předmětu postavě i přesto, že na hráče nereaguje pozitivně.
2. Konfliktní úkoly, kdy si hráč může vybrat mezi agresivním a diplomatickým rozhodnutím.
3. Questy, které pro splnění vyžadují interakci s postavami, a jejichž cíl není explicitně definovaný v popisu questu.

3.2.4 Nelineárnost

Nelineárnost a míra prozkoumávání světa poskytují další zdroj dat. Velká část obsahu hry neleží na hlavní linii, proto je pro jeho objevení nutný zájem hráče prozkoumávat danou mapu. Pohyb hráče v rámci mapy je monitorovaný. Stejně tak je svět navržený tak, aby se pro splnění některých zadání hráč musel vrátit do již prozkoumané oblasti, čímž se dá měřit ochota splnit úkoly, případně svědomitost.

Nelineární návrh je prezentovaný i charakterem úkolů, kdy není možné splnit vše, ale hráč je nucený vybrat si z několika konfliktních možností.

3.2.5 Single player

Je žádoucí, aby hra vykazovala konzistentní výsledky, pokud bude opakovaně hraná stejným nebo podobným způsobem. Reakce světa a postav v něm by proto měly fungovat algoritmicky. Toto je jeden z hlavních důvodů, proč je hra navržena striktně jako *single player*.

3.2.6 Modulárnost a extensibilita

Záměrem této práce není implementovat hru bohatou na obsah a příběh. I když jsou tyto aspekty pro RPG důležité, jejich kvalitní realizace v dostupném čase pro implementaci práce je nad rámec možností autora. Jako kompenzaci této možné slabiny je nutné hru od začátku navrhovat tak, aby bylo možné obsah jednoduše upravovat a přidávat bez nutnosti velkého zásahu do jádra hry a její logiky.

Stručný popis rozšíření hry o další obsah je popsán v sekci 6.4.

3.2.7 Logování herních akcí

Každou akci hráče je třeba zaznamenávat pro pozdější analýzu. Tato data je nutné mít možnost ze hry exportovat do souboru. Hru je třeba navrhnout tak, aby toto umožňovala.

3.3 Grafické a zvukové assety

Vzhledem k tomu, že tvorba grafických a zvukových assetů nesouvisí s tématem práce a místy není v rámci schopností autora, budou použity assety třetích stran.

V práci jsou použity výhradně soubory z Public Domain, nebo soubory, ke kterým má autor zakoupenou licenci, proto není legálně nutné uvádět zdroje na každý konkrétní použitý soubor, ale budou uvedeny odkazy na celé knihovny popř. online databáze, ze kterých byly assety pořízeny. Obecně platí, že všechny grafické soubory a zvukové efekty použité v této práci jsou outsourcovány, pokud nebude uvedeno jinak.

Odkazy na použité assety jsou dostupné přímo v menu hry, v sekci *Credits*.

3.4 Metodika zpracování dat

Protože měření osobnosti hrou má odlišný charakter od vyplňování možností v dotazníku, a nedá se přesně namapovat na žádný konkrétní dotazník, byla zvolena následující metodika: fráze z dotazníků budou použity jako inspirace pro návrh herních situací, pomocí kterých bude možno sledovat chování hráče a jeho reakce na tyto situace. Pro testování funkčnosti a efektivity profilovací metody bude provedeno srovnání odpovědí hráče v přiloženém profilovacím dotazníku s herními daty. Pro zpracování dat ze hry bude vytvořena malá samostatná aplikace — analyzér.

Profilovací data ze hry budou přístupná ve formě textového logu všech relevantních akcí hráče, Tento log bude následně v konfigurační fázi analyzovaný a porovnaný s odpověďmi z dotazníku pro vytvoření profilovacího modelu hry. K tomuto výsledku bude nutno použít množinu dvojic logů a dotazníkových odpovědí.

Tento model bude následně možno použít pro vytvoření osobnostního profilu hráče analýzou jeho herního logu, bez další nutnosti používat dotazník.

3.5 Požadavky na analyzér

Aplikace musí umožňovat ovládání pomocí grafického rozhraní, které bude rozdělené na dvě sekce: *konfigurační* a *produkční*.

Konfigurační sekce umožní nahrání zdrojových datasetů hráčů, na základě kterých bude sestavený výpočetní model. V této fázi bude možné každý dataset zobrazit v tabulce. Dataset se skládá z herního logu a CSV souboru se seznamem odpovědí na BFI dotazník. Výstupem konfigurační části bude soubor výpočetního modelu. Součástí této sekce je funkcionality pro kalkulaci FFM z poskytnutých dotazníkových odpovědí hráče podle BFI algoritmu.

Produkční sekce umožní pomocí modelu sestaveného v rámci konfigurace a jednoho herního logu kalkulovat osobnostní profil. Výsledná data budou exportovatelná ven z aplikace ve strojově zpracovatelném formátu.

4. Implementace řešení

V této kapitole je vysvětlen technický postup návrhu a implementace software. Staví na informacích formulovaných v rámci analýzy a převádí obecné požadavky do implementačně závislého návrhu.

4.1 Technologický stack

Při volbě technologií byla zohledněna jejich vhodnost vzhledem k charakteru projektu. V některých ohledech neexistuje objektivně lepší volba. V těchto případech byly technologie voleny podle předešlé zkušenosti autora s danou technologií.

4.1.1 Java

Pro vývoj bude použita Java SE 8 pro její multiplatformnost a předchozí zkušenosti autora. Verze 8 je aktuální dostupná verze Javy v době psaní práce.

4.1.2 LibGDX

LibGDX [18] je framework, který sdružuje spoustu existujících knihoven a umožňuje jejich jednoduché využití v jazyce Java. Frameworků je velké množství. Alternativy pro Javu jsou např. Slick 2D [19], nebo JMonkeyEngine [20]. Slick 2D je v porovnání s LibGDX velmi omezený jak funkcionalitou, tak i dokumentací [21], JMonkey je zase příliš komplexní engine vhodný pro vývoj 3D her. Proto se volba LibGDX jeví jako nejrozumnější.

4.1.3 Box2D

Box2D [22] je dvojrozměrný fyzikální engine, který umožňuje pokročilou simulaci pevných 2D těles a interakcí mezi nimi. I když v projektu nebylo využito komplexních fyzikálních interakcí, samotná detekce kolize a výpočet pohybu zjednodušuje implementaci. Box2D je knihovna napsaná v C++. LibGDX umožňuje její použití pomocí Java wrapperu.

4.1.4 Tiled

Tiled je editor 2D map, který poskytuje grafické rozhraní pro vytvoření herního levelu z nahraných textur. Tyto mapy jsou ve formátu XML, a LibGDX poskytuje

možnost jejich přímého načtení v rámci programu. Další výhodou je možnost přidání objektové vrstvy do mapy, a následné parsování objektů do Box2D těles pro vytvoření kolizní vrstvy. Více o editoru na [23].

4.1.5 JUnit

JUnit je framework pro zjednodušení tvorby jednotkových testů pro Java projekty. LibGDX přímo podporuje jeho využití pro testování částí hry.

4.1.6 Git

Git je systém pro verzování (nejen) zdrojových kódů. Je dostupný na [24].

Bitbucket [25] je online služba pro hosting Git repozitářů. Je zvolena proto, že umožňuje zdarma hostovat soukromé repozitáře a vést u nich jednoduchý issue tracker. Pro složitější projekty, na kterých pracuje více vývojářů, je vhodné použít komplexnější nástroj project managementu (např. Redmine [26]), ale pro účely této práce je tracker na bitbucket.org dostačující.

4.1.7 Gradle

Použití Gradle jako nástroje pro spravování závislostí je zdůvodněno tím, že je to výchozí volba pro LibGDX projekty, a pro jeho změnu nebyl důvod. Více o Gradle na [27].

4.1.8 Apache Maven

Podobně jako Gradle, Maven je nástroj pro správu závislostí a sestavování projektů. Je použitý v projektu analyzáru dat. Více o nástroji na [28].

4.2 Metodika vývoje

Metodika pro tuto práci byla zvolena následovně: prvotní analýza a hrubý design veškerého software je řešen rigorózně za pomoci UML. Účel tohoto postupu je vytvoření konkrétní představy o struktuře a fungování software. Další postup vývoje a změny v návrhu jsou řešeny agilní formou z několika důvodů:

- Z principu rigorózního návrhu vyplývá, že je velmi neefektivní z hlediska přizpůsobení novým požadavkům a změnám v projektu. Dá se předpokládat, že během vývoje tohoto software bude zjištěno mnoho nutných změn, které bude za použití agilního přístupu mnohem jednodušší implementovat.

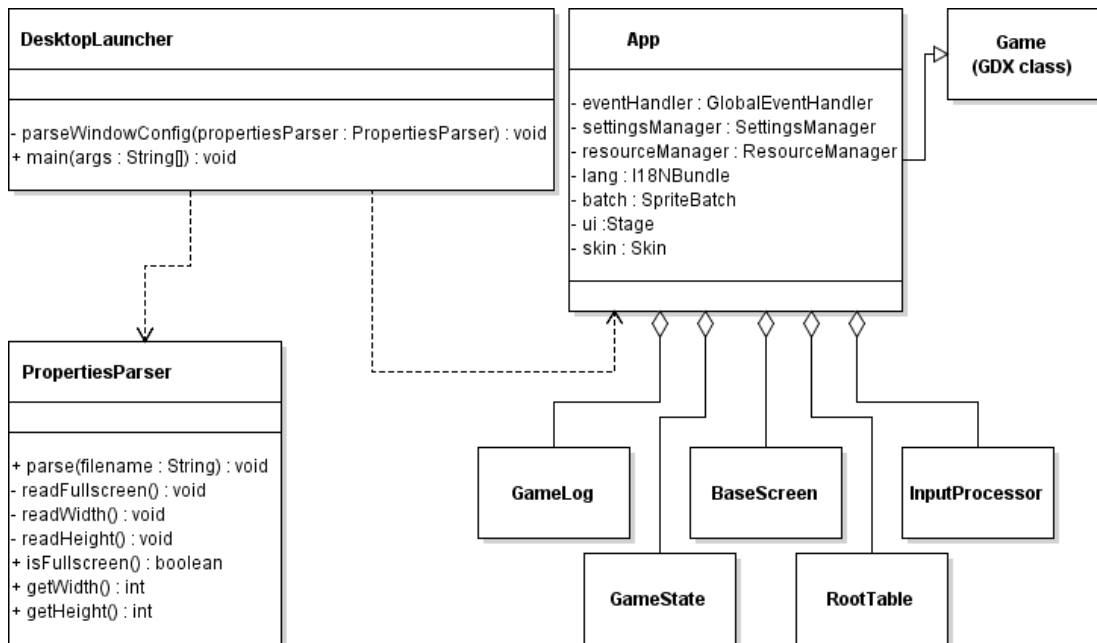
- Design software je zpravidla prováděn zkušeným členem vývojového týmu. Vzhledem k omezené předešlé zkušenosti autora s vývojem her by plný rigorózní návrh v požadované kvalitě nebyl realistický.

Pro efektivní řízení vývoje bude použitý už zmíněný issue tracker dostupný na Bitbucket [25].

4.3 Struktura projektu podle LibGDX

Framework LibGDX má jasně definovanou adresářovou strukturu. Vzhledem k tomu, že automaticky kompiluje projekt pro několik cílových platform (podle konfigurace projektu), veškerý kód hry je umístěný v modulu *core*. Platformně závislé launchery, případně další závislý kód je potom umístěný v modulech podle cílové platformy (*android*, *desktop*, *ios* apod.). Tato práce využívá pouze *core* a *desktop*.

Desktop část obsahuje pouze 2 třídy. *DesktopLauncher* se stará o samotné instanciování a zavedení hry. *PropertiesParser* je utilitní třída, která z konfiguračního souboru čte informaci o velikosti okna. Tuto informaci je potřeba mít k dispozici už při tvorbě instance hry, proto nebylo možné pro jejich načtení použít mechanismy LibGDX, které jsou dostupné až uvnitř kódu hry. *DesktopLauncher* obsahuje metodu `main()`, což z něj dělá vstupní bod aplikace. Grafické znázornění této struktury je vidět na obrázku 4.1.



