



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Vytvoření počítačové verze deskové hry v jazyce Java
Student:	Karel Bušta
Vedoucí:	Ing. Josef Pavlíček, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2019/20

Pokyny pro vypracování

Vytvořte softwarovou verzi deskové hry (dle výběru - tanky, letadla, rybolov, pouštní rallye - vše dle návrhu školitele a grafika Ing. Petra Netíka).

1. Navrhněte základní kostru hry formou diagramu tříd (UML)
2. Vyberte vhodné herní módy (sám proti počítači, dva u počítače, víc hráčů na serveru)
3. Hru naprogramujte dle pravidel JEE pro běh na aplikačním serveru a s využitím prezentační vrstvy formou tenkého klienta (Vaadin framework, či jiný dle výběru a konzultace se školitelem)
4. Hru jako funkční blok umístěte na aplikační server běžící v prostředí firmy USPIN.cz
5. Definujte závěry a navrhněte další doporučení v rozvoji hry

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 11. února 2019



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Bakalářská práce

Vytvoření počítačové verze desktopové hry v jazyce Java

Karel Bušta

Katedra softwarového inženýrství
Vedoucí práce: Ing. Josef Pavlíček, Ph.D.

15. května 2019

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 15. května 2019

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2019 Karel Bušta. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Bušta, Karel. *Vytvoření počítačové verze desktopové hry v jazyce Java*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.

Abstrakt

Stolní hra, která je zvolena jako předloha bakalářské práce se nazývá Lovci žraloků, byla vymyšlena panem Ing. Josefem Pavlíčkem, Ph.D. a graficky ztvárněna panem Ing. Petrem Netíkem. V této práci je hra podrobně analyzována, na základě analýzy je navrhována počítačová verze, která je rozšířená o vlastní vylepšení, znázorněna pomocí UML diagramu a následně je implementována v jazyce Java. Samotná aplikace je nasazena na server a je spustitelná v prohlížeči za pomoci frameworku Vaadin. V závěru praktické části jsou předloženy návrhy na budoucí rozšíření počítačové verze.

Klíčová slova webová aplikace, návrh a implementace, 2D videohra, Lovci žraloků, Java, Vaadin framework, UML

Abstract

Board game which was chosen as a model of Bachelor's work is called Shark hunters, it was invented by Mr. Ing. Josef Pavlíček, Ph.D. and it was graphically portrayed by Mr. Ing. Petr Netík. In this work, the game is analyzed in-depth, from analyse experiencies is designed a virtual version of the board game, which is extended by work's author, then represented by an UML diagram and afterwards implemented in Java programming language. Application itself is deployed on a server and it is runnable in browser with the help of Vaadin framework. In the end of practic part of the work are presented future options to extend the virtual version of the game.

Keywords web application, design and implementation, 2D videogame, Shark Hunters, Java, Vaadin framework, UML

Obsah

Úvod	1
1 Cíl práce	3
2 Hraní obecně	5
2.1 Evoluce her	5
2.2 Vliv videoher na hráče	5
2.3 Ekonomická situace herního průmyslu	6
2.4 Rozdělení podle platforem	6
3 Analýza existujících řešení	9
3.1 Stolní předloha	9
3.2 UML diagramy	9
3.3 Implementace	10
4 Analýza vybrané hry	13
4.1 Lovci žraloků	13
4.2 Stolní předloha	13
4.3 Počítačová verze a vlastní úpravy	15
4.3.1 Změny oproti stolní verzi	15
4.3.2 Herní módy	15
4.3.2.1 Proti počítači	15
4.3.2.2 Hráč proti hráči	16
4.3.2.3 Dvě lodě	16
4.3.3 Mapa	16
4.3.3.1 Základní mapa - náhodná	16
4.3.3.2 Zrcadlová mapa	16
4.3.3.3 Mapa s vnitřním ostrovem	16
5 UML diagram tříd	19

6	Použité technologie	21
6.1	Java	21
6.1.1	Java EE	21
6.2	Vaadin	21
6.2.1	Výhody	22
6.2.2	Vaadin Framework 8	23
6.2.3	Vaadin Flow	23
7	Implementace	25
7.1	Nastavení projektu	25
7.2	Grafické rozhraní	25
7.2.1	Rozlišení prohlížeče	27
7.3	Propojení frontendu a backendu	27
7.4	Řídící třída logické části hry	27
7.5	Políčka	28
7.6	Mapa	29
7.6.1	Zrcadlová	30
7.6.2	Náhodná	30
7.6.3	Se středním ostrovem	31
7.7	Třídy hráčů a pohybu	31
7.8	Umělá inteligence	32
8	Testování	33
8.1	Automatické	33
8.2	Uživatelské	33
9	Budoucí rozvoj	35
9.1	Hra více hráčů	35
9.1.1	Hra v týmech	35
9.2	Nové mapy a herní módy	35
9.3	Statistiky a žebříčky	36
10	Dokumentace	37
10.1	Prostředí	37
10.2	Ovládání	37
	Závěr	39
	Bibliografie	41
A	Obrazová příloha	45
B	Obsah příloženého CD	49

Seznam obrázků

2.1	Podíl her na trhu podle platforem	7
3.1	Hra Pexeso od Ing. Josefa Pavlíčka, Ph.D.	11
3.2	Běžná aplikace prezentovaná pomocí Vaadin frameworku	11
4.1	Herní deska, která tvoří mapu hry	14
4.2	Vygenerování středního ostrova.	17
5.1	UML diagram návrhu počítačové verze	20
6.1	Struktura projektu používajícího Vaadin Framework	22
7.1	Možnosti výběru mapy, módu a jazyka	26
7.2	Část hry zobrazená po potvrzení výběru	26
7.3	Rozdělení ostrova typu 4 na jednotlivá políčka	29
7.4	Ukázka označení políček při hodu 2 hrací kostkou	32
A.1	Hrací plocha po vygenerování zrcadlové mapy	45
A.2	Hrací plocha po vygenerování náhodné mapy	46
A.3	Hrací plocha po vygenerování mapy se středním ostrovem	46
A.4	Hrací plocha přibližně ve střední části hry	47

Seznam výpisu kódu

1	Vytvoření tlačíka v Javě	27
2	Výčtový typ políček	28
3	Přiřazení obrázku k objektu třídy Tile	28
4	Inicializace ostrova typu 4 a podtypu 1, první část	29
5	Generování náhodné mapy v metodě SetUpIslands	30
6	Generování souřadnic 1. části ostrova typu 4	31

Úvod

Mnoho lidí považuje hraní her jako skvělou formu odpočinku, protože bývají nenáročné a zábavné. Velmi populární jsou stolní hry, které je téměř vždy nutno hrát ve více lidech a slouží tak k ucelování lidských vztahů. Stolních her existuje mnoho druhů, ať už znalostní, karetní, tematické nebo například dětské. Právě jednou takovou je hra Lovci žraloků od vedoucího práce a Petra Netíka.

Problém stolních her je, že abychom je mohli hrát, musíme je fyzicky vlastnit. Někdy je tento problém na straně hráče, protože hra může být drahá a někdy je zase na straně vydavatele, kdy hra nemusí být dostupná. Právě druhý důvod se objevuje u Lovců žraloků. Hra není prodejná veřejnosti, neboť ještě nebyla vydána, tudíž nemůže být ani hrána.¹

Zároveň v dnešní době téměř kdokoliv vlastní nějaký přístroj s internetovým připojením a prohlížečem, ať už stolní počítač, notebook nebo například tablet, na kterém se dají hrát některé stolní hry v jejich počítačové verzi.

Právě proto bude v této bakalářské práci nejprve podrobně analyzována hra Lovci žraloků a poté navrhnutá a implementována její počítačová verze, která bude umístěna na server uspin.cz. Na tomto serveru bude volně dostupná každému s internetovým připojením a prohlížečem. Vznikne tak možnost, jak si hru zahrát, aniž by bylo nutné vlastnit její fyzickou kopii.

¹k 27. 4. 2019

Cíl práce

Cílem bakalářské práce je vybrat určitou stolní hru, kterou je potřeba zanalyzovat a navrhnout, dále implementovat, otestovat a zdokumentovat.

V rámci teoretické části bude provedeno seznámení se základními principy hry a s jejími pravidly. Následně bude vytvořen diagram tříd pro celou aplikaci, poté podrobně rozebráno, jak aplikaci sestavit, aby vyhovovala zásadám objektivě orientovaným jazykům a na závěr popis dokumentace.

Praktická část práce bude zaměřena na implementaci, jak už v jazyce Java, tak pomocí frameworku Vaadin. Součástí bude testování funkčnosti, nasazení na server podle zadání, dále návrh a doporučení v rozvoji aplikace.

Hraní obecně

Hraní her jako forma zábavy existuje odjakživa, ale teprve v posledních desítkách let se pořádně rozrůstá. Dnešní doba je poměrně mírnější a člověk má více času na zábavu, kterou u mnoha lidí ztvárňují právě hry. V této kapitole si o hrách obecně řekneme několik informací, jako krátký historický úvod, vliv videoher na hráče nebo například na jakých platformách se hry hrají nejvíce.

2.1 Evoluce her

Hraní her doprovázelo člověka vždy a čím vyvinutější lidstvo bylo, tím sofistikovanější hry vznikaly jako forma zábavy. Dlouhá léta se hrály pouze deskové hry, nejstarší takovou, která byla objevena, je Senet.² V 70. letech minulého století se však stal hardware mnohem dostupnější a započala tak éra videoher. Prudce se rozšířily herní automaty a vznikaly jedny z nejslavnějších her vůbec, také byly představeny první domácí a přenosné herní konzole. [1] V dalším desetiletí tento průmysl zažil ještě větší rozvoj. Byl vytvořen první herní 3D svět, byla vydána hra Pac-Man a Mario, zakladatel nejprodávanější herní série [2], a celkově byly utvořeny základy herním žánrům všeho druhu. Od té doby se hranice kvality počítačových her posouvají jen výš. V dnešní době jsou herní světy tak kvalitní, že si je lehké splést se světem opravdovým, je možné hrát hry ve virtuální realitě a dostává se nám herních příběhů tak kvalitních, že vyčnívají nad většinou filmů. Právě úroveň nynějších her je dělá tak chtěnými a návykovými, až si klademe otázky, jestli hraní her spíše neškodí.

2.2 Vliv videoher na hráče

Dřívější mínění o hraní videoher bylo poněkud záporné. Dopad na takový úsudek byl způsoben především tím, že kdysi byly hry tak jednoduché, že

²K 8. 5. 2019

člověku nepřinášely nic jiného, než zábavu. Nyní je však přístup ke hrám a jejich hraní poněkud jiný, a to kvůli rozsáhlému výběru pro všechny věkové kategorie a takové komplexnosti určitých titulů, která při hraní napomáhá rozvíjet mysl, jak už v taktickém rozhodování, soustředění se nebo například v rozvoji cizího jazyka.

