

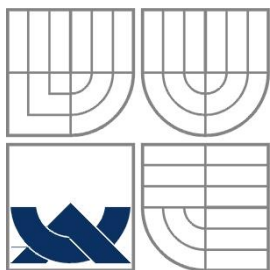
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

Fakulta informačních technologií
Faculty of Information Technology

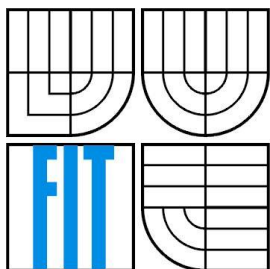
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

Brno, 2016

David Mikuláščík



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

TRANSFORMACE DESKOVÉ HRY MURBACH NA POČÍTAČOVOU HRU

THE COMPUTER GAME INSPIRED BY A DESK GAME

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

David Mikuláščík

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Martin Krčma

BRNO 2016

Abstrakt

Tato bakalářská práce se zabývá převodem deskové hry na počítačovou hru a její implementací. Shrnuje pravidla původní hry a rozebírá problematiku jejich převodu na pravidla počítačové hry. Popisuje praktické aspekty implementace a využití návrhových vzorů. Vyhodnocuje zpětnou vazbu a navrhuje další vývoj hry.

Abstract

This thesis deals with a transformation of a board game to a computer game and its implementation. It summarizes rules of original game and discusses about issues of transformation them to rules of the computer game. It describes practical aspects of implementation and usage of design patterns. It evaluates feedback and design further development of the game.

Klíčová slova

desková hra, návrhové vzory, počítačová hra, tahová strategie, C#, .NET Framework, Windows Presentation Foundation, XAML

Keywords

board game, design pattern, computer game, turn based strategy, C#, .NET Framework, Windows Presentation Foundation, XAML

Citace

Mikuláščík David: Transformace deskové hry Murbach na počítačovou hru, bakalářská práce, Brno, FIT VUT v Brně, 2016

Transformace deskové hry Murbach na počítačovou hru

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením ing. Martina Krčmy. Další informace mi poskytli členové skautského oddílu Dvojkaři. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jméno Příjmení David Mikuláščík
18.5.2016

Poděkování

Chtěl bych poděkovat Michalu Tulkovi za konzultace původní deskové hry. Vedoucímu bakalářské za neobvyklé věcné konzultace.

© David Mikuláščík, 2016

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

Obsah.....	1
1 Úvod	2
1.1 Původní hra	2
1.1.1 Pojmy a pravidla.....	2
1.1.2 Příprava a průběh hry.....	4
1.1.3 Hodnocení hry.....	4
1.1.4 Modelové situace.....	5
2 Návrh počítačové hry.....	6
2.1 Převod hry.....	6
2.1.1 Mapa hry a základní principy	6
2.1.2 Služba zasílání zpráv a diplomacie	6
2.1.3 Města a jejich založení.....	7
2.1.4 Vojska	10
3 Implementace.....	12
3.1 Návrhové vzory	12
3.1.1 MVVM: Model-View-ViewModel	13
3.1.2 Observer.....	13
3.1.3 Command	14
3.1.4 Repository.....	14
3.2 Návrh aplikace.....	15
3.2.1 Návrh databáze.....	16
3.2.2 Návrh uživatelského rozhraní	16
3.2.3 Návrh simulátoru	20
4 Zpětná vazba a plán vývoje hry	21
4.1 Plánovaná rozšíření.....	21
4.2 Rozšíření na základě zpětné vazby	22
4.2.1 Tvorba mapy	22
4.2.2 Vylepšení uživatelského rozhraní.....	23
4.2.3 Přidání hexagonální mapy	23
4.2.4 Přidání počítačem řízených hráčů	25
5 Závěr.....	26

1 Úvod

Cílem práce je převedení strategické deskové hry na počítačový program. Proces převodu hry je rozdělen na analýzu původní hry, vytvoření návrhu počítačové hry, vytvoření návrhu aplikace a její implementaci. Výsledná počítačová hra je porovnána s původní deskovou hrou a na základě získaných informací a zpětné uživatelské vazby jsou navrženy úpravy hry a další vývoj projektu. Návrh počítačové hry je rozdělen do iterace návrhu základní hry a návrhu rozšíření hry.

Převod deskové hry na počítačovou hru, předpokládá analýzu a zjištění stěžejních prvků a pravidel původní hry, jednou z nesporných výhod deskových her, je možnost modifikace pravidel hráči, na druhou stranu pravidla her, mohou být různými hráči pochopena různě. Na rozdíl od toho, počítačová hra musí mít danou konečnou množinu pravidel, které jsou implementovány do výsledné aplikace.

Práce je dělena do pěti kapitol, v první kapitole je řešen rozbor deskové hry na základě získaných informací. Druhá kapitola se zabývá návrhem počítačové hry a řeší problémy s tím spojené. Třetí kapitola popisuje návrh implementace aplikace za použití návrhových vzorů. Ve čtvrté kapitole je formou reakce na zpětnou vazbu rozebráno srovnání počítačové hry s původní deskovou hrou a je předložen nástin dalšího možného vývoje hry.