Obrázek 4.1: Struktura desktop aplikace

4.4 Logické moduly projektu

Následující text popisuje strukturu aplikace uvnitř *core* modulu. Vzhledem k požadavku na jednoduchou extensibilitu, a taky v souladu s obecnými zásadami softwarového inženýrství, je struktura hry rozdělena do tří hlavních částí:

1. *Base* — obecná funkcionalita. Tento balík definuje veškerou herní strukturu i funkční logiku na obecné úrovni.
2. *Content* — konkrétní obsah hry — levely, postavy, předměty apod. Všechny třídy z balíku *content* jsou odvozené z obecných tříd balíku *base*.
3. *Logging* — veškerá logika pro zaznamenávání herních akcí a jejich export pro pozdější analýzu.

Veškeré resource soubory jsou ve složce */core/assets*. Obsah této složky je možné pomocí LibGDX filehandling mechanismu jednoduše načíst.

Všechny message stringy jsou extrahované do properties souboru a načítané do hry pomocí LibGDX mechanismu pro internacionalizaci. Toto umožňuje jednoduché rozšíření aplikace o další jazyky.

4.4.1 Base

Tato část projektu definuje kompletní herní strukturu a logiku na obecné úrovni, a dá se chápat jako framework hry. Cílem je zapouzdřit co nejvíce obecného kódu tak, aby tvorba uživatelského obsahu byla z hlediska kódu co nejstručnější, a aby nevyžadovala programování žádné nízkoúrovňové logiky. Třídy balíku *base* jsou závislé na použitých knihovnách buď jejich využitím, nebo přímou dědičností od těchto tříd. Jedná se zejména o třídy LibGDX, případně LibGDX wrapper třídy nad Box2D. Třídy jsou z velké části abstraktní, a konkrétní třídy hry jsou z nich odvozené. Tímto je docílené oddělení backend logiky hry od herního obsahu.

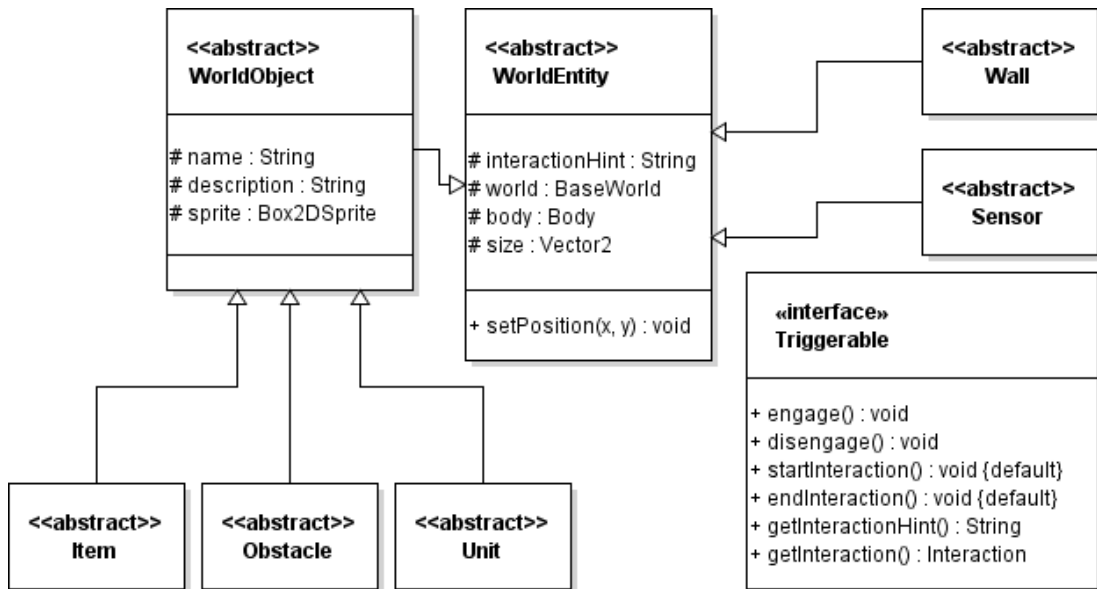
V dalších částech textu se slovy *base* a *framework* myslí tento balík. Na framework LibGDX je odkazováno jeho jménem. V dalších odstavcích následuje popis některých hlavních částí frameworku.

Entity

Balík entit obsahuje třídy, které reprezentují všechny objekty, které se mohou vyskytovat v herním světě. Jedná se jak o jednotky a předměty, tak i o meta entity typu senzory a neviditelné překážky. Každá entita v sobě zapouzdřuje Box2D

těleso, které zpracovává kolize a kontakty. Třídy entit s vizuální reprezentací navíc obsahují instanci `Box2DSprite`, vázanou na těleso.

Součástí tohoto balíku je rozhraní `Triggerable`, které umožňuje zpracovat logiku při kontaktu dvou těles. V praxi se jedná o kontakt hráče s objektem.



Obrázek 4.2: Struktura entit

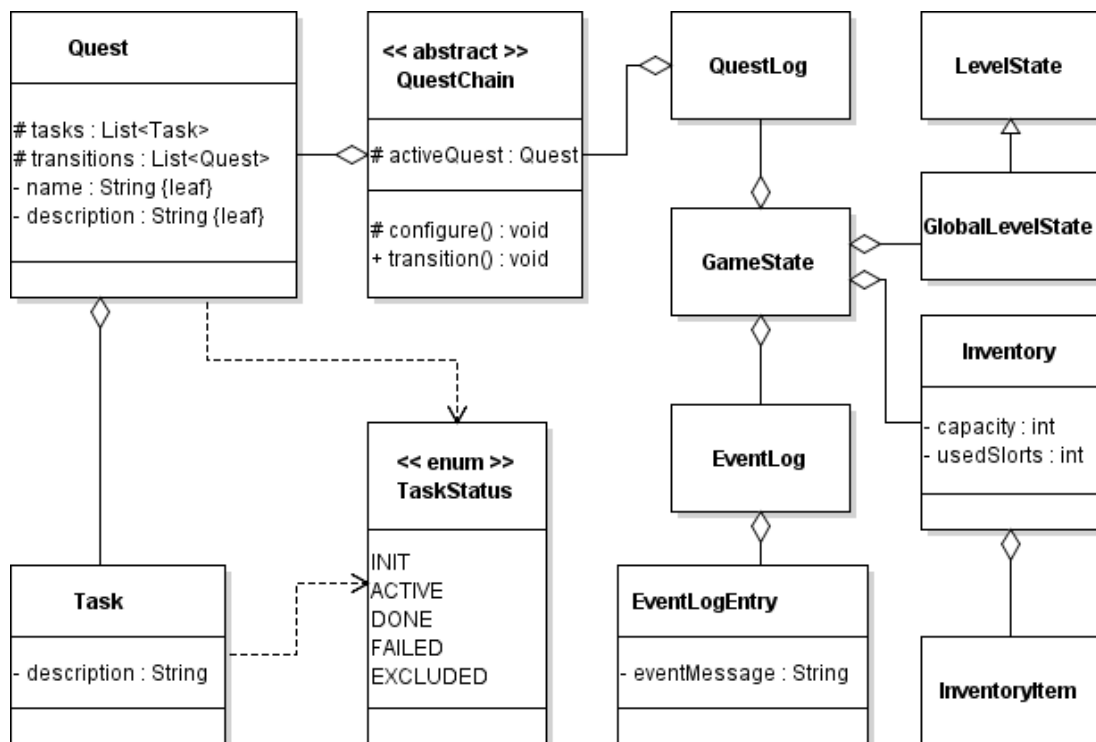
Data Container

Tyto třídy reprezentují logické herní containery, jako quest log, inventář hráče, a log událostí. Jedná se ryze o balík modelu. Struktura datových containerů je znázorněná na obrázku 4.3.

Interaction

Jednou z hlavních funkcionalit hry je interakce hráče s entitami umístěnými ve světě. Technicky jsou řešeny pomocí třídy `Interaction`, která funguje jako container stavů této interakce, a váže se na danou entitu. Stavům se v rámci kódu říká situace, a každý ze stavů má předdefinovanou množinu reakcí, ze kterých hráč volí. Celá interakce se dá chápat jako konečný automat, kde situace reprezentují stavy, a rozhodnutí hráče jsou vstupy, které stav mění. Každé rozhodnutí navíc vyvolává herní událost, která může modifikovat stav herní instance. Vizuálně je situace reprezentovaná popisným textem, možnosti jsou tlačítka, pomocí kterých hráč reaguje na prezentovaný text.

Reakce na rozhodnutí hráče a přechody mezi situacemi řeší interakce ve své metodě `handle()`, která se volá pokaždé, kdy hráč zvolí některou z nabízených



Obrázek 4.3: Datové containery frameworku

možností. Interakce nese informaci o aktuální situaci, a o fallback situaci, která se stane aktuální při přerušení interakce.

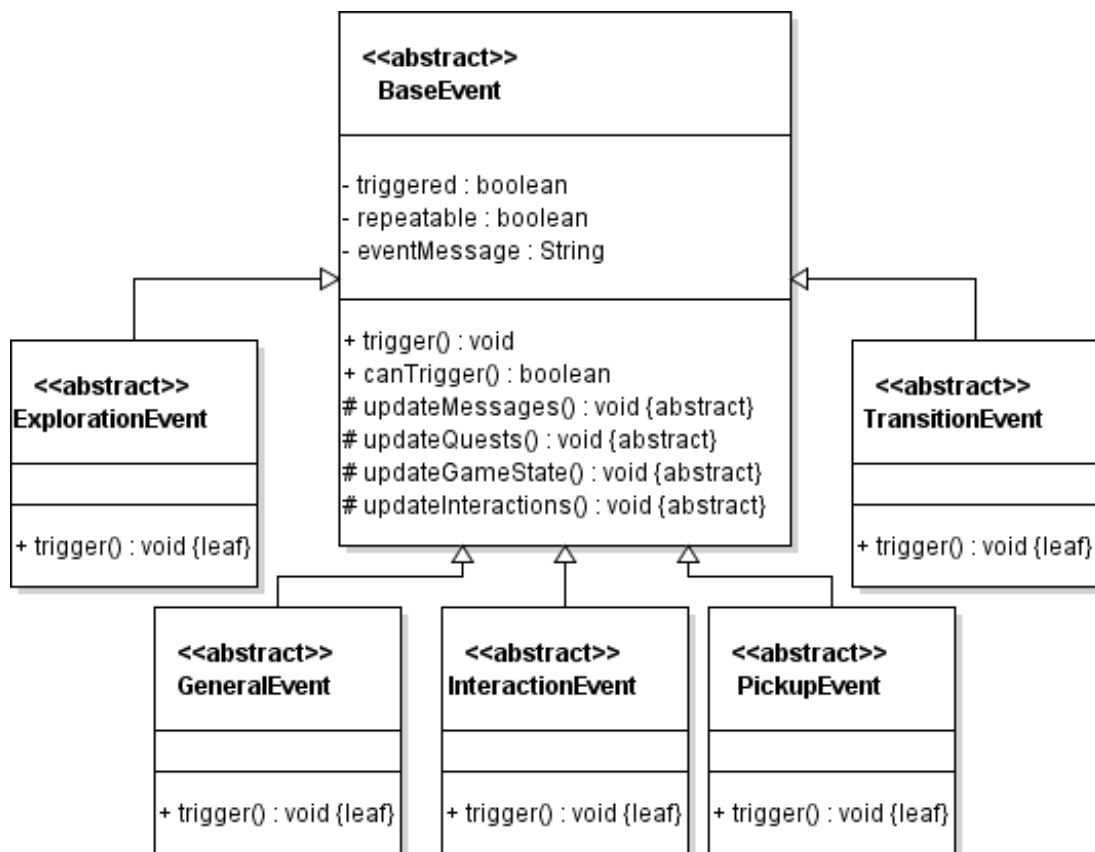
Možnosti reakce hráče mohou být nepřístupné. Ve výchozím stavu jsou povolené všechny možnosti interakce. Jejich zpřístupnění a blokování řeší eventy.