Například hraní 3D videoher má pozitivní vliv na hipokampus, část limbického systému, která má velký podíl na schopnosti orientace v prostoru a udržování krátkodobé paměti. [3] Dále slouží jako velice účinné analgetikum, kdy pacienti při hraní pocítují bolest až o 50% méně, nebo k rozvoji čtení, kdy děti s dyslexií dosáhly během 12 hodin hraní lepších výsledků, než za celý rok odborných lekcí. [4]

2.3 Ekonomická situace herního průmyslu

Od dob, co se hry staly dostupnými pro širokou veřejnost, se v herním průmyslu pohybuje obrovská suma peněz a každý rok se tato suma zvětšuje. Například v roce 2008 byly odhadované tržby v herním průmyslu 73,5 miliardy dolarů, kdežto v roce 2018 toto číslo vzrostlo na 138,5 miliardy dolarů. [5] Ke srovnání odhadované tržby filmového průmyslu za rok 2018 jsou 41,7 miliard dolarů [6], a to je méně než třetina herního průmyslu. Například v hitu Fortnite z roku 2017 je již zaregistrovaných 250 miliónů hráčů a hru v jednu chvíli hraje až téměř 11 miliónů lidí z celého světa. [7] Proto není divu, že ředitel největší streamovací³ společnosti Netflix uvedl, že Fortnite je pro Netflix větší konkurencí než všechny ostatní streamovací služby. [8]

Žádnou výjimkou v herních tržbách není ani Česká republika. V roce 2018 činily odhadované tržby 3 miliardy korun, a to je dvakrát více než obrat filmového průmyslu v Čechách. [9] Velkou zásluhu na tržbách v roce 2018 měla velmi očekávaná hra Kingdom Come: Deliverance od studia Warhorse, které se prodalo přes dva milióny kopií.

Při pozorování dat na obrázku 2.1 si všimneme, že celkový peněžní obrat se zvětšuje především rozmachem mobilního hraní v posledních letech - v roce 2018 tvořil 51% a předpokládá se, že bude nadále růst. Dále má velký vliv na stálé zvyšování fakt, že hraní se dnes považuje za sport a vznikající kompetitivní soutěže přitahují další zájemce, kteří mohou vidět v esportu svou budoucnost.

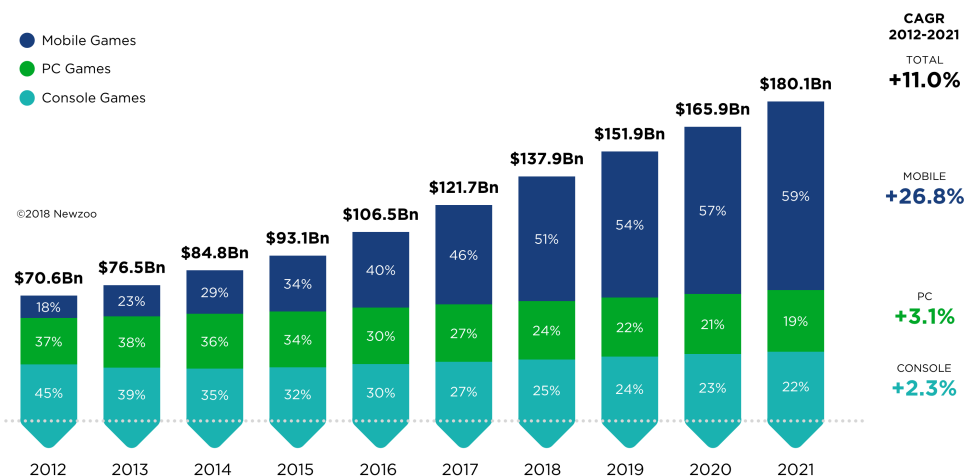
2.4 Rozdělení podle platform

Dvě desetiletí zpět byla nejrozšířenější herní platforma domácí počítač. Nynější technologická situace nám umožňuje spouštět náročné tituly, které byly dříve určené pouze na osobní počítače, i na mobilních přístrojích, a to především

³streamovat znamená vysílat přenos přes internet

2012-2021 GLOBAL GAMES MARKET

REVENUES PER SEGMENT 2012-2021 WITH COMPOUND ANNUAL GROWTH RATES



Obrázek 2.1: Podíl her na trhu podle platform

kvůli masivnímu nárůstu výkonu u těchto zařízení. Velkou roli představuje fakt, že mobilní telefony nebo tablety jsou levné a široce dostupné⁴, a téměř každý titul na nich lze spustit, zatímco do počítače je nutné investovat velké částky, aby mohl hráč využít plný potenciál her. [10] Další samostatná kapitola jsou konzole, které se dají pořídit za zlomkovou cenu výkonného herního počítače. Také vychází nové verze až po několika letech⁵ a každá generace je plně podporována téměř každým konzolovým vývojářem. Na obrázku 2.1 vidíme, jak se v důsledku zmíněných vlastností mění podíl platform používaných ke hraní.

Nejdůležitějším poznatkem, který nám obrázek 2.1 dává je, že podíl mobilních tržeb bude stále růst [11], především kvůli přenosnosti a jednoduchosti, díky čemuž se vývojáři budou více zaměřovat na tvorbu mobilních her. Aplikace vytvářená v této práci má být spustitelná v prohlížeči, tudíž bude hratelná na jakémkoliv přenosném zařízení s internetem, z čehož vyplývá, že aplikace je mířená na část herního trhu s největším podílem tržeb.

⁴66% lidstva vlastní telefon (ke dni 27. 3. 2019)

⁵Playstation 3 - 2006, Playstation 4 - 2013

Analýza existujících řešení

V této kapitole budou nastíněny existující řešení podobné samotné stolní verzi hry, dále budou podrobně analyzovány UML diagramy již existujících počítačových verzí podobných her a v závěru kapitoly bude proveden rozbor jejich implementační stránky.

3.1 Stolní předloha

Samotná stolní verze hry je originální, přesto se v ní dají nalézt prvky převzaté z jiných her. Například pohyb po mapě o tolik políček, kolik hráč hodí na kostce je běžná herní mechanika a používá se například ve hře Člověče, nezlob se. Dále systém sbírání bodů a výhra toho hráče, který na konci hry má bodů nejvíce je také běžný prvek používaný téměř v jakékoliv kompetitivní hře.

Části hry, které jsou přidány navíc do počítačové verze, byly také inspirovány již existujícími prvky ostatních her. Například herní mód hráč proti počítači je v dnešní době v téměř každé hře pro dva hráče. Také možnost generování více druhů map je běžným prvkem her tohoto druhu.

Z předchozích odstavců vyplývá, že hra je sice inspirovaná několika jinými hrami, ale je vskutku originální, a tudíž neexistují žádná vzniklá řešení a tato bakalářská práce bude prvním takovým.

3.2 UML diagramy

Tyto diagramy slouží k navrhnutí základní kostry aplikace na základě předchozí analýzy, a proto jsou nezbytně nutné k hladké implementaci. Protože stolní předloha je originální, neexistují k ní žádné počítačové verze, a tudíž ani navrhované diagramy, proto je nutné zanalyzovat UML diagramy některých jiných her. Například stolní hra Šachy má lehce jednodušší diagram, ale základní myšlenka je stejná jako u Lovců žraloků [12] - aplikace je tvořena hlavní třídou, která řídí běh hry a podpůrnými třídami, které se starají o jednotlivé

části aplikace. Stejně je tomu u diagramu Člověče, nezlob se, kde můžeme vidět, že v aplikaci existují dvě hlavní třídy, kdy jedna se stará o grafickou část a druhá o logickou část aplikace. [13] Tento přístup je velmi důležitý, protože se s ním zajišťují jednodušší úpravy v případě budoucích změn. Například pokud by bylo v budoucnu nutné změnit grafické rozhraní, stačilo by provádět úpravy ve dvou hlavních třídách a ostatní by nebylo třeba vůbec navštěvovat.

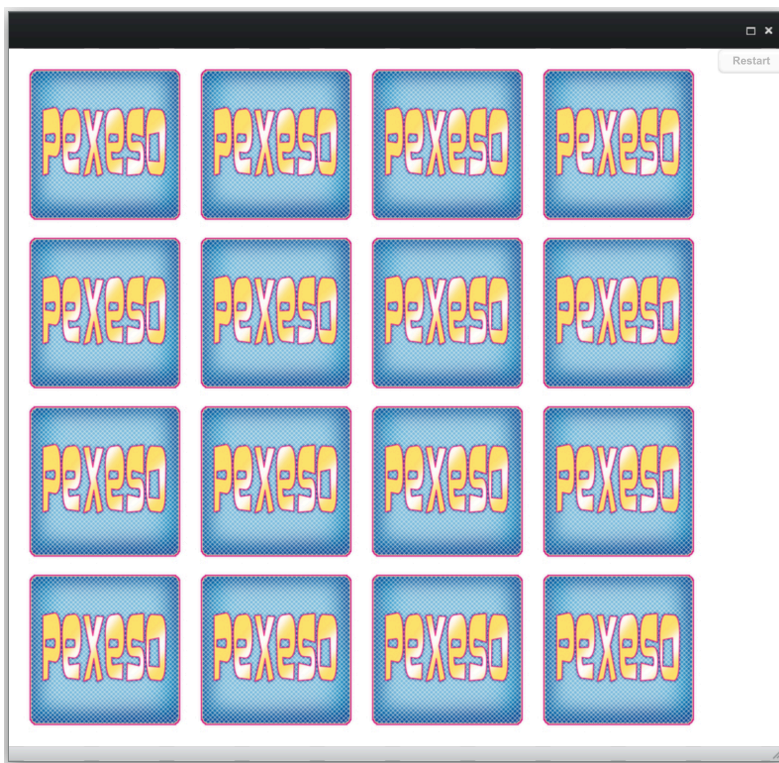
Ovšem u těchto diagramů velice záleží na tom, jak zkušený je autor s tvorbou aplikací. Výsledné diagramy jsou velice individuální a je téměř jisté, že pokud by dva zkušení programátoři měli za úkol vytvořit diagram pro jednu aplikaci, výsledky by byly odlišné. Například výše zmíněný diagram Šachů by mohl být mnohem složitější, nebo naopak diagram Člověče, nezlob se by mohl být navržen jednodušeji. Proto tyto schémata slouží jen jako „odrazový můstek“.

3.3 Implementace

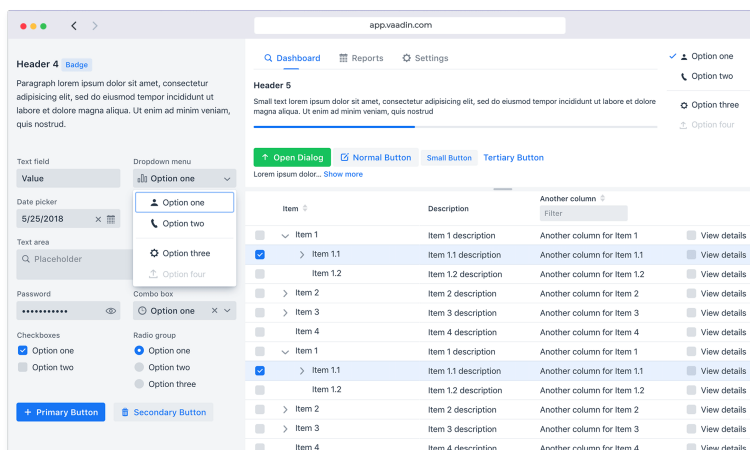
Co se týče implementace, tak už vzniklo mnoho aplikací tohoto typu, jak v programovacím jazyce Java, tak ve frameworku Vaadin. Jednou z nich je například hra Pexeso od vedoucího této práce. [14] Samotná hra Pexeso a její mechaniky nemají moc společného s Lovci žraloků, ale po implementační stránce jsou si velmi podobné. Obě hry fungují na principu mapy, která je tvořena tabulkou o určitých rozměrech a uvnitř každého políčka tabulky jsou objekty, se kterými hra pracuje. Pexeso je tvořeno tabulkou 4x4 políček a v každém políčku je objekt se stejným obrázkem, jak můžeme vidět na obrázku 3.1. Pokud hráč na objekt klikne, změní se mu obrázek a hráč vidí, že se otočila karta. V Lovcích žraloků jsou určitá políčka zakrytá a pokud na nějaké hráč dopluje lodí, změní se objektu na určitém políčku obrázek a hráč také vidí, že se karta otočila.