1.1 Původní hra

Hra byla vymyšlena jako skautská oddílová hra v brněnském skautském oddíle *Dvojkaři* [1]. Sestávala se z tvorby mapy, ustanovení pravidel hry a samotného hraní, kdy bylo v rámci domluvy upravit pravidla. Hra se hrála v kolech, kdy každý z hráčů připravil své akce. Akce byly v rámci kola vyhodnoceny v případě, kdy je připravili všichni hráči nebo se na vyhodnocení shodli. Pokud všichni hráči nemohli během kola provést své akce, dostali bonusové akce do následujícího kola, které byly vyhodnoceny přednostně s jistými omezeními. Pravidla původní hry se nedochovala v psané podobě a byla rekonstruována na základě popisu hráčů dané hry.

Název hry je inspirován benediktinským klášterem Murbach, který byl založen v 8. století našeho letopočtu ve Francii. Klášter disponoval značným majetkem v podobě tří měst a třiceti vesnic [6]. V roce 1789 během velké francouzské revoluce byl opuštěn. Moc soustředěná pod opatství byla inspirací pro vytvoření deskové hry založené na získávání vlivu a zakládání měst, pro zpestření herního zážitku, byla ve hře přidána vojenská složka.

1.1.1 Pojmy a pravidla

Kolo je iterace hry, při jehož vyhodnocení jsou provedeny změny v herním světě a na mapě hry.

Mapa světa je složena z polí různých typů, která jsou uvedena v Tabulka 1.1, reprezentujících reálný svět. Hráč je účastníkem hry a provádí následující úkony:

- Spravuje svá města
- Zakládá nová města
- Verbuje a spravuje vojska
- Pohybuje vojsky na herní mapě

Typ pole	Pohyb vojska	Poznámka
Voda	Ne	Reprezentuje řeky a vodní plochy
Hory	Ano	Omezuje vzdálenost akce pohybu vojska v rámci kola, kdy je vojsko přechází
Louky	Ano	Pole poskytuje městu, pod jehož je vlivem jednotky jídla
Lesy	Ano	Pole poskytuje městu, pod jehož je vlivem jednotky dřeva
Vysoké hory	Ne	Pole poskytuje městu, pod jehož je vlivem jednotky kovu

Tabulka 1.1 Typy polí na mapě

Města jsou stěžejním prvkem celé hry. Město je umístěno na mapě hry a zabírá jedno herní pole, je prostředkem hráče pomoci, kterého rozšiřuje zónu svého vlivu, získává suroviny, verbuje a vylepšuje vojska. Jsou rozšiřována pomocí surovin, které utrží z vlastní produkce. Pole pod vlivem města poskytují městu za kolo suroviny, každé pole vždy jednu jednotku jemu odpovídající suroviny, pokud ji poskytuje.

Vojska jsou dalším stěžejním prvkem hry. Vojsko se nachází na jednom poli na mapě hry a může být přemístěno na jiné pole. Vojsko se může pohybovat do určité vzdálenosti v jednom kole. Vojsko může zakládat nová města. Vojsko může útočit na jiná vojska. Vojsko může útočit na města a obsazovat je. Vojsko získává od města, ve kterém bylo verbováno, bonusové body do útoku a obrany, pokud město provedlo příslušné vylepšení armády. Tabulka 1.2 udává předpokládaná vylepšení vojsk v deskové hře.

Vylepšení	Cena
+ 1 bod do útoku	3 jednotky jídla, 2 jednotky kovu
+ 1 bod do útoku	3 jednotky jídla, 3 jednotky kovu

Tabulka 1.2 Vylepšení armády

Diplomacie určuje a vymezuje herní vztahy mezi hráči ve hře. Hráči během hry vzájemně domlouvají spojenecké smlouvy, dohody neútočení, mírové dohody a vyhlašují války. Tabulka 1.3 popisuje základní diplomatické vztahy a smlouvy s jejich nastolením spojené.

Vztah	Smlouva	Veřejnost
Spojenecký	Spojenecká dohoda	Veřejné
Neutrální	dohoda o neútočení, mírová dohoda	Veřejné i neveřejné
Nepřátelský	Vyhlášení války	Veřejné

Tabulka 1.3 Diplomacie a smlouvy

Desková hra umožňuje hráčům uzavírat i jiné dohody mezi hráči a pomocí nich manipulovat s rozhodnutími ostatních hráčů. Všechny smlouvy nejsou podchyceny striktními pravidly a dávají prostor pro improvizaci. U deskové hry se mezi hráči mohou sdělovat porušení dohod a smluv, což může mít za následek, že ostatní hráči nebudou danému hráči věřit a nebudou s ním uzavírat dohody, případně většina hráčů může danému hráči vyhlásit válku a zkusit jej eliminovat, pravidla hry nepostihují věrolomné chování hráčů, ani neodměňují čestné chování formou hodnocení.