Event

Existují dvě hlavní metody pro zásah do stavu hry. Prvním z nich je uživatelský vstup (stisk tlačítka, pohyb apod.), druhou je kontakt dvou těles. Ke kontaktu dojde vždy v důsledku pohybu hráče.

Objekty eventů jsou zavolány pokaždé, kdy se má vykonat nějaká logika v závislosti na akci hráče. Typicky se jedná o vstup hráče do určité oblasti, nebo interakce s entitou. Eventy obsahují metodu `trigger()`, která je v takovém případě zavolána a vykoná potřebnou logiku. Tato metoda je zároveň jediné místo, ze kterého se logují záznamy pro analýzu dat popsanou v sekci 4.7.

Eventy se dělí do několika skupin podle typu události. Toto rozdělení je znázorněné na obrázku 4.4. Jednotlivé podtřídy ve své `trigger()` metodě obsahují další potřebnou logiku pro daný typ události. Metoda `trigger()` je v těchto podtřídách definována jako `final`, aby bylo zamezeno jejímu překrytí v potomcích, a bylo zajištěno vykonání tohoto kódu. Konkrétní potomci eventů pouze



Obrázek 4.4: Struktura tříd událostí

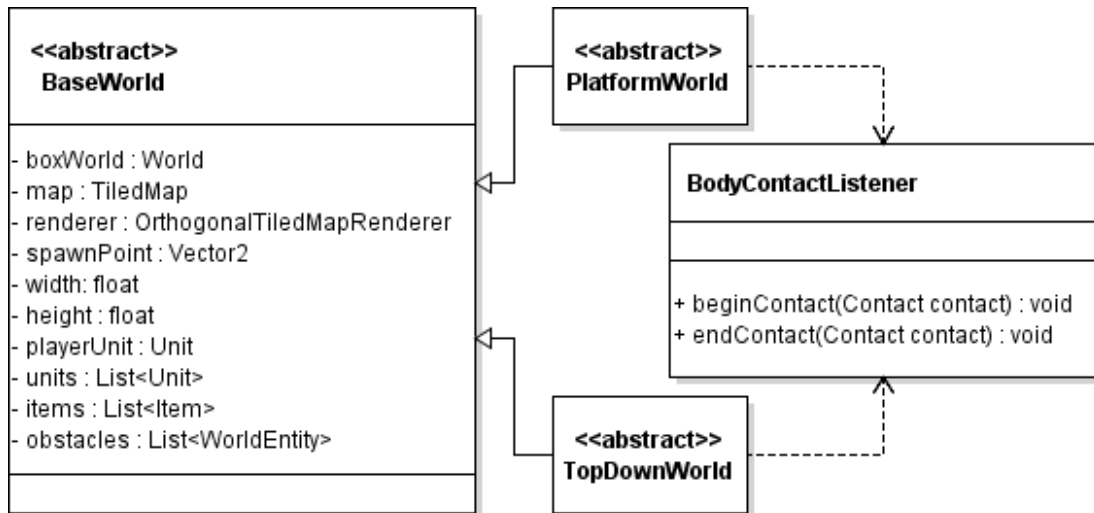
překrývají metody `updateMessages()`, `updateQuests()`, `updateGameState()` a `updateInteractions()`, jejichž vykonání je zajištěno předky v rámci `trigger()` metody.

Každému eventu může být nastavena hodnota možnosti opakování. Pokud event není možné opakovat, jeho metoda `trigger()` vykoná svůj kód pouze při prvním zavolání.

World

Balík *world* obsahuje hierarchii herních světů. Pro zvýšení obecnosti frameworku jsou zde třídy rozdělené na `PlatformWorld` a `TopDownWorld`, aby framework umožňoval použití i platformních levelů. Jediný rozdíl je v konfiguraci gravitace, která je v *TopDown* režimu vypnutá. Herní svět v sobě obsahuje mj. instanci simulace fyzikálního světa pomocí frameworku `Box2D`, tile mapu a renderer mapy. Více informací o principu inicializace světů a map v `LibGDX` je na [29].

Tile mapy obsahují objektovou vrstvu, jejíž objekty jsou při inicializaci světa transformovány na Box2D tělesa, která slouží jako kolizní vrstva terénu. Pro transformaci objektů z mapy na Box2D tělesa byla implementována metoda v rámci třídy `BaseWorld`, zároveň je využito třídy `WorldObjectFactory` pro tvorbu těles.



Obrázek 4.5: Třídní struktura herních světů

Senzory

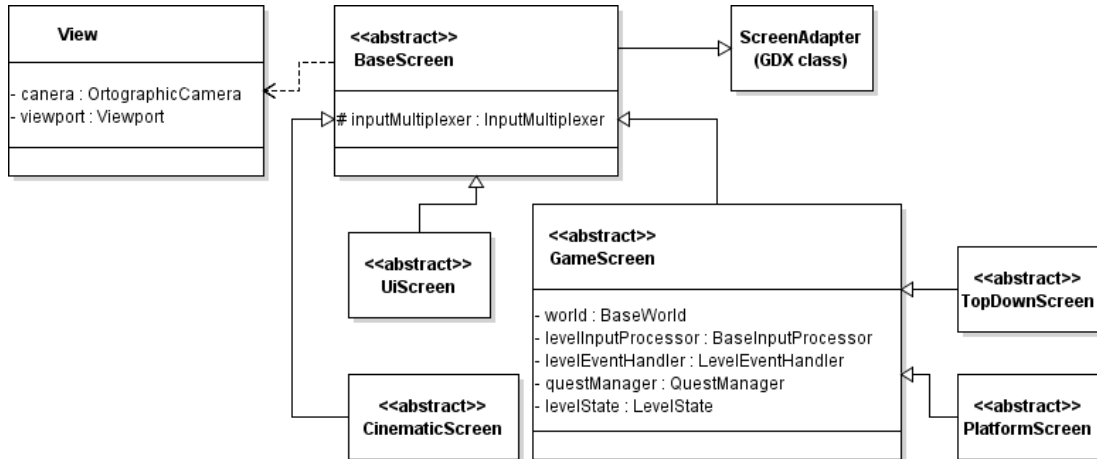
Pro vyvolání události bez explicitní volby hráče je ve hře využito Box2D senzorů, což jsou tělesa umístěná ve světě. Narozdíl od klasických těles nemají grafickou reprezentaci, a nekolidují s ostatními tělesy (např. s hráčem), ale zaznamenávají kontakt s tělesem. Tímto způsobem je zajištěno monitorování pohybu hráče v herním světě. Ukázka umístění senzoru je vidět na obrázku 4.11.

Screen

Screen reprezentuje herní stage. Dá se chápat jako level. Je to hlavní třída každé herní oblasti, která v sobě drží data o daném levelu. Existují specificky třídy pro *TopDown* a *Platform* verzi levelu. Jejich rozdíl je v druhu herního světa a input processoru, které obsahují. Zpracování vstupu je popsáno v sekci 4.4.1.

V implementované hře jsou použity pouze instance *TopDown* obrazovek. *Platform* verze je přiložena v rámci frameworku pro možnost dalšího rozšiřování hry.

Součástí tohoto balíku je i třída `View`, která zapouzdřuje instanci LibGDX tříd `Camera` a `Viewport`, které zjednodušují vykreslování aktuální části světa v závislosti na pozici hráče. `View` zároveň obsahuje logiku pro zamezení kamery překročit hranice herní mapy.



Obrázek 4.6: Struktura typů obrazovek

Input

Balík `input` obsahuje hierarchii tříd, které dědí od třídy `InputAdapter` z frameworku LibGDX, která obstarává zpracování uživatelského vstupu. V projektu je implementovaná možnost mít aktivních více `input processorů` zároveň (pomocí třídy `InputMultiplexer`). Na každou instanci `Screen` je proto aplikovaný `GlobalInputProcessor`, a potom volitelně specifické herní procesory, pokud se jedná o `screen` herního levelu.

Vstupy jsou zpracovávány sekvenčně všemi aktivními procesory, proto je možné implementovat logiku zpracování jedné vstupní události ve více procesorech podle relevance.

Struktura questů

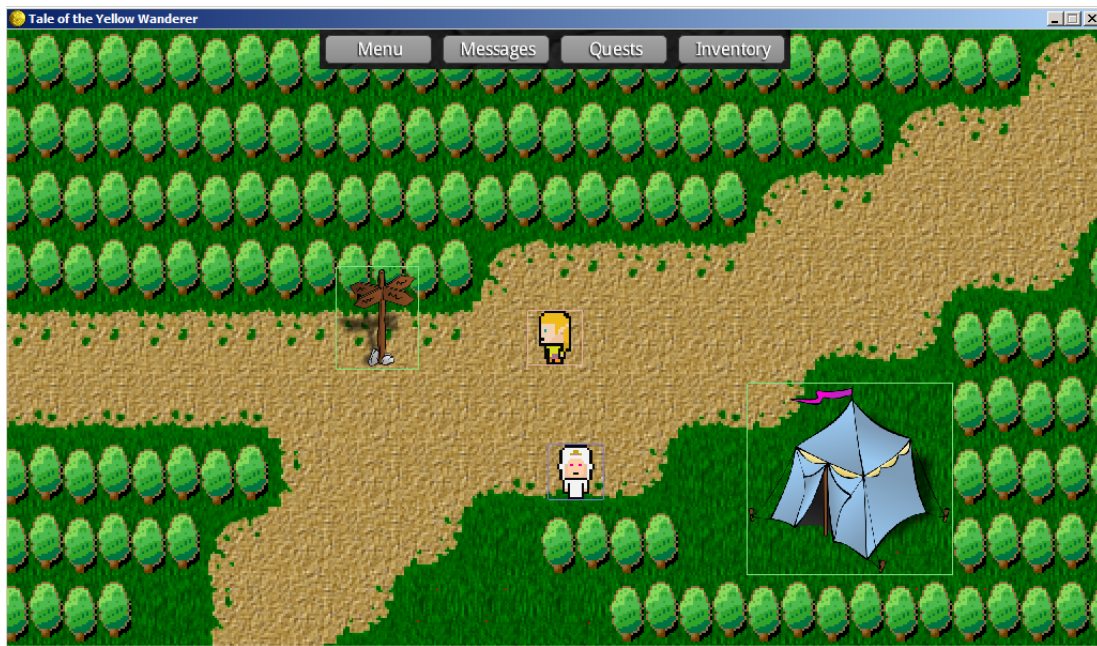
Každý `Quest` je implementovaný jako část sekvence `QuestChain`. V rámci sekvence je vždy aktivní právě jeden `Quest`. Každý `Quest` má 1–n `Tasků`, které hráč postupně plní. Charakter `Tasků` není omezený. Každý `Quest` i `Task` se nacházejí v právě jednom stavu (`TaskStatus`). Tento stav se mění pomocí volání herních událostí.

Debug režim

Pro účely vývoje byl implementovaný přepínač debug renderingu. Pomocí klávesové zkratky F2 je možno zobrazit renderování Box2D těles (normálně se vykresluje jen sprite objektu), a rendering struktury UI (zkratka F1). V konfiguračním souboru je dále možno přepsáním hodnoty `VIEWPORT_MULTIPLIER` resp. `VIEWPORT_WIDE_MULTIPLIER` docílit zoom-out celé hry pro zobrazení větší části mapy, a úpravou hodnoty `DEBUG_OBJECT_POSITION` přemístit všechny relevantní objekty blízko k hráči pro snadnější testování interakčních závislostí.

Dále je možné explicitně vyvolat dump log souboru (klávesou F5), a vygenerovat ručně soubor herního modelu (klávesa F6). V provozním režimu je herní log generovaný na konci každé hry, debug switche jsou zneprístupněné a export game modelu není možný.

Tyto klávesové zkratky nejsou uvedeny v seznamu zkratk v menu hry, protože jsou pro běžného hráče nepřístupné.



Obrázek 4.7: Kolizní debug režim

Shrnutí

Framework obsahuje ještě další třídy, které už nejsou tak zásadní pro fungování aplikace jako výše zmíněné. Celý framework je dokumentovaný pomocí Javadoc, který je dostupný na příloženém DVD.