Implementace v jazyce Java se velmi odvíjí od vytvořeného UML diagramu a je kvůli rozšířenosti jazyka poněkud snazší, protože lze mnoho problémů dohledat. Naopak Vaadin framework tolik rozšířený není, navíc je určený pro tvorbu podnikových aplikací, tudíž k tématu tvorby her ve Vaadinu tolik informací neexistuje. Na obrázku 3.2 můžeme vidět, jak vypadá běžná aplikace vytvořená v Javě a prezentovaná pomocí Vaadin frameworku.

3.3. Implementace



Obrázek 3.1: Hra Pexeso od Ing. Josefa Pavlíčka, Ph.D.



Obrázek 3.2: Běžná aplikace prezentovaná pomocí Vaadin frameworku

Analýza vybrané hry

Obsahem této kapitoly bude přiblížení vybrané hry a její kompletní analýza. Budou rozebrána herní pravidla, také budou popsány jednotlivé kartičky a herní mechaniky a následně budou podrobně vylíčena všechna vylepšení v počítačové verzi oproti stolní předloze.

4.1 Lovci žraloků

Zvolenou hrou a tématem bakalářské práce je titul Lovci žraloků. Samotný nápad hry vymyslel vedoucí práce Ing. Josef Pavlíček, Ph.D. a originální grafický návrh dodal pan Ing. Petr Netík.

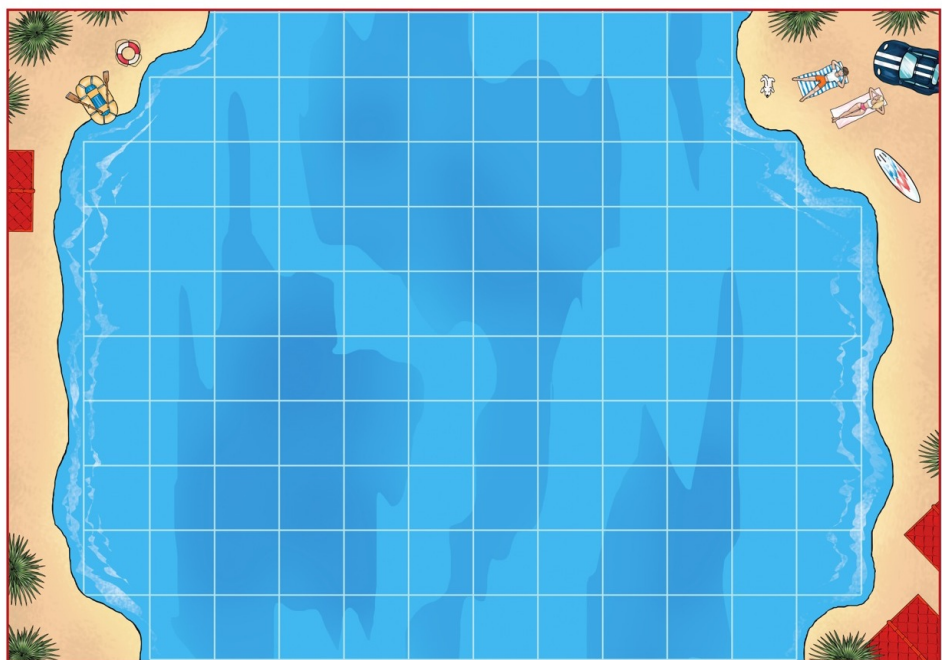
4.2 Stolní předloha

Stolní verze hry je poněkud nenáročná na pochopení, jednoduše hratelná a určená pro dva hráče. Součástí balení je herní deska, která slouží jako mapa, dále kartičky čtvercového rozměru, které se umísťují na herní plochu.

Herní deska neboli mapa, je ztvárněna jako moře mezi dvěma krajními ostrovy, kde na moři je rozložena téměř celá tabulka o rozměrech 12x10 sloužící pro pokládání čtvercových kartiček. V tabulce ovšem chybí několik políček proto, aby se hráči mohli pohybovat pouze po moři, jak můžeme vidět na obrázku 4.1. Mírná nevýhoda takového rozložení políček je, že na straně pravého hráče je o dvě políčka méně, tudíž se hra stává lehce nevyrovnanou.

Samotné kartičky jsou rozděleny do několika druhů, a to podle jejich funkcí ve hře. Velmi důležitým prvkem hry jsou kartičky ostrovů, které hráči rozmísťují na herní plochu. Typů ostrovů je celkem pět, a to

- samotné políčko,
- dvě políčka za sebou,



Obrázek 4.1: Herní deska, která tvoří mapu hry

- tři políčka za sebou,
- tři políčka do tvaru L,
- pět políček do tvaru [.

Každý hráč rozmístí na mapu náhodně tři ostrovy. Poté si každý hráč vybere jednu ze sedmi možných barev lodí a umístí ji na okraj svého břehu.

Ve chvíli, kdy jsou na mapě rozmístěny ostrovy a hráči mají vybrané lodě, zamíchají se ostatní kartičky tak, aby nebylo vidět co na nich je, a poté se náhodně rozmístí na volná políčka na herní ploše. Zbývající kartičky náhodně rozmístěné na zbytek mapy jsou třech typů, a to ryby, žraloci a bubliny. Ryby a žraloci se ještě dělí na podtypy podle počtu kusů na kartičce. Každá z těchto kartiček má svůj herní účinek, pokud je vylovena hráčem. Odkrytí

- jedné ryby přidá do finálního skóre jeden bod,
- dvou ryb přidá do finálního skóre dva body,
- tří ryb přidá do finálního skóre tři body,
- jednoho žraloka ubere od finálního skóre jeden bod,

- dvou žraloků ubere od finálního skóre dva body,
- bublin neprovede žádnou úpravu na finálním skóre.

Smysl celé hry spočívá v tom, že hráč na tahu háže kostkou, poté pluje se svou lodí o daný počet políček, odkryje políčko, na které doplul a zapůsobí na něj efekt kartičky, která na políčku ležela. Poté hraje další hráč. Hra končí ve chvíli, kdy na herní ploše nejsou žádná políčka k odkrytí a vyhrává hráč, který má nasbíraných více bodů.

Důležité je také zmínit styl pohybu po mapě. Lodě se mohou pohybovat po svislé a vodorovné ose, nikoliv však po diagonále. To znamená, že pokud hráč hodí například 3, může o tři políčka nahoru po svislé ose, také o dvě políčka nahoru po svislé a jedno políčko vpravo po vodorovné ose, ale nemůže plout o tři políčka po svislé a zároveň o tři políčka po vodorovné ose. Lodě mohou plout kdekoliv po moři, nemohou ale na políčka, která jsou ostrovy. Pokud hráč hodí kostkou takový počet, že by se mohl pohybovat jen po ostrovech, ale nemohl by plout nikam po moři, zůstává na místě a hraje druhý hráč.

4.3 Počítačová verze a vlastní úpravy

Cílem této bakalářské práce je přetvořit stolní verzi na počítačovou, a to se všemi vlastnostmi jaké stolní verze má, tudíž veškeré inovace a vylepšení, která budou nyní zmíněna, jsou v bakalářské práci přidána z vlastní vůle autora.

4.3.1 Změny oproti stolní verzi

Nevyhnutelnou změnou počítačové verze je automatizace některých funkcí hry. Například hod kostkou již nebude na hráčích, ale bude náhodně generovaný aplikací. Další automatizací bude pokládání ostrovů na herní plochu, i když tato změna je spíše kladná, protože hráči u stolní verze pokládali ostrovy náhodně a aplikace bude rozmístování provádět také náhodně, jen hráčům ulehčí práci. Dále kvůli automatizaci rozmístování ostrovů budou moci být předgenerované některé mapy.

4.3.2 Herní módy

U stolní verze hry je nutné, aby hráči byli dva, jinak je hra nehratelná. U této aplikace tomu tak nebude, neboť bude možnost hrát proti počítači. Také zde bude mód, kdy hráči budou moci dokupovat lodě.

4.3.2.1 Proti počítači

Jak již název kapitoly napovídá, tento mód postaví hráče proti umělé inteligenci, která ovšem bude hrát podle stejných pravidel jako hráč. Tím je

myšleno, že umělá inteligence nebude vědět co se pod danými políčky skrývá, stejně jako hráč, a bude tedy políčka vybírat náhodně.

4.3.2.2 Hráč proti hráči

Tento mód je obrazem originální stolní verze, kdy proti sobě nastoupí dva hráči. Každý bude mít vlastní loď a kostku a budou mezi sebou soupeřit o to, kdo nasbírá více ryb.

4.3.2.3 Dvě lodě

Díky tomuto módu se bude moci každý z hráčů v určité části hry rozhodnout, zda prodat několik svých ryb a koupit za ně další loď, kdy poté ovládá v jednom kole obě dvě a má tudíž větší šanci získat ryby, ale také větší šanci o nějaké přijít. Loď navíc bude možno pořídit za deset ryb, přičemž na celé herní ploše bývá okolo padesáti ryb, takže hráči musí pečlivě přemýšlet, zdali se nákup lodě navíc vůbec vyplatí.

4.3.3 Mapa

Ještě dříve, než se na herní plochu rozmístí políčka s rybami, žraloky a bublinami, je část mapy pokryta ostrovy. Každý ostrov má ještě čtyři podtypy, které jsou u prvních tří typů pouze vzhledové, u posledních dvou však určují i orientaci, kterou bude ostrov mít. Pro každou mapu platí, že se na ní rozmístí celkem šest ostrovů.

4.3.3.1 Základní mapa - náhodná

Tato mapa je vytvořena zcela náhodně, tudíž se může stát, že budou například všechny ostrovy na levé části mapy, nebo že budou všechny ostrovy prvního typu. Z toho důvodu není herní plocha vždy rovnoměrně rozložená, a tudíž má jeden z hráčů lepší startovní pozici.

4.3.3.2 Zrcadlová mapa

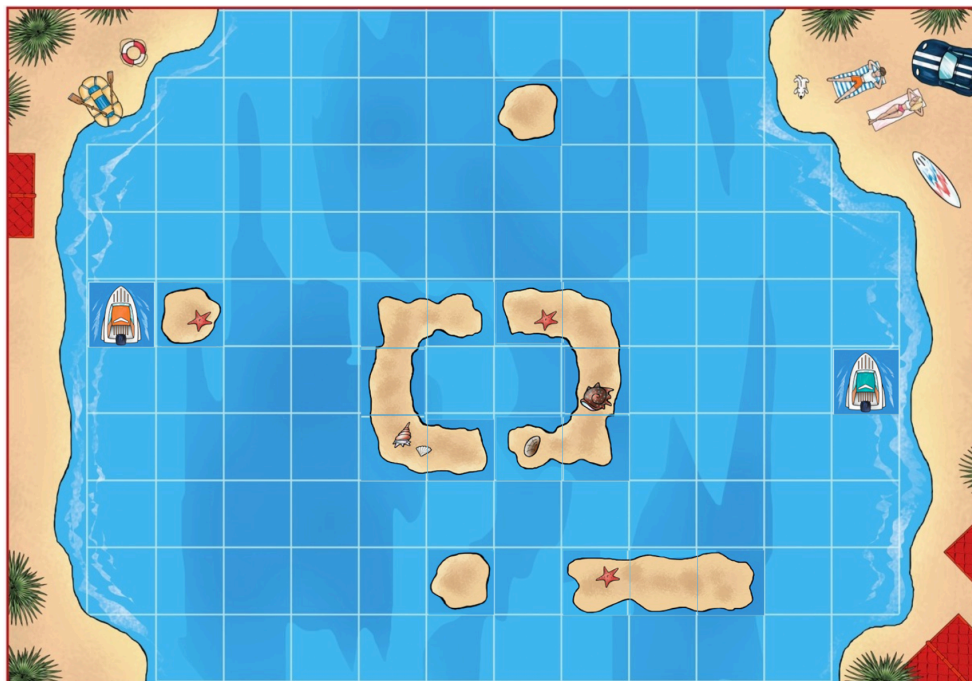
Tento druh mapy je nejvíce vyvážený vůči hráčům, neboť rozestavení ostrovů se snaží být co nejvíce stejné při pohledu na mapu, jako na pravou a levou polovinu. To znamená, že se hra pokusí postavit herní plochu tak, aby na levé polovině byl stejný počet ostrovů jako na pravé polovině, a zároveň aby ostrovy z většiny odpovídaly svým protějškům i typem.

4.3.3.3 Mapa s vnitřním ostrovem

Při generování této herní plochy se nejdřív umístí doprostřed dva ostrovy pátého typu tak, že uvnitř vzniknou dvě volná políčka, jak můžeme vidět na obrázku 4.2. Na jedno z těchto dvou míst se umístí speciální políčko deset

4.3. Počítačová verze a vlastní úpravy

ryb a na druhé osm žraloků, a to náhodně. Zbylé čtyři ostrovy se rozmístí náhodně, ale v každé čtvrtině mapy bude pouze jeden. Přidáním této mapy vznikne nový herní prvek, kdy hráč může velmi riskovat a buďto získat nebo ztratit velký počet ryb v jednom tahu.



Obrázek 4.2: Vygenerování středního ostrova.

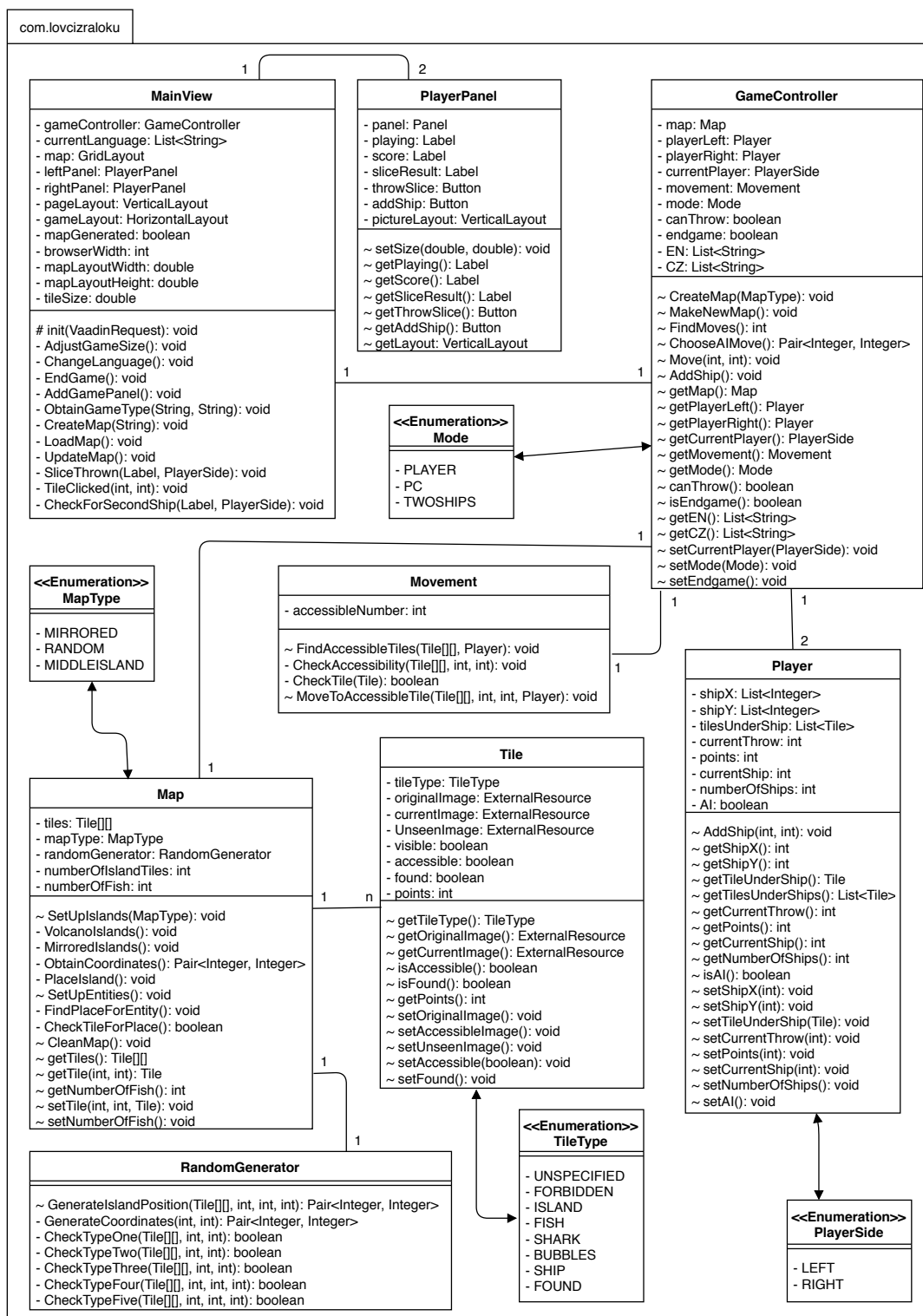
UML diagram tříd

Každá moderní webová aplikace dnešní doby má nějaký takzvaný frontend - část aplikace tvořící grafické rozhraní, se kterým komunikuje uživatel - a backend - část aplikace, se kterou komunikuje daný frontend, a ve které je obsažena všechna logika, neboli způsob jak aplikace funguje. Jednou z nejdůležitějších myšlenek při návrhu takové aplikace je mít frontend a backend co nejvíce na sobě nezávislý. Toho se docílí tak, že v návrhu bude hlavní třída řídící ostatní třídy frontendu a hlavní třída řídící ostatní třídy backendu. Uživatelské rozhraní a logická strana aplikace pak mezi sebou komunikují pouze přes tyto dvě hlavní třídy a zaručí se tak téměř absolutní nezávislost. Celá tato myšlenka se využije například v případě, kdy je grafické rozhraní zastaralé a vytváří se nové.

V diagramu Lovců žraloků je navržena třída `MainView`, která ovládá grafické rozhraní, také třída `GameController`, která řídí logickou stránku hry, a pouze tyto dvě třídy mezi sebou komunikují v zájmu komunikace frontendu a backendu.

Dále jsou v aplikaci navrženy třídy `Player`, `Map` a `Movement`, které komunikují s třídou `GameController`. Třída `Player` zastupuje hráče a v jejích objektech se ukládají veškeré informace o jednotlivých hráčích. Třída `Movement` řídí pohyb hráčů po mapě a třída `Map` obsahuje všechny informace o mapách a kontroluje jejich generování pomocí další třídy `RandomGenerator`. Mapy jsou vytvářeny z objektů třídy `Tile`, která reprezentuje jednotlivá políčka.

5. UML DIAGRAM TRŽÍD



Obrázek 5.1: UML diagram návrhu počítačové verze

Použité technologie

6.1 Java

Programovací jazyk Java vznikl již v roce 1995 a od té doby se stal velice rozšířeným, podle počtu depozitářů na uložišti GitHub je třetím nejpopulárnějším a podle výsledků vyhledávačů je dokonce na prvním místě [15]. Stalo se tak především kvůli přenositelnosti a možnosti spuštění téměř na jakémkoliv zařízení. Od roku 2007 je Java vyvíjena pod open-source licencí. [16]

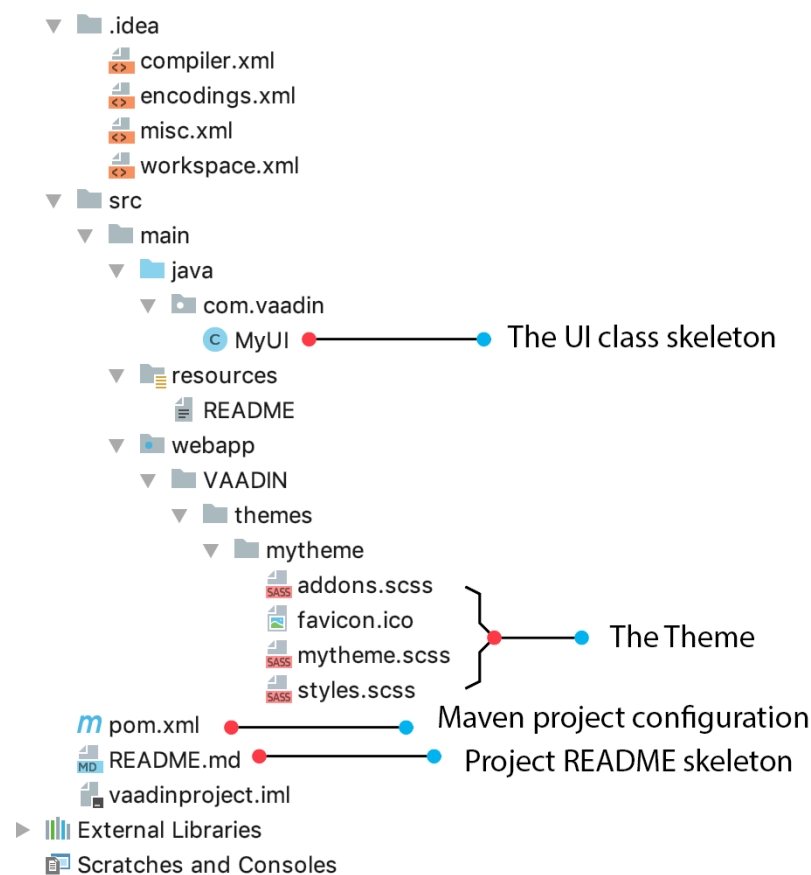
6.1.1 Java EE

Tato varianta Javy je nadstavbou základní verze Java SE, její označení EE reprezentuje Enterprise Edition, neboli verze určená k vývoji a údržbě podnikových aplikací a informačních systémů. Hlavními přínosy oproti běžné variantě Javy jsou především specifikace pro vývoj webových aplikací, přístup k relačním databázím a podpora webových služeb. [17] Právě kvůli prvnímu zmíněnému přínosu je při vývoji hry v této bakalářské práci používána Java EE.

6.2 Vaadin

Vaadin je dlouhodobě vyvíjený framework pro Javu, který umožňuje vytváření grafického rozhraní pro webové aplikace pouze za pomoci Javy neboli k tvorbě není téměř potřebné znát programovací jazyky jinak nutné. [18] Takových možností framework dosahuje pomocí využívání Google Web Toolkit⁶ a jeho schopnosti překládat části aplikace či celé aplikace psané v Javě do JavaScriptu, který je následně spuštěn v prohlížeči.