4.4.2 Content

Content definuje jednotlivé herní levely a vše, co se v nich nachází. Jedná se o fyzické entity (jednotky, předměty apod.), ale i abstraktní entity (questy, eventy), logiku levelů a propojení jednotlivých entit. Každý herní level má následující strukturu:

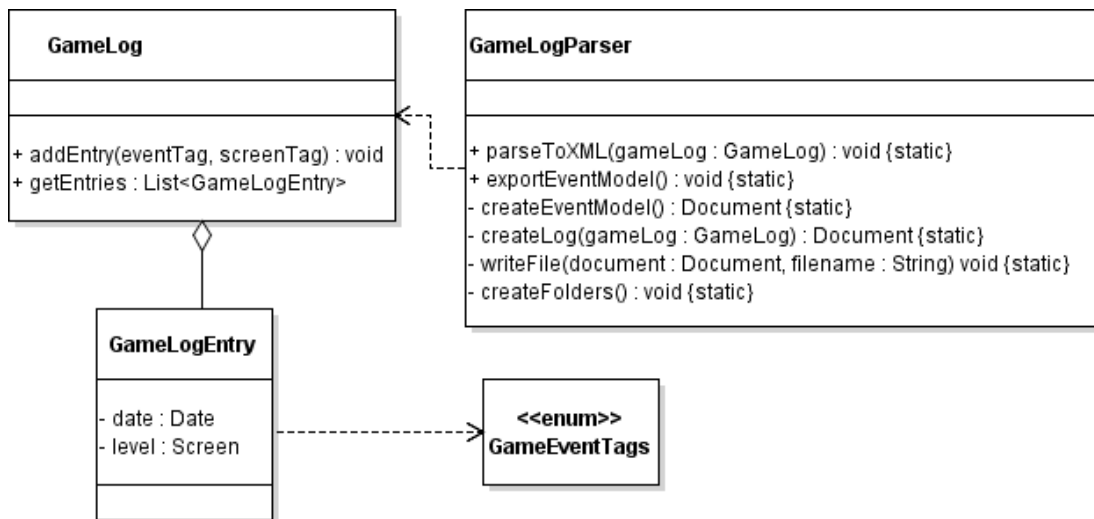
- *Event* — balík všech eventů daného levelu.
- *Item* — balík všech objektů typu `Item` a jejich interakcí.
- *Obstacle* — balík všech objektů typu `Obstacle` a jejich interakcí.
- *Quest* — balík všech questů levelu, a třída `QuestManager`, která se stará o tvorbu a poskytování instancí questů.
- *Sensor* — balík všech senzorů daného levelu.
- *Unit* — balík všech objektů typu `Unit` a jejich interakcí.
- *EventHandler* — třída, která vytváří instance eventů levelu, a poskytuje jejich reference.
- *Screen* — hlavní třída daného levelu.
- *World* — herní svět, který se stará o inicializaci mapy a všech entit levelu.

4.4.3 Logging

Existuje samostatný modul pro třídy, které implementují logiku logování událostí a exportu dat. Každá vyvolaná událost vytváří záznam v logu hry. Tento log je později použitý pro analýzu chování hráče. Způsob využití logovaných dat je popsán v sekci 4.5.

4.5 Management získaných dat

Hra generuje dva typy datových souborů. Prvním z nich je *game event model*, jehož generování je vyvolané ručně v debug módu klávesou F6. S tímto souborem běžný hráč nepřichází do styku a je vygenerovaný předem pro účely analýzy dat. Obsahuje seznam všech možných událostí, které mohou ve hře nastat. Ukázka modelu je vidět v příloze C.



Obrázek 4.8: Struktura logovacích tříd

Druhým souborem je *playthrough log*. Pro každé přehrání hry je vygenerovaný samostatný playthrough soubor, který obsahuje sekvenci všech eventů v takovém pořadí, v jakém je hráč aktivoval. V debug módu je možné vyvolat dump logu explicitně klávesovou zkratkou F5.

4.6 Game design a development

Metodiky game developmentu nesouvisí přímo s tématem práce, proto nebudou detailně vysvětleny. Autor vychází z doporučení a postupů uvedených v oficiální dokumentaci LibGDX [29] a naučných publikacích [30] [31] [32].

4.7 Technický design analyzáru

Cílem práce je zjištění osobnostního profilu hráče na základě jeho akcí v rámci hry. Pro tento účel byl vytvořený analyzáru dat. Analyzáru je malá grafická utilita, jejíž funkcionality se dá rozdělit na dvě části:

1. *Konfigurační režim* — vytvoření profilovacího modelu z poskytnuté množiny vstupních herních dat a odpovědí na BFI dotazník.
2. *Produkční režim* — použití tohoto modelu k následnému získávání osobnostních profilů na základě herních logů hráčů.

Porovnáváním s existujícím dotazníkem pro osobnostní profilování je zajištěno získání dvojice vstupních dat (herní akce hráče), a požadovaného výstupu

(FFM výsledek z dotazníku). Jedná se o optimalizační úlohu, kdy je třeba vstupním událostem ze hry přiřadit vhodné váhy tak, aby se výstup blížil výsledným hodnotám, které poskytují odpovědi dotazníku.

Původním plánem bylo nastavení těchto hodnot ručně. V průběhu implementace ale byla zjištěna řada problémů, které vedly k rozhodnutí zaměřit se na konceptuální popis analýzy, ověření existence korelace mezi některými konkrétními herními akcemi a traity FFM (pro potvrzení funkčnosti), a plnou implementaci společně s rozsáhlejší analýzou pro přesnější výsledky nechat jako námět pro pokračování na této práci.

4.7.1 Konfigurační rozhraní

Konfigurační fáze analyzáru funguje na principu parsování vstupních souborů, které je možné nahrát pomocí grafického rozhraní aplikace. Jedná se o tyto soubory:

1. *Model událostí hry* — XML soubor obsahující seznam všech událostí, kterým během hry může dojít. Obsahuje jejich unikátní identifikátor a textový popis. Tento model je pro usnadnění generovaný hrou, protože ta už implementuje objektovou reprezentaci tohoto modelu. Část modelu je vidět v příloze C.
2. *BFI model* — XML soubor, který reprezentuje výpočetní model FFM profilu na základě odpovědí BFI dotazníku. Tento model byl vytvořený ručně podle existujícího BFI klíče. Ukázka BFI modelu je vidět v příloze D.
3. *Seznam data setů* (každý testující hráč poskytuje tyto 2 soubory):
 - (a) *BFI odpovědi* — CSV soubor se seznamem odpovědí na BFI dotazník.
 - (b) *Herní log* — XML soubor se sekvencí herních událostí, tak jak je hráč vyvolal. Tento log se generuje automaticky po dokončení hry. Ukázka herního logu je vidět v příloze E.

Parsing

Po úspěšném načtení zmíněných souborů se v rámci aplikace vytvoří datové modely pro hru a pro BFI. Těch je využito pro kalkulaci následujících hodnot:

1. *Osobnostní profil* každého hráče na základě jeho dotazníkových odpovědí — tato data tvoří požadovaný výstup optimalizační úlohy popsané v následujícím odstavci.

2. *Souhrn akcí hráče* — je zjištěno, které události hráč vyvolal, a kolikrát. Tato data reprezentují vstup optimalizační úlohy popsané v následujícím odstavci.

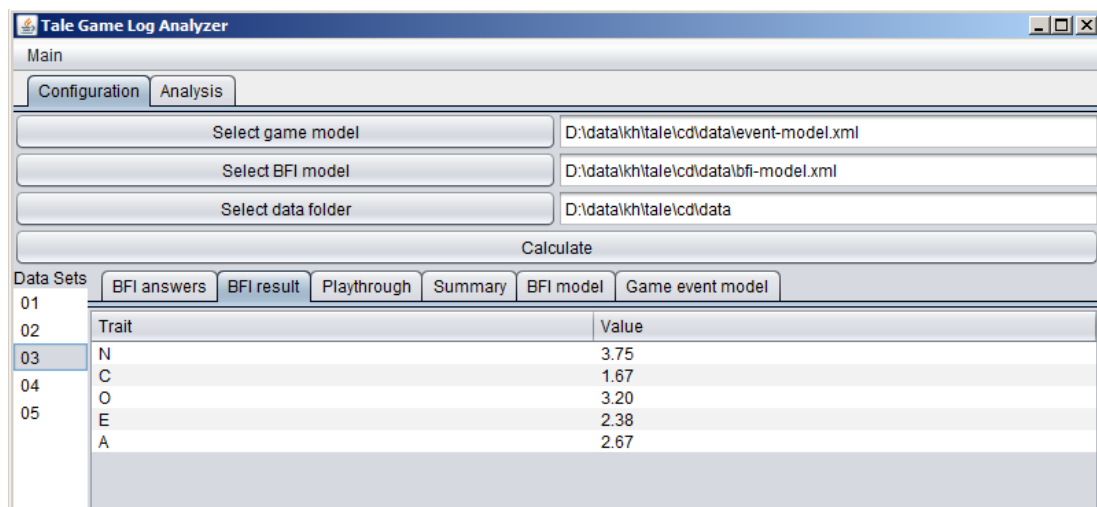
Na konci procesu načtení dat vznikne množina dvojic vstupních dat a požadovaných výstupů. Úkolem analyzáru je zjistit optimální váhy a vliv na jednotlivé osobnostní charakteristiky pro každý event vstupních dat takovým způsobem, aby se výsledek co nejvíce blížil hodnotám BFI dotazníku. Tyto váhy by tvořily výpočetní model pro zjišťování osobnostního profilu na základě samotného herního logu.

Na tuto úlohu by bylo vhodné aplikovat techniku učení modelu (neuronová síť). Tato fáze nebyla z časových důvodů implementovaná. Příklad užití konfiguračního režimu analyzáru je popsán v příloze G.1.

4.7.2 Produkční rozhraní

V tomto rozhraní se momentálně nachází výčet některých logovaných herních akcí, a jejich korelace s určitými charakteristikami osobnosti. Tyto akce a jejich vztah k určité charakteristice byly vybrány manuálně v závislosti na znalosti BFI modelu.

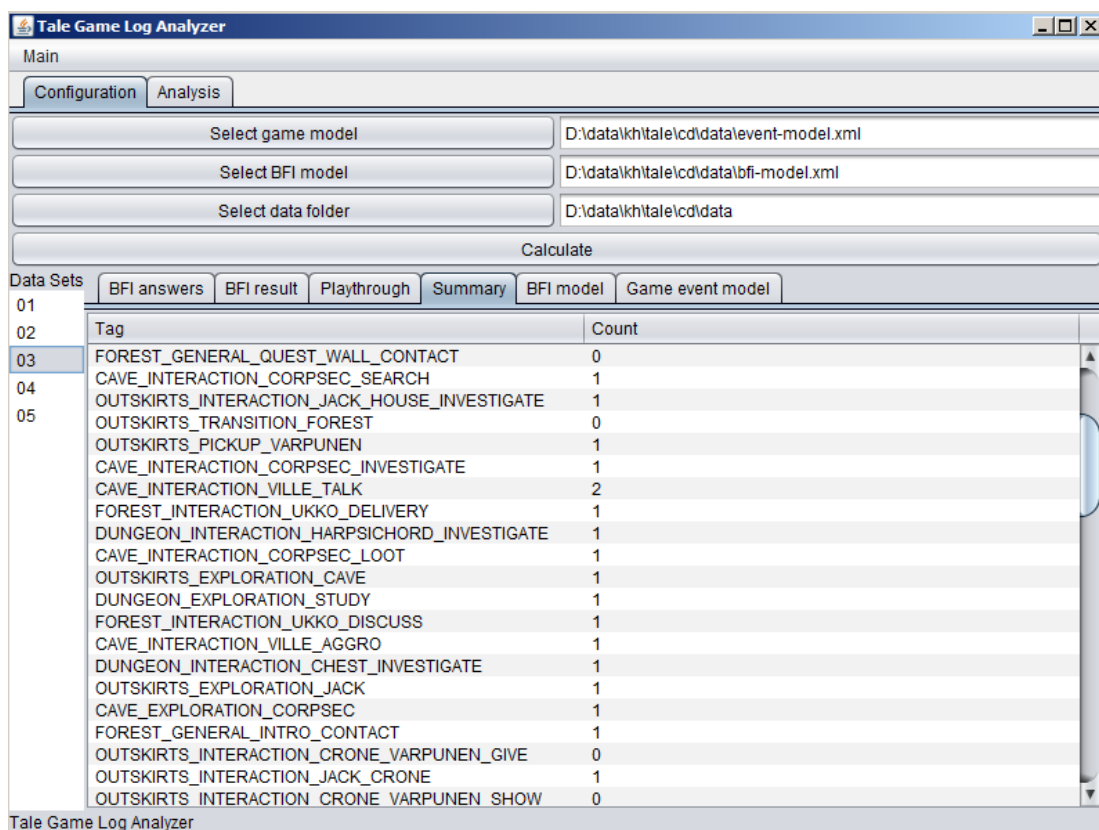
Finální cíl produkčního rozhraní je poskytnout možnost nahrát existující výpočetní model (vytvořený v rámci konfigurační fáze), a jeden herní log hráče. Po poskytnutí těchto souborů je cílem kalkulovat osobnostní profil daného hráče. Příklad užití produkčního režimu analyzáru je popsán v příloze G.2. Aktuálně implementovaná funkcionalita a její využití je popsáno v příloze G.4.