⁶souhrn nástrojů umožňující vytvářet a spravovat JavaScript aplikace v Javě



Obrázek 6.1: Struktura projektu používajícího Vaadin Framework

6.2.1 Výhody

Při tvorbě logické stránky aplikace Vaadin práci lehce urychlí při samosprávě komunikace mezi serverem a prohlížečem, naopak poněkud zpomalí tvorbou nového projektu, protože je nutné dodržet dopředu určenou strukturu, jak můžeme vidět na obrázku 6.1. Kde svou výraznou sílu framework ukazuje je rychlost a jednoduchost implementace uživatelského rozhraní, které je skládáno z různých komponent. Každá komponenta se v Javě prezentuje jako samostatná třída a do aplikace se umísťuje vytvořením objektu. Nejen tato vlastnost urychlí vznik aplikace, ale také fakt, že oficiálních neplacených hotových komponent je k dispozici již přes 25⁷, a ještě mnohem více od ostatních vývojářů.

V nynější době jsou plně podporovány dvě verze frameworku - Vaadin Framework a Vaadin Flow.

⁷ke dni 3. 4. 2019

6.2.2 Vaadin Framework 8

Tato verze je předchůdce Vaadin Flow, to ale neznamená, že je starší nebo horší. Verze je stále plně podporována a bude podporovaná až do roku 2022. [19] Framework 8 byl představen v roce 2017, ale je velice podobný předešlým verzím, a tudíž k němu existuje rozsáhlá komunita s již mnoho vyřešenými problémy a nedostatky a velmi přehledá dokumentace. Tento fakt je velmi důležitý při implementaci jakékoliv aplikace, a proto bude tato práce implementována ve verzi 8.

6.2.3 Vaadin Flow

Flow se označují verze 10, 11, 12 a 13. Jejich rozdílné označení ukazuje především schopnost používat nejen oficiální nebo komunitou vytvořené komponenty přímo pro Vaadin, ale podpora všech technologií kompatibilních s Web Components⁸.

⁸Souhrn znaků pro standardní model komponentů

Implementace

V této kapitole bude probrána kompletní implementační stránka bakalářské práce. Bude zde popsáno vše od založení projektu až po nasazení funkční aplikace na server.

7.1 Nastavení projektu

První krok, který musí být splněn před začátkem samotného programování je vytvoření a nastavení projektu. Způsob provedení se může a nejspíše bude lišit podle používaného editoru/IDE⁹. K vývoji Lovců žraloků je používáno vývojové prostředí IntelliJ IDEA od JetBrains. V tomto IDE je možné buďto vytvořit nový Maven projekt, ve kterém se zvolí Vaadin archetype, poté se vyplní GroudID a IDE samo získá všechny potřebné soubory a vytvoří nový projekt. Další možnost je stáhnout již vytvořený projekt z oficiálních stránek Vaadin a ten nahrát do IDE. Když má projekt stejnou strukturu jako na obrázku 6.1, můžeme začít programovat vlastní hru.

7.2 Grafické rozhraní

Z uživatelského hlediska bývá u většiny aplikací na první pohled nejdůležitější vzhled. Mnoho uživatelů se kvůli neintuitivní grafické stránce rozhodne určité aplikace vůbec nepoužívat, ačkoliv mohou nabízet velmi mnoho. Pokud jde o implementaci hry, musí být důraz na vzhled ještě větší. I přesto, že tato hra není vyvíjena v žádném grafickém enginu¹⁰, nýbrž ve webovém frameworku, bude aplikace vytvořena s velkým důrazem na příjemné uživatelské rozhraní.

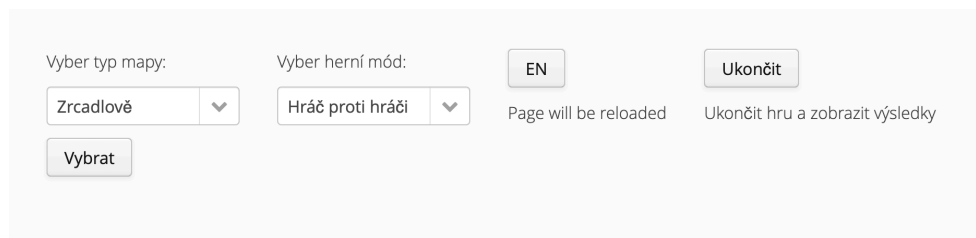
Po spuštění aplikace bude mít uživatel na výběr, zdali bude chtít vygenerovat mapu náhodnou, zrcadlovou nebo s prostředním ostrovem, dále jestli bude chtít hrát proti jinému uživateli nebo proti počítači, také jaký počet lodí

⁹Integrated Development Environment - vývojové prostředí usnadňující vývoj aplikací

¹⁰Vývojové prostředí usnadňující vývoj her

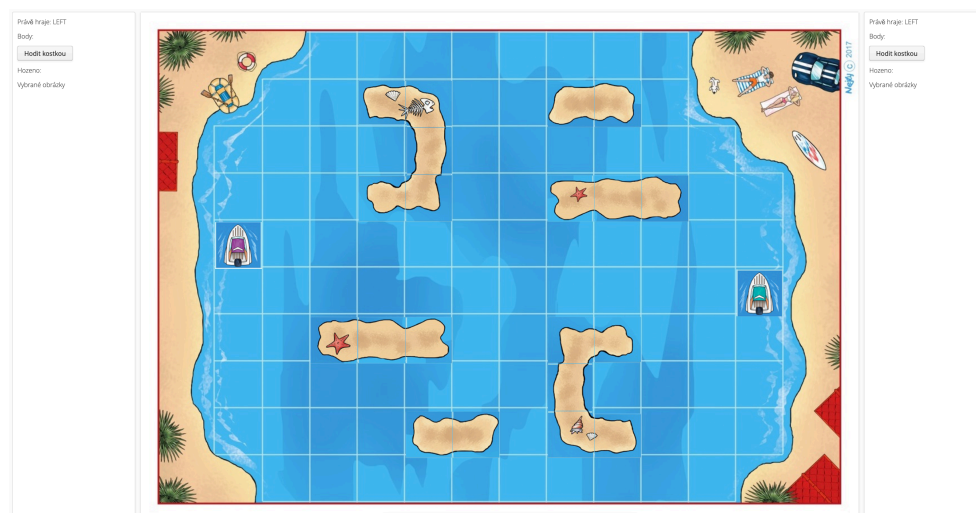
7. IMPLEMENTACE

bude moci ve hře každý hráč používat a na závěr zdali chce, aby byla hra v češtině nebo v angličtině, jak můžeme vidět na obrázku 7.1. Poté co uživatel zvolí všechny tyto požadavky a potvrdí tlačítkem, zobrazí se mu zbylá část hry.



Obrázek 7.1: Možnosti výběru mapy, módu a jazyka

Po stisknutí potvrzovacího tlačítka se zobrazí tři panely samotné hry, a to panel levého hráče, panel herní plochy a panel pravého hráče. Oba hráči mají ve svých panelech informace o tom, kdo je zrovna na tahu, dále kolik bodů již nasbíral hráč vlastníci daný panel, poté tlačítko, které u každého hráče slouží jako hod hrací kostkou, pod nímž je zobrazeno kolik bylo hozeno a na závěr seznam políček, které už daný hráč navštívil a získal. Panel herní plochy se skládá z herní mapy a políček, které jsou na ní položené. Vše můžeme vidět na obrázku 7.2.



Obrázek 7.2: Část hry zobrazená po potvrzení výběru

Všechny výše popsané části grafického rozhraní se nazývají komponenty a jsou to vytvořené objekty v jazyce Java, které Vaadin framework překládá do

jazyku JavaScript, a proto jsme schopni je takto vidět v prohlížeči. Například komponent tlačítko se v Javě vytvoří takto:

```
Button názevTlačítka = new Button("TextNaTlačítku");
```

Výpis kódu 1: Vytvoření tlačíka v Javě

Button je třída tlačítka, *názevTlačítka* je název nově vytvořeného objektu a parametr *TextNaTlačítku* je text promítnutý na tlačítku při zobrazení v grafickém rozhraní.

7.2.1 Rozlišení prohlížeče

Nemalým problémem při implementaci aplikací bývá rozlišení monitoru nebo velikost okna prohlížeče. Při vytváření Lovců žraloků byl na tento problém kladen důraz a povedlo se ho téměř vyřešit. Aplikace při spuštění získá údaje o šířce a výšce okna prohlížeče, ve kterém je spuštěna a podle nich poté vypočítá rozměry všech komponent, které zobrazuje. Hlavním údajem je šířka okna prohlížeče, od které se poté vypočítá výška hráčských panelů a panelu herního, což ale vede k jednomu problému. Pokud by uživatel vlastnil monitor s poměrem stran 2:1 nebo více¹¹, mohl by si hru zobrazit celou jen v případě, že by zmenšil šířku okna prohlížeče.

Samozřejmě má aplikace také omezené rozlišení. Spodní hranice je nastavena na 1200x800, kdy pokud prohlížeč má menší rozlišení, hra se již nebude zmenšovat a zůstane na velikosti spočítané ze zmíněných 1200x800. Naopak horní hranice není nastavena, ale je třeba brát v úvahu, že grafika dodaná k vývoji hry má určité rozlišení a při zobrazování hry na velkých monitorech může nastat situace, kdy jednotlivé obrázky budou nekvalitní.

7.3 Propojení frontendu a backendu

V aplikaci jsou vytvořené dvě hlavní třídy, první - *MainView* - je primární třídou grafického rozhraní, neboli frontendu a druhá - *GameController* - je takzvanou řídicí jednotkou logické části hry, neboli backendu. Právě tyto dva zástupci slouží ke komunikaci mezi frontendem a backendem. Takový vývoj aplikací je velmi důležitý, protože se může stát, že v budoucnu bude potřeba například změnit uživatelské rozhraní a s tímto návrhem nebude potřeba zasahovat do jiných částí backendu než do třídy *GameController*.

7.4 Řídicí třída logické části hry

Jak bylo zmíněno v předešlé kapitole, hlavní řídicí jednotkou logické části hry je třída *GameController*. Tato třída ovládá tvorbu a údržbu mapy, dále

¹¹Šířka ku výšce, tedy například ultraširoké monitory, které mají poměr stran až 32:9

kontroluje hráče a jejich pohyb po mapě, ale především je tato třída takový „kontejner“ na objekty tříd backendu. Při spuštění hry si GameController ve svém těle udržuje objekty dvou hráčů, pokud je zvolený herní mód hráč proti hráči, dále objekt mapy a také objekt třídy Movement, která zařizuje pohyb hráčů po mapě.

7.5 Políčka

Abychom mohli popsat zbylé části backendu, je nutné nejdříve znát třídu Tile, která vytváří jednotlivá políčka a z těch se poté skládá herní plocha. Druhů všech políček je ve hře celkem pět - *ostrov*, *loď*, *ryba*, *žralok* a *bublíny* - ovšem protože herní plochu tvoří tabulka 10x12 políček a na herní mapě některá rohová políčka chybí, byl přidán typ políčka *zakázaný*. Tento typ jako jediný nepředstavuje ve hře využitelná políčka, a proto nemá žádný obrázek jako pozadí. Jako poslední byl přidán typ políčka *nespecifikovaný*, kterým se při inicializaci vyplní téměř celá mapa a poté se tyto políčka přepisují na jiná. Všechny druhy políček jsou v aplikaci zaznamenány výčtovým typem.

```
enum TileType {  
    UNSPECIFIED, FORBIDDEN, ISLAND, FISH, SHARK, BUBBLES, SHIP  
}
```

Výpis kódu 2: Výčtový typ políček

Třída Tile obsahuje tři různé konstruktory, které lze přetížít podle toho, jaký typ políčka je nutné inicializovat. Konstruktor s jedním parametrem slouží k inicializování políček *zakázaný* a *nenalezený*, konstruktor s dvěma parametry inicializuje políčka typu *ryba*, *žralok*, *bublína* nebo *loď*, kde druhý parametr udává počet nebo barvu konkrétního typu, a nakonec konstruktor se čtyřmi parametry pomocí něhož se vytvářejí políčka typu *ostrov*, který potřebuje znát velikost, orientaci se vzhledem a část ostrova.

Do každého objektu třídy Tile se při inicializaci uloží cesta ke konkrétnímu obrázku, který se poté zobrazuje v grafickém rozhraní. Například pokud se vytváří objekt typu *loď* a konstruktor dostane parametr s barvou lodě, pak je obrázek přiřazen takto:

```
this.originalImage =  
new ExternalResource("VAADIN/images/entities/ship_"+type+".png");
```

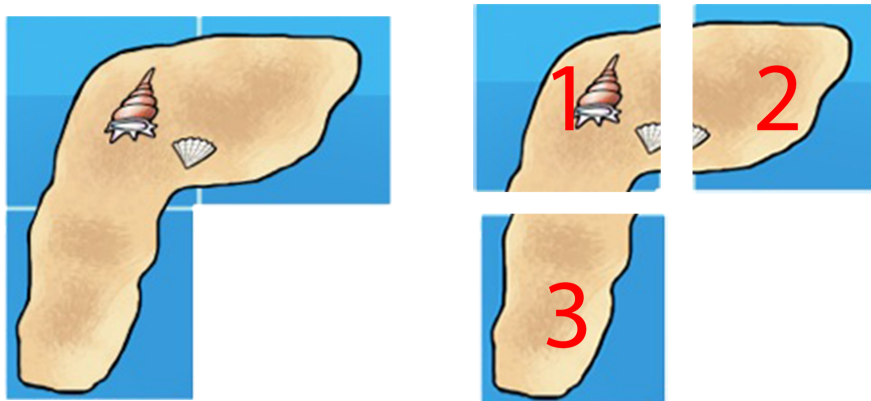
Výpis kódu 3: Přiřazení obrázku k objektu třídy Tile

Kořenová složka třídy ExternalResource je „webapp“, v této složce je další s názvem „VAADIN“, která obsahuje všechny externí zdroje jako jsou právě obrázky nebo například kaskádové styly, jak jsme již mohli vidět na obrázku 6.1.

7.6 Mapa

Nejspíše nejsložitější částí celé aplikace je třída `Map`. Tato třída řídí vše, co se herní plochy týče, ať už položení ostrovů, náhodné rozmístění skrytých kartiček nebo generaci celých map. Samotná herní plocha je ve třídě reprezentována jako dvourozměrné pole objektů třídy `Tile`. Hra zatím obsahuje celkem tři mapy, jejichž odlišnost je vidět při generování ostrovů.

Ostrovů je několik typů, které se liší velikostí a tvarem a každý z těchto typů se ještě dělí na čtyři podtypy, ve kterých se ostrovy odlišují jen vzhledem a orientací, nikoliv tvarem ani velikostí. Samotný způsob generování ostrovů u různých map není odlišný. Odlišný je postup výběru, jaký typ ostrova bude vytvořen. Každý ostrov je tvořený tak, že se vytvoří nový objekt třídy `Tile` s parametry velikosti, orientace a částí ostrova. Část ostrova je nutné znát, protože každý objekt třídy `Tile` obsadí pouze jedno políčko na herní ploše, ale většina ostrovů se skládá z více políček. Řešením takového problému je rozdělení každého ostrova na určitá políčka, jak můžeme vidět na obrázku 7.3.



Obrázek 7.3: Rozdělení ostrova typu 4 na jednotlivá políčka

Na výše zmíněném obrázku 7.3 je znázorněn ostrov typu čtyři s orientací a vzhledem typu jedna. Kdybychom tedy chtěli inicializovat políčko první části tohoto ostrova, bude kód v programu vypadat takto:

```
Tile ostrov = new Tile(4, 1, 1);
```

Výpis kódu 4: Inicializace ostrova typu 4 a podtypu 1, první část

7.6.1 Zrcadlová

Tato mapa, jak už bylo zmíněno v kapitole 4.4.3, se snaží být co nejvíce vyvážená vůči oběma hráčům. Proto jsou vytvořeny dva ostrovy typu pět na předem určených souřadnicích, každý v jedné čtvrtině mapy, první v levé horní části, druhý v pravé dolní části. Ve zbylých dvou čtvrtinách jsou ostrovy generovány částečně náhodně. Vygeneruje se typ ostrova v rozmezí jedna až čtyři s náhodným vzhledem, umístí se na předem určené souřadnice a poté se ve druhé zbývající čtvrtině vytvoří stejný typ ostrova s náhodným vzhledem, také na předem určených souřadnicích. Tento postup se opakuje ještě jednou, aby na mapě bylo celkem šest ostrovů. Celá tato herní plocha je generována pouze za pomoci lokálního generátoru náhodných čísel¹² a předem určených souřadnic, nikoliv podpůrné třídy RandomGenerator, kterou používají další dvě mapy.

7.6.2 Náhodná

Kompletně náhodně generovaná mapa je nejspíše operačně nejnáročnější. Umístění ostrovů na herní plochu zařizuje podpůrná třída RandomGenerator, jejíž inicializovaný objekt je uložený ve třídě Map. V metodě SetUpIslands třídy Map se šestkrát provede cyklus, ve kterém se náhodně vygeneruje typ a podtyp ostrova a tyto údaje se předají funkci GenerateIslandPosition jako parametry. Výše popsaný kód můžeme vidět ve výpisu 5.

```
for(int i = 0; i < _NUMBER_OF_ISLANDS; i++)
{
    // generovani typu a podtypu
    int t = ThreadLocalRandom.current().nextInt(1,6);
    int s = ThreadLocalRandom.current().nextInt(1,5);
    // metoda RandomGenerator vracejici pozici 1. casti ostrova
    Pair<Integer, Integer> position
    = randomGenerator.GenerateIslandPosition(tiles, t, s);
    // metoda pro polozeni ostrova na hraci plochu
    PlaceIsland(position.getKey(), position.getValue(), t, s);
}
```

Výpis kódu 5: Generování náhodné mapy v metodě SetUpIslands

Nyní se dění přesune do třídy RandomGenerator, kde se nejdříve rozhodne, o jaký typ ostrova se jedná a poté se v cyklu volá daná metoda, dokud nevrátí hodnotu true. Daná metoda je pro každý typ ostrova specifická, jako parametry bere náhodně vygenerované souřadnice v cyklu, ze kterého je volána, a samotná metoda každého typu kontroluje, zdali jsou všechna políčka, které

¹²z Java knihovny java.util.concurrent.ThreadLocalRandom

ostrov potřebuje, dostupná. Zároveň metoda kontroluje, zdali jsou dostupná políčka, která by ležela okolo ostrova, pokud by na takových souřadnicích bylo místo. Tímto se docílí toho, že na herní ploše nebudou ostrovy natěsnané přímo u sebe, ale vždy mezi nimi bude alespoň jedno políčko volné.

```
while(!CheckTypeFour(tiles, x, y, subtype))
{
    x = ThreadLocalRandom.current().nextInt(1,8);
    y = ThreadLocalRandom.current().nextInt(2,10);
}
return new Pair<>(x, y);
```

Výpis kódu 6: Generování souřadnic 1. části ostrova typu 4

Právě kód 6. znázorněný výše je důvodem, proč je náhodná mapa operačně nejnáročnější. Výběr pozice probíhá náhodně, poté se kontroluje, zdali by se ostrov mohl na tyto souřadnice umístit a pokud tomu tak není, celý proces se opakuje. V nejhorších případech může cyklus provést mnohem více kroků než v průměrném případě. Aby k takovým případům vůbec docházelo, musely by se například generovat jen ostrovy pátého typu, a to je při osazování šesti ostrovy pravděpodobnost 0,000064%. To je poměrně zanedbatelné číslo v poměru s výkonem dnešních počítačů a při osobním testování nebyly spatřeny žádné pomalejší výsledky oproti ostatním mapám.

7.6.3 Se středním ostrovem

Tento druh mapy kombinuje prvky obou předchozích. Nejdříve se uprostřed mapy vygenerují dva ostrovy pátého typu tak, aby uvnitř vznikly dvě prázdná políčka. Tyto ostrovy mají dopředu zvolené souřadnice a orientaci a generuje se pouze jejich vzhled. Dále se mapa rozdělí na čtvrtiny a v každé z nich se vygeneruje náhodný ostrov typu jedna až čtyři. Tím vznikne na herní ploše celkem šest ostrovů. Ostrovy ležící v různých čtvrtinách mapy se opět generují pomocnou třídou RandomGenerator.

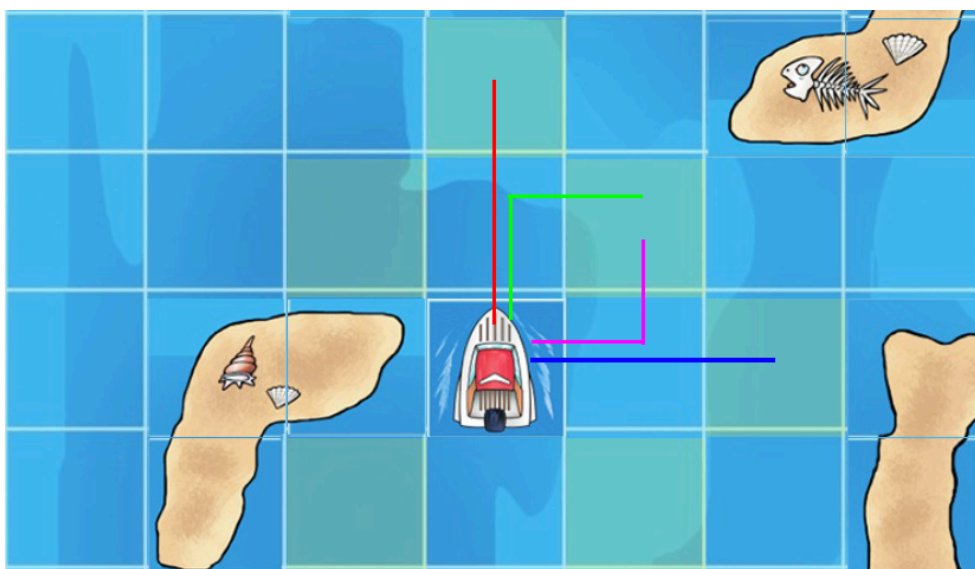
7.7 Třídy hráčů a pohybu

Třída Player uchovává všechny informace každého hráče, jako jsou pozice lodí, počet lodí nebo počet nachytaných ryb. Také si každý z hráčů uchovává adresy na objekty políček, na kterých zrovna stojí jejich loď. Toto je nutné, protože na každém políčku herní mapy může být v jednu chvíli jen jeden objekt a kdyby si hráči neukládali odkazy na objekty, které loď nahradila, tak by nebyla možnost po jejím přeplutí na jiné políčko nahrazený objekt zobrazit.

Pohyb lodí po mapě řídí třída Movement, která nejdříve označí dostupná políčka, a poté co hráč vybere jedno z nich, loď na dané políčko přesune.

7. IMPLEMENTACE

Označení probíhá na základě hodu kostkou. Označí se všechna dostupná políčka v okruhu lodě, který je vypočítán podle výsledku hodu. Například pokud hráč hodí dva, označí se políčko ve vzdálenosti dva nad hráčova lodí, poté ve vzdálenosti jedna nad lodí a ve vzdálenosti jedna vpravo od lodí a tak dále, dokud se neoznačí políčka v celém okruhu, jak můžeme vidět na obrázku 7.4.