The screenshot shows the 'Tale Game Log Analyzer' application window. The 'Main' menu is open, and the 'Analysis' tab is selected. The interface includes fields for selecting a game model, BFI model, and data folder, all pointing to files in the 'D:\data\kh\tale\cd\data' directory. A 'Calculate' button is visible. Below, the 'Data Sets' section shows a table with columns for 'Trait' and 'Value'. The table contains five rows of data, with the first row (N) highlighted.

	Trait	Value
01		
02		
03	N	3.75
04	C	1.67
05	O	3.20
	E	2.38
	A	2.67

Obrázek 4.9: Výsledek klakulace FFM podle BFI



Obrázek 4.10: Souhrn herních akcí jednoho hráče



Obrázek 4.11: Ukázka umístění senzoru

5. Testování

Testování funkčnosti implementační části práce se dá rozdělit do dvou skupin: testování technické funkčnosti software, a testování funkčnosti osobnostního profilování. Technické testování je prováděno jak automaticky (unit testy), tak pomocí připravených testovních scénářů.

Testování profilování je provedeno formou výběru několika typů akcí, které může hráč vykonat, a následným porovnáním, zda získaný výsledek koreluje s osobnostními charakteristikami FFM. Pro tento účel bylo vybráno několik hráčů, kteří hrou prošli, poskytli herní log a vyplnili osobnostní dotazník.

Požadavek ostrého provozu je takový, aby hra mohla poskytovat relevantní výsledky sama, bez nutnosti používat jiných měřících nástrojů (dotazník apod.). Tato funkcionalita je konceptuálně popsána v sekci 4.7, a nebyla v rámci této práce z časových důvodů implementována.

5.1 Testování technické funkcionality

Pro technické testování bylo využito dvou přístupů: (1) automatické testování pomocí jednotkových testů, a (2) testování logické integrity hry pomocí testovních scénářů.

5.1.1 Unit testing

Pro zajištění automatického testování byl použitý framework JUnit. Z důvodu velkého objemu funkčního kódu nebyla testově pokrytá celá aplikace, pouze některé hlavní části. Použití JUnit v projektu mimo jiné demonstruje jeho integraci s frameworkem LibGDX.

5.1.2 Scenatio testing

Navržené testovní scénáře je možné rozdělit do tří hlavních kategorií: *testování konfigurace hry*, *testování exportu a debugovacích funkcionalit*, a *testování konzistence stavu hry*. V příloze F jsou uvedeny konkrétní případy scénářů. Z důvodu rozsahu a repetitivnosti scénářů se nejedná o pokrytí celého projektu, ale pouze o výběr scénářů pro demonstraci.

5.2 Testování osobnostního profilování

Konceptuální popis finální funkcionality osobnostního profilování je popsán v sekci 4.7. Zde následuje demonstrace momentálně implementované funkcionality analyzáru a jejího využití při testování funkčnosti aplikace.

Testování osobnostního profilování bylo provedeno formou následujících kroků:

1. Vypočítání FFM všech hráčů podle existujícího algoritmu (BFI).
2. Vytvoření souhrnného seznamu všech akcí hráčů, s počtem jejich opakování.
3. Vytvoření korelační matice mezi počty vyvolaných herních eventů a každou z osobnostních charakteristik.
4. Ověření získaných hodnot s očekávanými, na základě znalosti herního modelu.

Byly zjištěny následující informace:

- Události, které byly implementované za účelem zjištění konkrétní charakteristiky, vykazují vysoké míry korelace u dané charakteristiky. Výčet a popis těchto událostí znázorňuje tabulka 5.1.
- Jednotlivé události typu *průzkum oblasti* nebo *interakce s postavou* mají samy o sobě malou vypovídající hodnotu. Tento problém byl řešený seskupením těchto typů událostí, a opětovným výpočtem korelací u vytvořených skupin. Tento krok popisuje sekce 5.2.1 a tabulka 5.1. Skutečný výstup aplikace demonstruje obrázek 5.1.
- Události, u kterých byl očekávaný určitý výsledek u některé z charakteristik, sice vykazují kladnou korelaci na daném místě, ale mohou vysoce korelovat i s jinými charakteristikami. Toto může být vysvětleno závislostí vyvolání události na předešlých akcích, nebo se může jednat o skutečný vliv na charakteristiku. Tento aspekt by bylo nutno ověřit na hře bohatší na obsah, a je podrobněji rozebraný v sekci 5.2.1.
- Existují události, které musí vykonat každý hráč pro úspěšný průchod hrou. Z toho vyplývá, že tyto události nesou nulovou informační hodnotu. Stojí za zvážení např. jejich ignorování během tvorby logu, nebo filtrace před kalkulací, pro zpřehlednění matice. Jedná se například o *GLOBAL_NEW_GAME* (začátek hry), *DUNGEON_GENERAL_EXIT_WARNING* (zobrazení dialogu před koncem hry) apod.

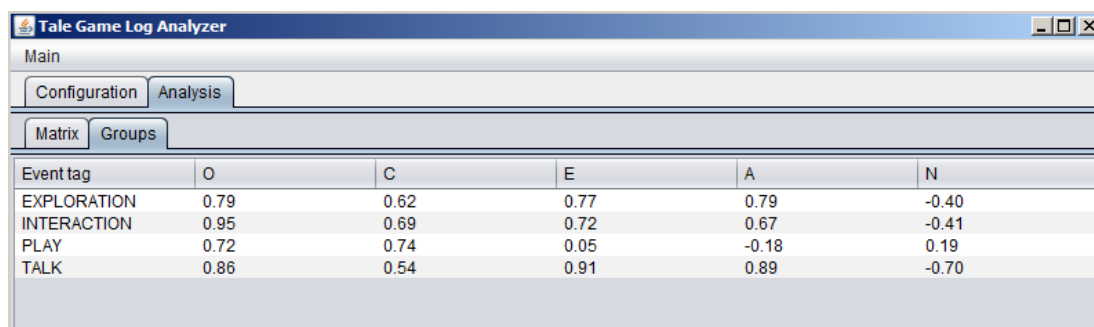
5.2.1 Skupiny typů událostí

Následující tabulka 5.1 prezentuje výsledky zjištěné u skupin událostí. Byly vybrané následující skupiny:

- *EXPLORATION* — Události prozkoumání určité oblasti herního světa, měřené pomocí senzoru.
- *INTERACTION* — Události interakce s objektem nebo postavou.
- *PLAY* — Identifikátor události vyvolání hry na harpsichord.
- *TALK* — Události komunikace s postavou v herním světě.

Event tag	O	C	E	A	N
EXPLORATION	0.79	0.62	0.77	0.79	-0.40
INTERACTION	0.95	0.69	0.72	0.67	-0.41
PLAY	0.72	0.74	0.05	-0.18	0.19
TALK	0.86	0.54	0.91	0.89	-0.70

Tabulka 5.1: Seskupené události



The screenshot shows a window titled "Tale Game Log Analyzer" with a "Main" menu. There are two tabs: "Configuration" and "Analysis". Under "Analysis", there are two sub-tabs: "Matrix" and "Groups". The "Matrix" tab is active, displaying a table with the same data as Table 5.1. The table has columns for "Event tag", "O", "C", "E", "A", and "N".

Obrázek 5.1: Výpočet korelací u seskupených typů událostí

EXPLORATION

Podle popisu charakteristik v kapitole 2 bylo předpokládáno zjištění vysokých hodnot u charakteristiky *Openness*. Z tabulky 5.2 je zřejmé, že toto měření není možné provádět u každého jednotlivého eventu, ale je nutné všechny tyto eventy seskupit a porovnávat počet jejich vyvolání u hráčů. Po seskupení vykazuje míra prozkoumávání oblasti hráčem očekávanou vysokou hodnotu *Openness*, znázorněnou v tabulce 5.1.

Event tag	O	C
FOREST_EXPLORATION_EAST	0.85	0.63
FOREST_EXPLORATION_NORTH	0.27	0.13
FOREST_EXPLORATION_SOUTH	-0.29	0.04

Tabulka 5.2: Výběr jednotlivých exploration eventů

INTERACTION

Takto označené eventy reprezentují interakci hráče s objekty nebo postavami. Tato skupina měla původně měřit především *Openness* a *Conscientiousness*, nicméně je vidět, že koreluje vysoce s každou z charakteristik (tabulka 5.1).

Závěr tohoto zjištění je takový, že při návrhu identifikátorů událostí by bylo vhodné interakce dělit do více skupin, pro zjištění konkrétnějších výsledků.

Tag *INTERACTION* je momentálně přiřazený každé události vyvolávající interakci s postavou, předmětem nebo objektem, což je velká část herního obsahu. I přesto je ale vidět, že charakteristika *Openness* s hodnotou 0.95 značí, že *Openness* hráči mají tendenci přijít do styku s mnohem větší částí obsahu hry.

PLAY

Podle kapitoly 2 je charakteristika *Openness* ovlivněná uměleckým zájmem testované osoby. Za účelem ověření tohoto vztahu byl v herním světě umístěný harpsichord, který umožňoval přehrání několika úseků klasických skladeb. Přehrání bylo možné opakovat, a tento počet opakování byl logovaný. Tabulka 5.3 ukazuje naměřené výsledky jak s eventem přehrání, tak u eventů, které tomuto předcházely (objevení dané místnosti a interakce s objektem). Vysoká hodnota charakteristiky *Openness* potvrzuje předpoklad.

Vysoká hodnota *Conscientiousness* se dá vysvětlit tím, že místnost ležela mimo hlavní linii a bylo ji třeba objevit, což zvyšuje svědomitost průchodu hrou. Ostatní charakteristiky nesly velmi nízkou hodnotu pod 0.2, což je pozitivní zjištění, protože tato interakce byla implementovaná specificky pro zjištění charakteristiky *Openness*, a korelace s jinými nebyla očekávaná.

Event tag	O	C	E	A	N
DUNGEON_EXPLORATION_STUDY	0.85	0.63	0.63	0.64	-0.24
..._HARPSICHORD_INVESTIGATE	0.92	0.61	0.42	0.23	-0.29
..._HARPSICHORD_PLAY	0.72	0.74	0.05	-0.18	0.19

Tabulka 5.3: Sekvence harpsichord eventů

TALK

U skupiny eventů vyvolaných při komunikaci s postavou byla očekávaná vysoká korelace s charakteristikou *Extraversion*, což se vzhledem k nejvyšší hodnotě u této charakteristiky, i přes relativně vysoké hodnoty u ostatních, potvrdilo. Kromě toho tato skupina vykazuje vysoké hodnoty u všech ostatních charakteristik, což se dá vysvětlit tím, že velká většina eventů ve hře do určité míry vyžadovala alespoň nějakou komunikaci s postavou.

Tuto slabinu identifikace událostí by bylo v případě dalšího rozvoje hry vhodné zohlednit takovým způsobem, aby se identifikátory nepřekrývaly takovou měrou, jako je tomu teď.