Obrázek 7.4: Ukázka označení políček při hodu 2 hrací kostkou

Poté co hráč zvolí, na jaké políčko se má loď přesunout, třída `Movement` navrátí na původní pozici objekt, který v ní byl uložen předtím, než pozici obsadila loď, a také zobrazí původnímu objektu jeho obrázek, aby hráči viděli, co bylo uloveno. Dále třída `Movement` hráči, který pluje s lodí, uloží odkaz na objekt ležící na políčku, na který zrovna loď míří, a poté loď na dané políčko přesune.

7.8 Umělá inteligence

Při implementování umělé inteligence byl kladen důraz na to, aby hráč i počítač měli stejné podmínky. Uživatel neví, co se skrývá pod políčky, na která loď posílá, takže to nesmí vědět ani hráč, kterého řídí aplikace. To ale znamená, že z umělé inteligence se spíše stává obyčejný generátor náhodných čísel, neboť jediné co dělá je, že náhodně vygeneruje číslo políčka, na které se poté přesune.

Testování

V této kapitole budou shrnuty výsledky testování finální verze aplikace. Testování je rozděleno na automatické a uživatelské, ale docházelo především k uživatelskému.

8.1 Automatické

Převážná část aplikace vykonává svou funkci na základě výsledku nějaké náhody, tudíž je téměř nemožné vytvořit automatické testy. Od vytváření mapy, kdy se všechna políčka generují náhodně, přes hod kostkou až k výběru pozice, na kterou loď popluje, vše je buďto náhodné nebo závislé na uživateli. Proto obecné automatické testy jako JUnit nebudou fungovat, protože je pro ověření nutné znát výsledek testu dopředu, aby program mohl ověřit, zda zkoušené části programu vracejí stejné výsledky jako je ten předem známý.

8.2 Uživatelské

Protože vytvořit automatické testy na tento typ aplikace je velice obtížné, až nemožné, bylo testování prováděno primárně lidmi. Základním pozorovacím aspektem byla celková funkčnost a správné chování aplikace, poté byl při testování kladen důraz na volbu prohlížeče a rozlišení obrazovky nebo velikost okna prohlížeče.

Již při implementaci celé aplikace byla hra důkladně ověřována, tudíž při testování finální verze nebylo nalezeno mnoho chyb, co se funkčnosti týče. Testy také probíhali na čtyřech nejrozšířenějších prohlížečích - Google Chrome, Safari, Opera a Mozilla Firefox. Nejlepší výsledky ukazoval prohlížeč Google Chrome, na kterém aplikace fungovala bezchybně, nejspíše z důvodu, že hra byla s pomocí tohoto prohlížeče vyvíjena. Hlavní nedostatek u Safari, Opery i Mozilly byla plynulost popojíždění¹³ po stránce aplikace, nicméně kvalitu

¹³Popojíždění nahoru a dolů po stránce prohlížeče - v angličtině scroll

8. TESTOVÁNÍ

hry to neovlivnilo, neboť popojet je nutné jen po potvrzení výběru nastavení, a to při velkém poměru šířky ku výšce prohlížeče.

Samotné změny v rozlišení nebo velikosti prohlížeče neobjevily žádné nové chyby. Hra se chovala přesně tak, jak bylo popsáno v kapitole 7.2.

Budoucí rozvoj

Při vývoji této aplikace bylo navrženo mnoho nápadů na úpravy, různé vylepšení a rozšíření hry, které bohužel nemohou být součástí této bakalářské práce a budou tedy v této kapitole uvedeny jako možnosti budoucího rozvoje.

9.1 Hra více hráčů

Nejambicióznější, ale zároveň nejpracnější rozšíření Lovců žraloků by byla hra více hráčů na serveru. Bylo by nutné implementovat samotný server, který by se staral o připojování a komunikaci uživatelů s aplikací. Dále by nejspíše bylo nutné zvětšit herní plochu, neboť nynější velikost je vhodná pro dva hráče. Ke zvětšení mapy by byla zapotřebí nová grafika od Petra Netíka a také nové druhy ostrovů.

9.1.1 Hra v týmech

Pokud by existovalo ve hře rozšíření o více hráčů na serveru, bylo by vhodné implementovat nový mód hry v týmech. Před vygenerováním mapy by si hráči buďto zvolili s kým by chtěli být v týmu, nebo by byli rozřazeni automaticky, a poté by se vygenerovala mapa. Hráčům ve stejném týmu by se počet natchytaných ryb sčítal a vyhrál by tým, který by na konci hry měl nejvíce ryb, jako je tomu u nynější verze.

9.2 Nové mapy a herní módy

Dalším přínosem do hry by mohly být nové mapy, ať už ve smyslu generování ostrovů nebo přímo nové grafické návrhy od Petra Netíka, a také herní módy, jako například výše zmíněná hra v týmech.

9.3 Statistiky a žebříčky

Důležitou součástí her pro více hráčů jsou žebříčky a statistiky, které do nich přináší kompetitivní prvky a na jejich základě nutí hráče více dané hry hrát. Statistiky by mohly například ukazovat kolik bylo za existenci hry nasbíráno ryb, kolik žraloků, kolik ryb v jednom kole nejvíce získal jeden hráč, kolik bylo nejvíce v jednom kole pořízeno lodí navíc, a tak dále.

Dokumentace

Jeden z požadavků v zadání je, aby aplikace byla umístěna na aplikační server, na kterém bude spuštěna. Ke spuštění hry je tedy potřeba vlastnit zařízení s prohlížečem a internetovým připojením. Poté stačí do prohlížeče zadat adresu <http://185.88.73.72:8080/Sharks/>, na které se spuštěná hra nachází a zábava může začít.

10.1 Prostředí

Po otevření aplikace v prohlížeči se nejdříve objeví možnosti nastavení. První položkou zleva je volba mapy, kterou chceme vygenerovat, dále můžeme vybrat jeden z herních módů, také můžeme tlačítkem přepnout volbu jazyka mezi češtinou a angličtinou a jako poslední položka zde leží tlačítko Ukončit hru, které slouží k umělému ukončení a oznámení vítěze. Poté, co máme zvolený typ mapy a herní mód, výběr potvrdíme tlačítkem vybrat a vygeneruje se nám zbylá část aplikace. Tato část aplikace se skládá ze tří panelů, kdy dva slouží jako informační panely pro hráče a třetí jako herní plocha.

10.2 Ovládání

Ovládání celé aplikace je velmi jednoduché. Není za potřebí používat klávesnici, hra se ovládá pouze pomocí myši na zařízeních, které myš mají, nebo pomocí dotyku na dotykových zařízeních.

Po vygenerování herní plochy má každý hráč svoji kostku, se kterou v jeho tahu háže. Hod provede stisknutím tlačítka Hodit kostkou, které najde ve svém informačním panelu. Po hosení kostkou se aktuálnímu hráči zobrazí dostupná políčka ve vzdálenosti výsledku hodu od jeho lodi. Dále hráč zvolí nějaké z dostupných políček a loď se na něj posune. Pokud aktuální hráč plul se všemi svými loděmi, je na řadě hráč druhý.

10. DOKUMENTACE

Hra se ukončí a vyhlásí vítěze ve chvíli, kdy na herní ploše již nejsou k dispozici žádné ryby k chytání, nebo když jsou hráči od ryb ve vzdálenosti takové, že už není možnost se k nim dostat. Ovšem takové situace nastávají málokdy a bývají poměrně nudné, takže bylo do hry přidáno tlačítko pro ukončení, kdy se oba hráči shodnou a po stisknutí tlačítka hra vyhlásí vítěze.

Závěr

Cílem bakalářské práce bylo vytvořit počítačovou verzi vybrané stolní hry, a to se všemi prvky, které samotná stolní hra nabízí. Tento cíl se podařilo zcela splnit, a dokonce i předčit, jelikož počítačová verze obsahuje všechny funkce jako její předloha a je obohacena o spoustu dalších. Podařilo se implementovat hru ve Vaadin frameworku tak, aby se přizpůsobovala velikosti rozlišení a okna prohlížeče, ve kterém je spuštěna. Dále byly úspěšně přidány dvě nové mapy. K náhodné mapě je možné ve hře vybrat ještě zrcadlovou, která se snaží rozmístit ostrovy tak, aby oba hráči měli téměř totožné podmínky, a poté ještě mapu s prostředním ostrovem, ve kterém jsou dvě políčka, kde na jednom je deset ryb a na druhém osm žraloků. Tím vznikne ve hře možnost, jak rychle získat velký počet ryb, ale také možnost, jak velký počet ryb ztratit. Dalším vylepšením oproti stolní předloze jsou herní módy. Ty přibyly dva, a to hra proti počítači a hra více lodí, ve které má každý hráč schopnost přikoupit si druhý člun. I když hra neobsahuje mnoho textu, je další výhodou to, že jazyk může být přepnut jak do češtiny, tak do angličtiny.

Závěrem tedy je bakalářská práce splňující celé zadání a obohacující ho o mnoho prvků. Byla vybrána hra, která byla analyzována a na základě analýzy byly nastíněny úpravy a různé vylepšení, a také vybrány vhodné herní módy. Po shrnutí všech požadavků byl navržen UML diagram, podle kterého byla poté naprogramována funkční aplikace. Tato aplikace byla implementována v souladu s pravidly Java EE, výsledná hra byla umístěna na aplikační server s adresou <http://185.88.73.72:8080/Sharks/> a v závěru praktické části byly navrženy možnosti na budoucí rozšíření.

Tvorba této bakalářské práce přinesla i velký osobní rozvoj. Nejdříve hraní a v posledních letech i vytváření počítačových her jsou mou velkou zálibou a není žádným tajemstvím, že vývojem her bych se chtěl živit. To je také pádný důvod, proč jsem si zvolil toto téma bakalářské práce. Značný přínos jsem pocítil jak při návrhu, tak při implementaci dané hry. Ujistil jsem se, že nejlepší zbraní dobrého programátora je schopnost hledat informace a učit se novým věcem, díky čemuž jsem si osvojil nové technologie, ve kterých vidím

ZÁVĚR

velký potenciál i v budoucnu. Také jsem si připomenul, že je velmi důležité nejdříve aplikaci důkladně promyslet a navrhnout a až poté jí začít implementovat, protože se tak vyhnu mnoho potížím při programování. Všechny tyto znalosti jsem přitom získal při velmi zábavné formě tvorby aplikace a velice jsem si vývoj užíval.