5.2.2 Závěr osobnostního profilování

V případě další analýzy znázorněnou formou by bylo vhodné pracovat s větším vzorkem, a zároveň kategorizovat herní eventy s větší granularitou, případně každému eventu přiřadit množinu typů, podle kterých by bylo možné eventy následně seskupovat. Aktuální implementace např. dělí eventy do skupin *INTERACTION* a *TALK*, přičemž každá akce komunikace s postavou ve hře je identifikovaná oběma klíčovými slovy. Díky tomu dochází k překrývání vytvořených skupin a může to být důvod, proč obě skupiny vykazují relativně vysoké korelace u všech charakteristik.

Problém s kategorizací eventů na typy by byl eliminovaný v případě algoritmického zpracování, navrženého v sekci 4.7, protože by každá událost tvořila jeden vstup s určitou vahou pro každou z charakteristik, bez ohledu na to, o jakou událost se jedná z hlediska hry.

Charakteristika *Neuroticism* nebyla zohledněna ve fázi návrhu obsahu, nicméně z tabulky 5.1 je vidět, že i tato charakteristika je měřitelná událostmi, které s ní při návrhu nepočítaly. Z prezentovaných dat se dá odvodit, že např. hráči s tendencí komunikovat s postavami budou vykazovat nižší hodnoty *Neuroticism*.

6. Možnosti rozšíření

Vývoj počítačových her je tak rozsáhlé téma, že ho zdaleka není možné obsáhnout v jedné bakalářské práci, obzvlášť pokud není zaměřená přímo na game development, ale vývoj software obecně. Následuje výčet několika návrhů na další rozvoj.

6.1 Návrh frameworku hry

Při implementaci obsahu hry za použití vytvořeného frameworku bylo odhaleno několik nedostatků. Při rozšiřování hry by bylo vhodné provést refactoring zejména logiky interakce s objekty, způsobu aktualizace questů a inicializovaných objektů apod. Ve stávajícím rozsahu je implementace dostačující vzhledem k objemu herního obsahu, nicméně při rozšiřování by docházelo k přílišnému nárůstu objemu kódu a postupné klesající modularitě celého projektu.

Jako největší slabina se jeví řešení interakce hráče s entitami. Pro tento účel je na těleso udržovaná reference v rámci objektu `InteractionSwitch`. Tato reference je jen jedna, a její hodnota se dynamicky aktualizuje podle pohybu tělesa hráče. Proto nastává problém, když by přes sebe bylo překryto několik těles. V takovém případě by bylo možné interagovat pouze s tím tělesem, se kterým přišel hráč do kontaktu jako poslední. Ve stávající verzi hry je tento nedostatek řešený návrhem herního světa tak, aby byl hráč v kontaktu vždy maximálně s jedním tělesem.

6.1.1 Přidání mechanik

Velká část herních akcí je řešena interakcí hráče s objekty a postavami pomocí dialogu a volby rozhodnutí. Tento typ řešení je velmi statický, a v případě dalšího vývoje by bylo vhodné zakomponovat do možností hry určitý způsob akčnějších reakcí na situace. Toto by zároveň zpřístupnilo více možností pro monitoring hráče.

6.2 Produkční kvalita

Tato práce byla zaměřená na technický návrh a implementaci hry a profileru, proto bylo nad její rámec vytvořit konkurenceschopnou hru po grafické a obsahové stránce. Vytvoření grafických assetů na míru, případně dobře provedená implementace trojrozměrného řešení by jistě upoutala pozornost více hráčů, a udržela

na delší dobu zájem těch, kteří začnou hrát. Více obsahu, delší herní doba, plnější svět a rozšíření možností cest k cíli ve hře by mohlo nejen zlepšit kvalitu samotné hry, ale i zpřesnit profilovací výsledky.

V případě obsahového rozšiřování je třeba ale definovat, zda je cílem výsledný produkt, který bude schopný určit osobnostní profil v relativně krátké době (jako stávající řešení), nebo se zaměřit na dlouhodobé profilování, kdy by se osobnost modelovala už během hraní, pro případné využití přímo ve hře apod.

6.3 Komplexnější analýza dat

Už zmíněná omezení implementace analyzáru v rámci této práce by bylo vhodné znova promyslet a implementovat algoritmické učení modelu za použití neuronové sítě. Tento přístup by nejen automatizoval celý proces konfigurace analyzáru, ale zároveň by umožnil vytvoření kompletního profilu, bez omezení na podmnožinu charakteristik, jako v případě stávající verze.

6.4 Postup přidání obsahu do hry

Jedním z požadavků na software byla jednoduchá rozšiřitelnost. V této sekci je stručně popsán postup přidání dalšího obsahu do existující aplikace. Inspirace se dá nalézt ve struktuře existujícího projektu. V následujících krocích je stručně popsán postup:

1. Vytvoření mapy — V případě rozšiřování hry o nový level je nutné vytvořit novou mapu. Mapy jsou tvořené v editoru Tiled [23] a umístěné v adresáři `/core/assets/maps`. Zde se taky nachází textury tilesetů, které byly použité v rámci práce, a je možné je použít i pro tvorbu nové mapy.
2. Přidání textur entit — každý level má svůj adresář pro textury `/core/assets/textures/levels`. Tento adresář je dále dělený podle typu textur. Jednotlivé entity v herním světě načítají své textury z těchto adresářů.
3. Vytvoření třídní struktury levelu v balíku `eu.janschupke.tale.content.stage.level`. V tomto kroku je vhodné se inspirovat některým z existujících levelů.

4. Inicializace levelu — samotný level je třeba inicializovat v hlavní třídě aplikace `eu.janschupke.tale.Tale`, v metodě `initScreens()`. Nutné je i přidání jeho identifikátoru ve třídě `eu.janschupke.tale.content.config.Config`.
5. Přidání přechodového senzoru do některého z existujících levelů — pro zpřístupnění obsahu pro hráče je nutné vytvořit v některém z existujících levelů senzor, po jehož aktivaci bude hráč přenesený do mapy nového levelu. Přesný formát je možné vyčíst ze tříd, které ve svém jméně obsahují `TransitionSensor` (triggerable entita) a `TransitionEvent` (událost s logikou změny levelu).

Postup se dá obecně odvodit z existující struktury adresáře `/core/assets`, obsahující `assets` hry, a balíku `eu.janschupke.tale.content.stage.level`, obsahující programový kód všech levelů a jejich logiky.

7. Závěr

V rámci práce byla navržena a implementovaná hra, schopná zaznamenávat chování svých hráčů. Tato hra, i když obsahově velmi omezená, je jednoduše rozšířitelná, a poskytuje základ pro komplexnější logování akcí hráče.

Herní logy ze hry byly analyzované za účelem zjištění vztahu mezi chováním hráče a jeho osobností. I přes relativně nízký počet vstupních datasetů byla zjištěna vysoká korelace mezi typy herních eventů a osobnostních charakteristik, které se k nim podle teorie váží. Hodnoty těchto charakteristik, oproti kterým byly logované akce porovnávány, byly vypočítané podle poskytnutých dotazníkových odpovědí uživatelů.

Analytické výsledky práce jsou vzhledem k rozsáhlosti tématiky zatím velmi omezené, nicméně prezentovaný koncept a implementovaný herní framework pokládají základ pro další výzkum v této oblasti, což se dá považovat za hlavní přínos práce.

Seznam použité literatury

- [1] CHRISTINA DEGRAFT-JOHNSON, YU-CHI WANG, MICHAEL BRADLEE SUTHERLAND, KENT L. NORMAN Relating Five Factor Personality Traits to Video Game Preference *Human-Computer Interaction Technical Report* 2013
- [2] SANDER C.J. BAKKES, PIETER H.M. STRONCK, GIEL VAN LANKVELD Player behavioural modelling for video games *Entertainment Computing*. 2012, (3): 71-79.
- [3] GIEL VAN LANDKVELD. *Quantifying Individual Player Differences* [online]. [cit. 2015-11-28].
Dostupné z: <http://www.researchgate.net/publication/261498768>
DOI: 10.1109/CIG.2011.6032007
- [4] *The Insight Game* [online]. [cit. 2015-11-27].
Dostupné z: <http://insightgame.org/game.php>
- [5] JOHN, O. P., NAUMANN, L. P., & SOTO, C. J. (2008). Paradigm Shift to the Integrative Big-Five Trait Taxonomy: History, Measurement, and Conceptual Issues. In O. P. JOHN, R. W. ROBINS, & L. A. PERVIN (EDS.), *Handbook of personality: Theory and research* (pp. 114-158). New York, NY: Guilford Press.
- [6] *New Directions in Enhancing Computer Games* [online]. [cit. 2015-11-28]. Dostupné z: <http://cie.acm.org/articles/new-directions-enhancing-computer-games/>
- [7] *The Big Five Personality Traits Model* [online]. [cit. 2015-11-27]. Dostupné z: https://www.mindtools.com/pages/article/newCDV_22.htm
- [8] *How The Big Five Personality Traits Can Create A More Effective team* [online]. [cit. 2015-11-27]. Dostupné z: <https://blog.bufferapp.com/how-the-big-five-personality-traits-can-help-you-build-a-more-effective-team>
- [9] *The 5 Domains of Play* [online]. [cit. 2015-11-27].
Dostupné z: <http://www.gdcvault.com/play/1015364/The-5-Domains-of-Play>
- [10] MCCRAE, ROBERT R. & COSTA, PAUL T., JR. Validation of the Five-Factor Model of Personality Across Instruments and Observers. *Journal of*

- Personality and Social Psychology*. Gerontology Research Center, National Institute of Aging, National Institute of Health, 1987, 52(1): 81-90. DOI: 10.1037/0022-3514.52.1.81
- [11] MCCRAE, ROBERT R. & COSTA, PAUL T., JR. Personality Trait Structure as a Human Universal. *American Psychologist*. National Institute of Aging, 1997, 52(5): 509-516. DOI: 10.1037/0003-066x.52.5.509.
- [12] *Measuring the Big Five Personality Domains* [online]. [cit. 2015-11-27].
Dostupné z: <http://pages.uoregon.edu/sanjay/bigfive.html>
- [13] *Great Ideas in Personality - Five-Factor Model* [online]. [cit. 2015-11-27].
Dostupné z: <http://www.personalityresearch.org/bigfive.html>
- [14] *International Personality Item Pool* [online]. [cit. 2015-11-27].
Dostupné z: <http://ipip.ori.org/>
- [15] JOHN, O. P., DONAHUE, E. M., & KENTLE, R. L. (1991). The Big Five Inventory—Versions 4a and 54. Berkeley, CA: University of California, Berkeley, Institute of Personality and Social Research.
- [16] BENET-MARTINEZ, V., & JOHN, O. P. (1998). Los Cinco Grandes across cultures and ethnic groups: Multitrait multimethod analyses of the Big Five in Spanish and English *Journal of Personality and Social Psychology*. 75, 729-750.
- [17] *New study throws into doubt the universality of the ‘Big Five’* [online]. [cit. 2015-11-27]. Dostupné z: <http://www.apa.org/monitor/2013/03/big-five.aspx>
- [18] *libgdx* [online]. [cit. 2016-03-30].
Dostupné z: <https://libgdx.badlogicgames.com>
- [19] *Slick2D, 2D Java Game Library* [online]. [cit. 2015-11-27].
Dostupné z: <http://slick.ninjacave.com/>
- [20] *jMonkeyEngine - 3D Game Development for Java developers* [online]. [cit. 2015-11-27]. Dostupné z: <http://jmonkeyengine.org/>
- [21] *Slick2D Wiki* [online]. [cit. 2015-11-27].
Dostupné z: <http://slick.ninjacave.com/wiki/index.php>

- [22] *Box2D, A 2D Physics Engine for Games* [online]. [cit. 2015-11-27].
Dostupné z: <http://www.box2d.org/>
- [23] *Tiled Map Editor* [online]. [cit. 2016-03-30].
Dostupné z: <http://www.mapeditor.org/>
- [24] *Git* [online]. [cit. 2016-03-30]. Dostupné z: <http://git-scm.com/>
- [25] *Bitbucket – The Git solution for professional teams* [online]. [cit. 2016-04-09].
Dostupné z: <http://www.bitbucket.org>
- [26] *Redmine* [online]. [cit. 2015-11-27]. Dostupné z: <http://www.redmine.org/>
- [27] *Gradle | Modern Open-Source Enterprise Build Automation - Gradle* [online].
[cit. 2016-03-30]. Dostupné z: <http://gradle.org/>
- [28] *Maven - Welcome to Apache Maven* [online]. [cit. 2016-03-30].
Dostupné z: <http://maven.apache.org/>
- [29] *LibGDX Developer's Guide* [online]. [cit. 2015-11-27].
Dostupné z: <https://github.com/libgdx/libgdx/wiki>
- [30] BOSE, Juwal. *LibGDX Game Development Essentials*. 1. Birmingham: Packt Publishing Ltd, 2014. ISBN 978-1-78439-929-0.
- [31] MÁRQUEZ, David Saltares and Alberto Cejas SÁNCHEZ. *Libgdx Cross-platform Game Development Cookbook*. 1. Birmingham: Packt Publishing Ltd, 2014. ISBN 978-1-78328-729-1.
- [32] NAIR, Suryakumar Balakrishnan and Anreas OEHLKE. *Learning LibGDX Game Development*. 2. Birmingham: Packt Publishing Ltd, 2015. ISBN 978-1-78355-477-5.

Seznam tabulek

5.1	Seskupené události	28
5.2	Výběr jednotlivých exploration eventů	29
5.3	Sekvence harpsichord eventů	29
F.1	TS-01 — Změna konfigurace aplikace	46
F.2	TC-0101 — Navigace do rozhraní nastavení	46
F.3	TC-0102 — Změna rozlišení hry	46
F.4	TC-0103 — Aktualizace zvukového nastavení	47
F.5	TC-0104 — Restart aplikace a kontrola stavu	48
F.6	TS-02 — Export souborů	48
F.7	TC-0201 — Export herního logu v produkčním režimu	49
F.8	TC-0202 — Testování nemožnosti ručního exportu herního logu a event modelu v produkčním režimu	49
F.9	TC-0203 — Ruční export herního logu v debug režimu	50
F.10	TS-03 — Test základních herních aspektů	50
F.11	TC-0301 — Spuštění nové hry	51
F.12	TC-0302 — Interakce a sebrání předmětu	52
F.13	TC-0303 — Splnění úkolu	52
G.1	UC-01 — Konfigurace modelu	53
G.2	UC-02 — Zjištění osobnostního profilu	54
G.4	UC-03 — Zjištění korelačních hodnot	55

Seznam obrázků

4.1	Struktura desktop aplikace	13
4.2	Struktura entit	15
4.3	Datové containery frameworku	16
4.4	Struktura tříd událostí	17
4.5	Třídní struktura herních světů	18
4.6	Struktura typů obrazovek	19
4.7	Kolizní debug režim	20
4.8	Struktura logovacích tříd	22
4.9	Výsledek klakulace FFM podle BFI	24
4.10	Souhrn herních akcí jednoho hráče	25
4.11	Ukázka umístění senzoru	25
5.1	Výpočet korelací u seskupených typů událostí	28

Seznam použitých zkratek

- AI — Artificial Intelligence
- B5 — Big Five
- BFI — Big Five Inventory
- CSV — Comma Separated Value
- FFM — Five Factor Model
- GUI — Graphical User Interface
- IPIP — International Personality Item Pool
- NEO-PI-R — Personality Inventory Revisited
- NPC — Non Player Character
- OCEAN — Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism
- RPG — Roleplaying Game
- SE — Standard Edition
- TC — Test Case
- TS — Test Scenario
- UC — Use Case
- UI — User Interface
- UML — Unified Modelling Language
- XML — Extensible Markup Language

A. Manuál hry

Tato příloha poskytuje stručný popis sestavení zdrojového projektu a následné spuštění hry.

Prerekvizity

Pro kompilaci hry je nutné mít nainstalovanou vývojovou verzi Javy (JDK) verze 8 nebo vyšší. Dále potom buďto vývojové prostředí, umožňující import Java projektů (Java SE), nebo přístup k příkazové řádce. Při první kompilaci je potřeba mít funkční připojení k internetu pro stažení potřebných závislostí. Pro spuštění hry postačuje běhové prostředí Javy (JRE) verze 8 nebo vyšší.

Sestavení a spuštění hry

Pro kompilaci hry je možno projekt importovat do vývojového prostředí jako Gradle project, nebo si naklonovat aktuální verzi z veřejného Git repozitáře projektu.¹ V době odevzdání práce je přiložená verze shodná s verzí repozitáře. Dále potom stačí kompilovat projekt klasickým způsobem.

Pro kompilaci z příkazové řádky slouží přiložený `gradlew` script v root adresáři projektu. Kompilace a spuštění aplikace pro desktop probíhá příkazem `./gradlew desktop:run`. Podrobné informace ohledně sestavení a distribuce hry lze získat na [29] v sekci *Deploying your application*.

Spuštění přiložených binárních souborů

Tato práce obsahuje přiložené dva spustitelné soubory. Jeden v *exe* formátu, druhý jako *jar* archiv. Oba reprezentují totožnou verzi hry, a k jejich spuštění není třeba žádných dalších technologií, kromě běhového prostředí Javy.

¹<https://bitbucket.org/Vehicless/tale.git>

B. Manuál analyzáru

Následuje stručný popis sestavení, spuštění a použití aplikace analyzáru.

Kompilace a spuštění

Stejně jako v případě hry, pro kompilaci analyzáru je potřeba JDK verze 8 nebo vyšší, pro jeho spuštění potom stačí JRE verze 8 nebo vyšší. Projekt využívá nástroj Maven, pomocí kterého je možné aplikaci sestavit. Maven vytváří příkazem `mvn clean compile assembly:assembly` samostatně spustitelný `.jar` soubor v adresáři `target`.

Použití analyzáru

Aplikace po spuštění vyžaduje specifikaci cest k souborům modelu a vstupních dat. Jedná se o soubory event modelu, BFI modelu, a všech uživatelských datasetů, které se budou analyzovat. Po poskytnutí cest a stisku tlačítka *Calculate* je prezentovaná množina tabulek výsledků, které umožňují prohlížet jak veškerá vstupní data pro každý dataset a vstupní modely, ale i výsledné výsledky BFI modelu. V záložce *Analysis* je možné zobrazit korelace vybraných druhů eventů a osobnostních charakteristik.

Požadovaný finální stav záložky *analysis* je popsán v kapitole 4.7 — jedná se o zmíněné produkční rozhraní. Aktuální prezentované výsledky pouze demonstrují konceptuální fungování parsování dat, a analyzáru je předmětem dalšího vývoje.

Konkrétní případy užití s konkrétními kroky (navržená plná funkcionality, i aktuálně implementovaná) jsou popsány v příloze G.

C. Event model

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<model>
  <event>
    <tag>GLOBAL_NEW_GAME</tag>
    <description>Starting new game</description>
  </event>
  <event>
    <tag>FOREST_EXPLORATION_EAST</tag>
    <description>Exploring forest east</description>
  </event>
  <event>
    <tag>FOREST_EXPLORATION_NORTH</tag>
    <description>Exploring forest north</description>
  </event>
  ...
</model>
```

D. BFI model

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<model>
  <question>
    <number>1</number>
    <description>Is talkative</description>
    <trait>E</trait>
    <reverse>>false</reverse>
  </question>
  <question>
    <number>2</number>
    <description>Tends to find fault with others</description>
    <trait>A</trait>
    <reverse>>true</reverse>
  </question>
  <question>
    <number>3</number>
    <description>Does a thorough job</description>
    <trait>C</trait>
    <reverse>>false</reverse>
  </question>
  ...
</model>
```