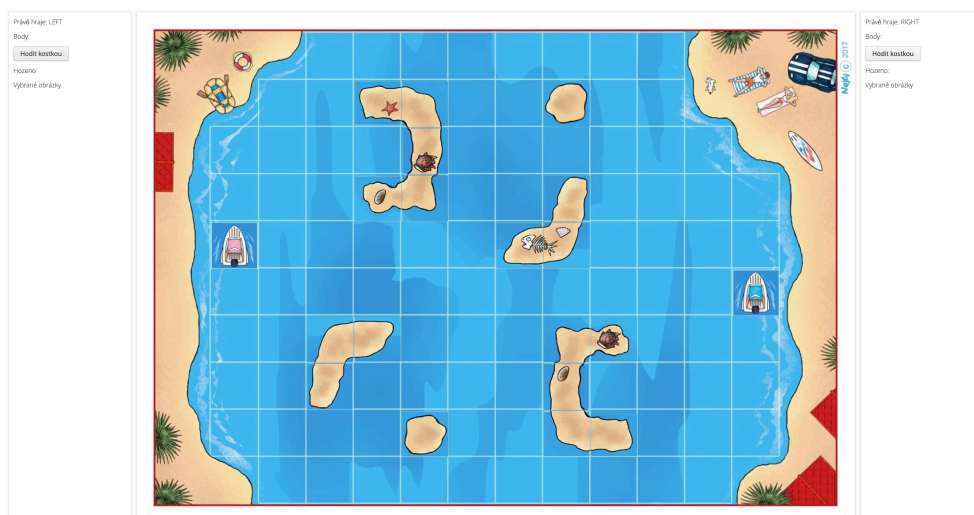
Bibliografie

1. EDITORS, History.com. Video Game History. *History* [[online]]. 2019. Dostupné také z: <https://www.history.com/topics/inventions/history-of-video-games>. [cit. 2019-04-06].
2. FRASER, Mick. The 10 best-selling video game franchises ever. *RedBull* [[online]]. 2016. Dostupné také z: <https://www.redbull.com/gb-en/10-best-selling-video-game-franchises-of-all-time>. [cit. 2019-05-03].
3. NICHOLS, Hannah. How video games affect the brain. *Medical News Today* [[online]]. 2017. Dostupné také z: <https://www.medicalnewstoday.com/articles/318345.php>. [cit. 2019-04-06].
4. SANDRO FRANCESCHINI Simone Gori, Milena Ruffino. Action Video Games Make Dyslexic Children Read Better. *Current Biology* [[online]]. 2013. Dostupné také z: [https://www.cell.com/current-biology/fulltext/S0960-9822\(13\)00079-1?_return](https://www.cell.com/current-biology/fulltext/S0960-9822(13)00079-1?_return). [cit. 2019-04-06].
5. NAKAMURA, Yuji. Peak Video Game? Top Analyst Sees Industry Slumping in 2019. *Bloomberg* [[online]]. 2019. Dostupné také z: <https://www.bloomberg.com/news/articles/2019-01-23/peak-video-game-top-analyst-sees-industry-slumping-in-2019>. [cit. 2019-05-03].
6. MCNARY, Dave. 2018 Worldwide Box Office Hits Record as Disney Dominates. *Variety* [[online]]. 2019. Dostupné také z: <https://variety.com/2019/film/news/box-office-record-disney-dominates-1203098075/>. [cit. 2019-05-03].
7. SAM LOVERIDGE, Ford James. How many people play Fortnite? Is it really as many as people say? *Games Radar* [[online]]. 2019. Dostupné také z: <https://www.gamesradar.com/how-many-people-play-fortnite/>. [cit. 2019-05-03].

8. ARCHER, John. Netflix Reveals That Its Biggest Threat Is... 'Fortnite'. *Forbes* [[online]]. 2019. Dostupné také z: <https://www.forbes.com/sites/johnarcher/2019/01/17/netflix-reveals-that-its-biggest-threat-is-fortnite/>. [cit. 2019-05-03].
9. SCHÖN, Otakar. Větší než film i s televizí: České hry baví desítky milionů lidí a vydělávají miliardy korun. *Hospodářské Noviny* [[online]]. 2019. Dostupné také z: <https://tech.ihned.cz/hry/c1-66521470-vsechno-na-export-videohry-v-cesku-zamestnavaji-jen-1500-lidi-loni-vynesly-tri-miliardy-korun>. [cit. 2019-05-03].
10. BRANTON. How much should you spend on a gaming PC? *PC Game Heaven* [[online]]. 2019. Dostupné také z: <https://pcgamehaven.com/how-much-should-you-spend-gaming-pc/>. [cit. 2019-04-13].
11. SLAVIK, Nathan. The Future of Mobile Gaming: Where we started and where we're going. *App Lovin* [[online]]. 2018. Dostupné také z: <https://blog.applovin.com/applovin-future-of-mobile-gaming-part-1/>. [cit. 2019-04-13].
12. HERMANA, Ridz. online chess game (Class Diagram (UML)). *Creately* [[online]]. Dostupné také z: <https://creately.com/diagram/example/hu3z6jqg1/online%5C%20chess%5C%20game%5C%20>. [cit. 2019-04-16].
13. IQBAL, Zainab. ludo game (Class Diagram (UML)). *Creately* [[online]]. Dostupné také z: <https://creately.com/diagram/example/htfxac102/ludo%5C%20game>. [cit. 2019-04-16].
14. PAVLÍČEK, Josef. Pexeso. *athena.pef.czu* [[online]]. Dostupné také z: <http://athena.pef.czu.cz:8080/Pairs/>. [cit. 2019-04-16].
15. GARBADE, Dr. Michael J. Top 3 most popular programming languages in 2018 (and their annual salaries). *Hackernoon* [[online]]. 2018. Dostupné také z: <https://hackernoon.com/top-3-most-popular-programming-languages-in-2018-and-their-annual-salaries-51b4a7354e06>. [cit. 2019-04-27].
16. ADMIN. JAVAONE : Sun: The bulk of Java is open sourced. *GrnLight* [[online]]. 2014. Dostupné také z: <https://web.archive.org/web/20140527220942/http://grnlight.net/index.php/programming-articles/115-javaone-sun-the-bulk-of-java-is-open-sourced>. [cit. 2019-04-27].
17. GUPTA, Kitty. WHAT IS THE DIFFERENCE BETWEEN JAVA SE AND JAVA EE? *Freelancing Gig* [[online]]. 2017. Dostupné také z: <https://www.freelancinggig.com/blog/2017/04/05/difference-java-se-java-ee/>. [cit. 2019-04-27].

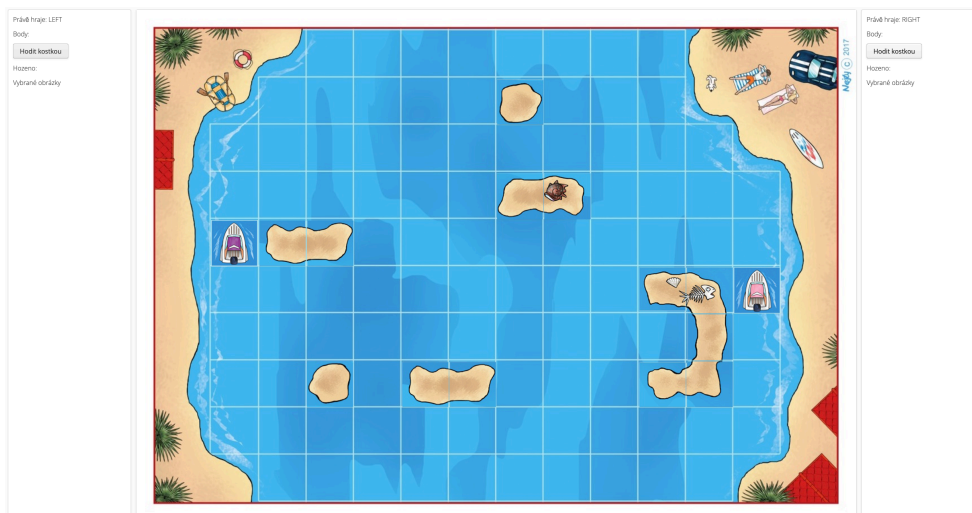
18. PERRY, J. Steven. What is Vaadin? A faster approach to Java web applications. *IBM* [[online]]. 2017. Dostupné také z: <https://developer.ibm.com/dwblog/2017/what-is-vaadin-java-web-applications/>. [cit. 2019-04-29].
19. SEPPÄNEN, Juha. Vaadin Framework 8 roadmap 2017 and beyond. *Vaadin* [[online]]. 2017. Dostupné také z: <https://vaadin.com/blog/vaadin-framework-8-roadmap-2017-and-beyond>. [cit. 2019-04-27].

Obrazová příloha

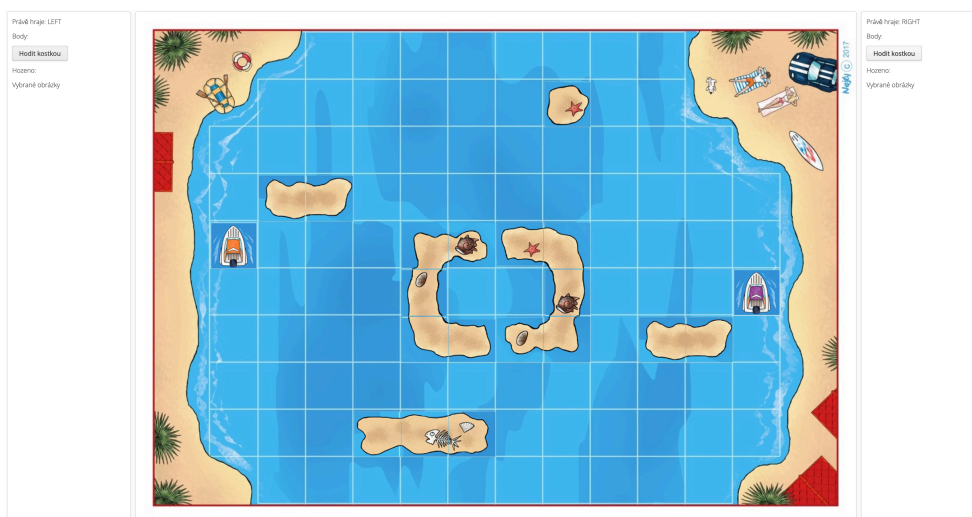


Obrázek A.1: Hrací plocha po vygenerování zrcadlové mapy

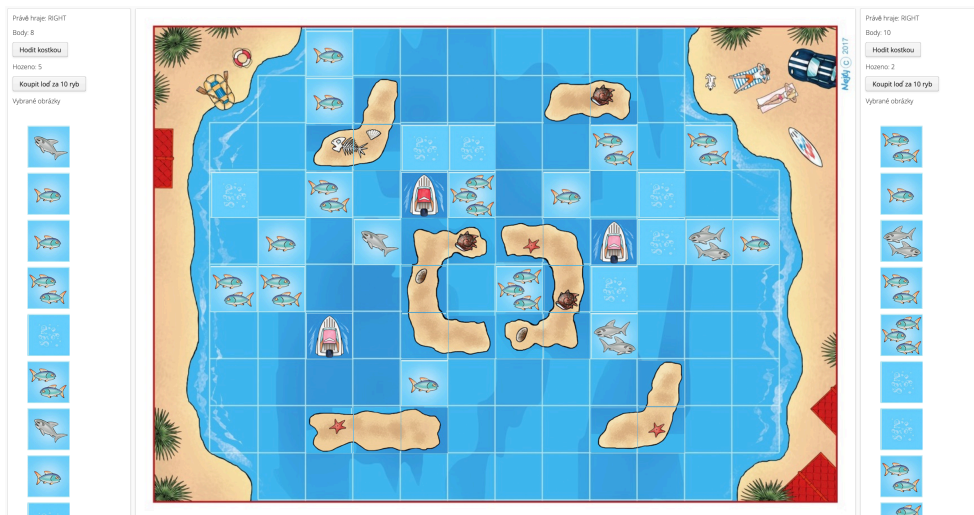
A. OBRAZOVÁ PŘÍLOHA



Obrázek A.2: Hrací plocha po vygenerování náhodné mapy



Obrázek A.3: Hrací plocha po vygenerování mapy se středním ostrovem



Obrázek A.4: Hrací plocha přibližně ve střední části hry

Obsah přiloženého CD

	readme.txt	stručný popis obsahu CD
	src	
	impl	zdrojové kódy implementace
	thesis	zdrojová forma práce ve formátu \LaTeX
	text	text práce
	thesis.pdf	text práce ve formátu PDF