E. Herní log

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<playthrough>
  <event>
    <time>1458428891711</time>
    <level>FOREST</level>
    <tag>GLOBAL_NEW_GAME</tag>
  </event>
  <event>
    <time>1458428898066</time>
    <level>FOREST</level>
    <tag>FOREST_GENERAL_INTRO_CONTACT</tag>
  </event>
  <event>
    <time>1458428902049</time>
    <level>FOREST</level>
    <tag>FOREST_GENERAL_ITEM_INTRO_CONTACT</tag>
  </event>
  <event>
    <time>1458428905118</time>
    <level>FOREST</level>
    <tag>FOREST_INTERACTION_COIN_INVESTIGATE</tag>
  </event>
  ...
</playthrough>
```

F. Testovní scénáře

Název scénáře	Změna konfigurace aplikace
Identifikátor scénáře	TS-01
Účel scénáře	Změnit aktuální nastavení pomocí nabídky hry, a ověřit správnou aplikaci změn a jejich persistenci po ukončení a opětovném spuštění aplikace.
Seznam kroků (testcases):	
TC-0101	Navigace do rozhraní nastavení
TC-0102	Změna rozlišení hry
TC-0103	Aktualizace zvukového nastavení
TC-0104	Restart aplikace a kontrola stavu

Tabulka F.1: TS-01 — Změna konfigurace aplikace

Název testcase	Navigace do rozhraní nastavení
Identifikátor testcase	TC-0101
Popis testcase	Uživatel spustí hru a stiskne tlačítko <i>Settings</i> .
Očekávaný výsledek	Uživatel vidí nabídku nastavení hry.

Tabulka F.2: TC-0101 — Navigace do rozhraní nastavení

Název testcase	Změna rozlišení hry
Identifikátor testcase	TC-0102
Popis testcase	Uživatel stiskne tlačítko <i>Graphics</i> a zvolí z nabízených možností rozlišení volbu, která není aktuálně vybraná. Po obdržení notifikace se vrátí do nabídky <i>Settings</i> stiskem tlačítka <i>Back</i> .
Očekávaný výsledek	<ul style="list-style-type: none">• Uživatel dostal notifikaci o nutnosti restartu hry.• Uživatel se nachází v základním menu nastavení.

Tabulka F.3: TC-0102 — Změna rozlišení hry

Název testcase	Aktualizace zvukového nastavení
Identifikátor testcase	TC-0103
Popis testcase	
<ol style="list-style-type: none"> 1. Uživatel otevře nabídku zvukového nastavení stiskem tlačítka <i>Audio</i>. Zkontroluje konzistenci hry s aktuálním stavem menu: <ul style="list-style-type: none"> • Pokud slyší playback hudby, checkbox <i>Enable music</i> je zaškrtnutý. • Pokud slyší po stisku tlačítek zvukový efekt, checkbox <i>Enable sound</i> je zaškrtnutý. 2. Uživatel změní stav obou checkboxů a zkontroluje reakci aplikace, která se projeví hned po změně stavu každého checkboxu. 3. Uživatel zaškrtně oba checkboxy pro povolení všech zvukových efektů, změní stav zvukových sliderů v menu, stiskne tlačítko <i>Apply</i>, a zkontroluje reakci hry: <ul style="list-style-type: none"> • <i>Master volume</i> modifikuje hlasitost všech zvuků. • <i>Music volume</i> mění hlasitost hudby. • <i>Sound volume</i> mění hlasitost zvukových efektů (stisk tlačítek apod.). 	
Očekávaný výsledek	
<ul style="list-style-type: none"> • Zvukové nastavení hry je v souladu se stavem checkboxů. • Opakovaná změna stavu checkboxů mění stav příslušného nastavení. • Slidery mění hlasitost jednotlivých zvukových kategorií. • Uživatel se nachází v základním menu nastavení. 	

Tabulka F.4: TC-0103 — Aktualizace zvukového nastavení

Název testcase	Restart aplikace a kontrola stavu
Identifikátor testcase	TC-0104
Popis testcase	
<ol style="list-style-type: none"> 1. Uživatel se vrátí do hlavního menu pomocí tlačítka <i>Back</i>, a ukončí hru stiskem tlačítka <i>Exit game</i>. 2. Uživatel znovu spustí hru a zkontroluje, zda je okno v požadovaném novém rozlišení. 3. Uživatel naviguje do menu nastavení zvuku a zkontroluje, zda jsou všechny prvky ve stavu, v jakém je zanechal, a zda jejich stav odpovídá stavu hry. 	
Očekávaný výsledek	
<ul style="list-style-type: none"> • Rozlišení hry je aktualizované do stavu vybraného v rámci TC-02. • Nastavení zvuku je ve stavu, do jakého ho uživatel nastavil v rámci TC-03. 	

Tabulka F.5: TC-0104 — Restart aplikace a kontrola stavu

Název scénáře	Export souborů
Identifikátor scénáře	TS-02
Účel scénáře	
Otestovat funkčnost exportování herních logů a event modelu ze hry. Otestovat správnou blokaci ručně vyvolaných exportů v produkčním režimu hry.	
Seznam kroků (testcases):	
TC-0201	Export herního logu v produkčním režimu
TC-0202	Testování nemožnosti ručního exportu herního logu a event modelu v produkčním režimu
TC-0203	Ruční export herního logu v debug režimu

Tabulka F.6: TS-02 — Export souborů

Název testcase	Export herního logu v produkčním režimu
Identifikátor testcase	TC-0201
Popis testcase	
<ol style="list-style-type: none"> 1. Uživatel smaže adresář <i>Data folder</i>, pokud existuje. Cesta k adresáři je uvedena v hlavním menu hry. 2. Uživatel spustí produkční build hry a stiskne tlačítko <i>New game</i>. 3. Uživatel otevře menu stiskem tlačítka <i>Escape</i> a zvolí možnost <i>Exit level</i>. 4. Uživatel vyvolá restart hry stiskem tlačítka <i>New Game</i> a potvrdí zobrazený dialog. 5. Uživatel projde hrou do konce. 6. Uživatel otevře adresář <i>Data folder</i> uvedený v hlavním menu. 	
Očekávaný výsledek	
<ul style="list-style-type: none"> • Po dohrání adresář <i>Data folder</i> obsahuje dva herní logy. 	

Tabulka F.7: TC-0201 — Export herního logu v produkčním režimu

Název testcase	Testování nemožnosti ručního exportu herního logu a event modelu v produkčním režimu
Identifikátor testcase	TC-0202
Popis testcase	
<ol style="list-style-type: none"> 1. Uživatel smaže adresář <i>Data folder</i>, pokud existuje. Cesta k adresáři je uvedena v hlavním menu hry. 2. Uživatel spustí produkční build hry a stiskne tlačítko <i>New Game</i>. 3. Uživatel stiskne klávesu F6. 4. Uživatel stiskne klávesu F5. 5. Uživatel ve filesystému naviguje na cestu k <i>Data folderu</i> uvedenou v hlavním menu (bez vypnutí hry). 	
Očekávaný výsledek	
<ul style="list-style-type: none"> • <i>Data folder</i> neexistuje. 	

Tabulka F.8: TC-0202 — Testování nemožnosti ručního exportu herního logu a event modelu v produkčním režimu

Název testcase	Ruční export souborů v debug režimu
Identifikátor testcase	TC-0203
Popis testcase	
<ol style="list-style-type: none"> 1. Uživatel smaže adresář <i>Data folder</i>, pokud existuje. Cesta k adresáři je uvedena v hlavním menu hry. 2. Uživatel spustí debug build hry a stiskne tlačítko <i>New Game</i>. 3. Uživatel stiskne klávesu F6. 4. Uživatel stiskne klávesu F5. 5. Uživatel ve filesystému naviguje na cestu k <i>Data folderu</i> uvedenou v hlavním menu (bez vypnutí hry). 	
Očekávaný výsledek	
<ul style="list-style-type: none"> • <i>Data folder</i> existuje. • <i>Data folder</i> obsahuje soubor herního logu. • <i>Data folder</i> obsahuje soubor event modelu. 	

Tabulka F.9: TC-0203 — Ruční export herního logu v debug režimu

Název scénáře	Test základních herních aspektů
Identifikátor scénáře	TS-03
Účel scénáře	
Otestovat funkčnost jednotlivých herních prvků v rámci levelu hry.	
Seznam kroků (testcases):	
TC-0301	Spuštění nové hry
TC-0302	Interakce a sebrání předmětu
TC-0303	Splnění úkolu

Tabulka F.10: TS-03 — Test základních herních aspektů

Název testcase	Spuštění nové hry
Identifikátor testcase	TC-0301
Popis testcase	
<ol style="list-style-type: none"> 1. Uživatel spustí aplikaci a stiskne tlačítko <i>New Game</i>. 2. Uživatel vyvolá menu stiskem klávesy <i>Escape</i>, stiskne tlačítko <i>Exit Level</i> a následně stiskne tlačítko <i>Continue Game</i>. 3. Uživatel vyvolá menu stiskem klávesy <i>Escape</i>, stiskne tlačítko <i>Exit Level</i> a následně stiskne tlačítko <i>New Game</i>. 4. Uživatel zamítne dialog. 5. Uživatel znovu stiskne tlačítko <i>New Game</i> a potvrdí dialog. 	
Očekávaný výsledek	
<ul style="list-style-type: none"> • Prvotní spuštění nové hry nevyvolá potvrzovací dialog. • Stisk tlačítka <i>Continue Game</i> vrátí uživatele do existující herní instance. • Následný stisk tlačítka <i>New Game</i> vyvolá potvrzovací dialog. • Odmítnutí dialogu skryje dialog a nevyvolá žádnou další akci. • Opětovné vyvolání a potvrzení dialogu vytvoří novou instanci hry. 	

Tabulka F.11: TC-0301 — Spuštění nové hry

Název testcase	Interakce a sebrání předmětu
Identifikátor testcase	TC-0302
Popis testcase	
<ol style="list-style-type: none"> 1. Uživatel ve hře dojde k prvnímu předmětu (mince). 2. Uživatel sebere minci pomocí volby v nabídnutém interakčním dialogu. 3. Uživatel otevře inventář stiskem klávesy <i>I</i>. 4. Uživatel otevře quest log stiskem klávesy <i>Q</i>. 	
Očekávaný výsledek	
<ul style="list-style-type: none"> • Objekt mince zmizí z herního světa. • Inventář obsahuje minci. • Quest log obsahuje nový úkol. 	

Tabulka F.12: TC-0302 — Interakce a sebrání předmětu

Název testcase	Splnění úkolu
Identifikátor testcase	TC-0303
Popis testcase	
<ol style="list-style-type: none"> 1. Uživatel dojde k první postavě a pomocí interakce nabídne minci. 2. Uživatel otevře inventář stiskem klávesy <i>I</i>. 3. Uživatel otevře quest log stiskem klávesy <i>Q</i>. 4. Uživatel ukončí a znovu začne interakci s postavou. 	
Očekávaný výsledek	
<ul style="list-style-type: none"> • Z inventáře zmizela mince. • Stav úkolu v quest logu se aktualizovaný na <i>DONE</i>. • Možnost nabídnutí mince nadále není mezi možnostmi interakce s postavou. 	

Tabulka F.13: TC-0303 — Splnění úkolu

G. Případy užití analyzáru

Název případu užití	Konfigurace modelu
Identifikátor případu užití	UC-01
Účel případu užití	Vytvoření konfiguračního souboru pro analýzu logů
Primární aktéři <ul style="list-style-type: none">• Administrátor logů	
Pomocní aktéři <ul style="list-style-type: none">• Aplikace	
Vstupní podmínky <p>Administrátor má k dispozici vstupní soubory (<i>model eventů hry, BFI model a adresář datasetů</i>).</p>	
Základní scénář <ol style="list-style-type: none">1. Administrátor zvolí cestu ke vstupním souborům.2. Administrátor stiskne tlačítko <i>Calculate</i>.3. Aplikace nakonfiguruje výpočetní model pro profilování hráče.4. Administrátor v menu aplikace zvolí možnost <i>Export model</i> a zadá cestu pro uložení souboru.5. Aplikace ve zvoleném adresáři vytvoří soubor výpočetního modelu.	
Alternativní scénáře <ol style="list-style-type: none">1. Vstupní soubory nejsou validní. Administrátor dostane upozornění, a má možnost zvolit jiný soubor.2. Výpočetní model není možné uložit. Soubor se neexportuje, administrátor dostane notifikaci a má možnost zvolit jinou cestu pro export.	

Tabulka G.1: UC-01 — Konfigurace modelu

Název případu užití	Zjištění osobnostního profilu
Identifikátor případu užití	UC-02
Účel případu užití	Zjištění osobnostního profilu uživatele podle jeho herního logu
Primární aktéři	
<ul style="list-style-type: none"> • Uživatel analyzáru 	
Pomocní aktéři	
<ul style="list-style-type: none"> • Aplikace 	
Vstupní podmínky	
Uživatel má k dispozici herní log hráče, kterého chce profilovat, a soubor výpočetního modelu, vytvořeného v rámci UC-01.	
Základní scénář	
<ol style="list-style-type: none"> 1. Uživatel zvolí cestu ke vstupním souborům (<i>výpočetní model a log hráče</i>). 2. Uživatel stiskne tlačítko <i>Calculate</i>. 3. Aplikace vypočítá hodnoty osobnostního profilu a zobrazí je uživateli. 4. Uživatel má možnost výsledný report exportovat do souboru pomocí nabídky v menu. 	
Alternativní scénáře	
<ol style="list-style-type: none"> 1. Vstupní soubory nejsou validní. Uživatel dostane upozornění a má možnost zvolit jiný soubor. 2. Report osobnostního profilu není možné uložit. Soubor se neexportuje, uživatel dostane notifikaci a má možnost zvolit jinou cestu pro export. 	

Tabulka G.2: UC-02 — Zjištění osobnostního profilu

Název případu užití	Zjištění korelačních hodnot
Identifikátor případu užití	UC-03
Účel případu užití	Otestování vztahu osobnostních charakteristik a akcí hráče, export výstupních dat
Primární aktéři	
<ul style="list-style-type: none"> • Administrátor logů 	
Pomocní aktéři	
<ul style="list-style-type: none"> • Aplikace 	
Vstupní podmínky	
Administrátor má k dispozici vstupní soubory (<i>model eventů hry, BFI model a adresář datasetů</i>).	
Základní scénář	
<ol style="list-style-type: none"> 1. Administrátor zvolí cestu ke vstupním souborům. 2. Administrátor stiskne tlačítko <i>Calculate</i>. 3. V sekci <i>Configuration</i> se zobrazí seznam vstupních datasetů s jejich hodnotami. 4. V sekci <i>Analysis</i> se zobrazí dvě matice korelací: (1) pro jednotlivé herní eventy, a (2) pro vybrané skupiny eventů určitého typu. 5. Administrátor v menu aplikace zvolí možnost <i>Export model</i> a zadá cestu pro uložení souboru. 6. Aplikace ve zvoleném adresáři vytvoří XML soubory se zjištěnými hodnotami. 	
Alternativní scénáře	
<ol style="list-style-type: none"> 1. Vstupní soubory nejsou validní. Možnosti chyb v souborech a reakce aplikace (možné jsou i kombinace): <ol style="list-style-type: none"> (a) Vadný soubor event modelu — záložka event modelu neobsahuje data. (b) Vadný soubor BFI modelu — záložka BFI model neobsahuje data. (c) Vadná vstupní data — seznam datasetů je prázdný. <p>Administrátor má možnost zvolit jiný soubor.</p> 2. Výstupní reporty není možné uložit. Soubory se neexportují, administrátor dostane notifikaci a má možnost zvolit jinou cestu pro export. 	

Tabulka G.4: UC-03 — Zjištění korelačních hodnot

H. Obsah DVD

- **analyzer** (zdrojový kód analyzáru)
- **bin**
 - **analyzer.jar** (spustitelná verze analyzáru)
 - **tale.exe** (spustitelná verze hry)
 - **tale.jar** (spustitelná verze hry)
 - **tale-debug.jar** (spustitelná verze hry, sestavená v debug režimu)
- **data**
 - **0x** (jednotlivé dataseťy)
 - **bfi-model.xml** (BFI model)
 - **event-model.xml** (event model hry)
- **javadoc** (dokumentace hry)
- **result** (export výsledků měření)
- **tale** (zdrojový kód hry)
- **text**
 - **src** (zdrojový kód textu práce)
 - **build.sh** (shell script pro kompilaci textu práce)
 - **BP_Jan_Schupke.pdf** (tento dokument)
- **readme.txt** (struktura obsahu DVD)