1.1.2 Příprava a průběh hry

Hráči je vytvořena mapa herního světa v papírové podobě, je tvořena čtvercovou sítí polí. Každý hráč určí, kde bude umístěno jeho první město, zakreslí je do mapy a dostane do něj jednu armádu o síle 10 jednotek.

Kolo hry začíná akcí města, kdy každý hráč určí přerozdělení jednotek z produkce na rozvoj města, armád a zadá verbování armády. Následně připraví akce armád. Pořadí akcí v rámci kola není striktně dané a může být provedeno v různém pořadí. Když mají všichni hráči připraveny své akce nebo odhlasují vyhodnocení kola, přechází se k fázi vyhodnocení kola.

Jsou provedeny pohyby vojsk po mapě a započaty souboje mezi nimi, případně útoky na města. Jsou vyhodnoceny souboje a odebrány jednotky ze zúčastněných vojsk. Měštům je přičtena produkce, jsou vytvořeny nové jednotky a provedena zadaná vylepšení.

Bonusové akce jsou prováděny na začátku kola, kde hráč může verbovat jednotky, případně přesunout vojska. Verbování v bonusové akci není možné, pokud je dané město pod útokem. Vojska se v rámci pohybu v bonusové akci nemohou přidat k boji, začít bojovat, případně vstoupit na cizí území. Bonusové akce jsou vyhodnoceny před provedením normálních akcí.

1.1.3 Hodnocení hry

Hodnocení hráčů je rozděleno do žebříčků hodnocení měst a vojsk. U měst je primárně hodnocena rozloha ovládaného území a sekundárně úroveň jednotlivých měst. Vojska jsou hodnocena primárně podle počtu vyhraných a prohraných bitev, do hodnocení jsou započítány ztracené a zničené jednotky. Rovnice 1.1 udává výpočet hodnocení hráče v žebříčku a v Tabulka 1.4 jsou definovány použité proměnné a konstanty.

$$h = r + k * \sum_m^M m$$

Rovnice 1.1 Hodnocení města hráče.

Symbol	Popis hodnoty
h	Celkové hodnocení měst jednoho hráče, které určuje jeho v žebříčku měst.
r	Počet polí po vlivem všech měst hodnoceného hráče
M	Množina všech měst vlastněných hodnoceným hráčem.
m	$m \in M$, hodnocení jednoho města.
k	Koeficient pro hodnocení úrovně měst. $\frac{1}{4}$

Tabulka 1.4 Popis symbolů z Rovnice 1.1

Žebříček pro umístění hráčů podle vojenských výsledků oceňuje hráče za vyhrané bitvy, vztah pro výpočet hodnocení hráče je definován v

$$h_a = v - \frac{p}{2} + \frac{1}{10} * \left(k - \frac{d}{2} \right)$$

Rovnice 1.2 Vojenské hodnocení hráče.

Symbol	Popis hodnoty
h_a	Celkové hodnocení vojsk jednoho hráče, které určuje jeho v žebříčku vojsk.
v	Počet vyhraných bitev hráčem a jeho vojsky.
p	Počet prohraných bitev hráčem a jeho vojsky.
m	Celkový počet zničených vojsky hráče.
d	Celkový počet ztracených vojenských jednotek hráče.

Tabulka 1.5 Popis symbolů z Rovnice 1.2

1.1.4 Modelové situace

Modelové situace se zabývají nastíněním aplikace pravidel deskové hry, případně z nich mohou vyplynout nová pravidla. Je rozebrána modelová situace souboje armád a modelová situace obléhání města.

1.1.4.1 Souboj armád

Souboje armád probíhají během několika kol, kdy se v rámci jednoho kola může střetnout pouze omezený počet jednotek. Boje jsou prováděny mezi hráči mezi všemi zúčastněnými hráči. Každý mohl nasadit pouze díl z maxima bojujících jednotek, počet nasazených jednoho hráče je dán celkovým poměrem jednotek v boji mezi hráči. Daný model umožňuje soupeřícím hráčům zásobovat souboje novými jednotkami. U souboje více než dvou hráčů je možné vyjednat mezi hráči koalici a tím podpořit vítězství hráčů v koalici. Každá jednotka má jeden bod života, v závislosti na vylepšení města může mít útok 1 až 3 body. Pro vyhodnocení kola souboje je spočteno kolik každé vojsko udělilo poškození a to je rozděleno rovným dílem mezi ostatní strany. Je spočteno celkové poškození udělené každému vojsku, jsou odečteny jednotky a upraveny hodnoty pro hodnocení hry.

1.1.4.2 Obléhání města

Mějme město s vojenskou posádkou o síle padesát jednotek, dále mějme vojsko A1 o síle 75 jednotek a vojsko A2 o síle 125 jednotek. V první situaci bude útočit vojsko A1 na město, v druhé bude útočit vojsko A2 na město a ve třetí bude útočit na město bez vojenské posádky vojsko A3 o síle 10 jednotek. V situacích nemají vojska ani vojenská posádka žádná vylepšení.

V první situaci má útočící vojsko A1 sílu 75, ale město přidává bránícímu vojsku bonus, který ve výsledku způsobuje, že vojenská posádka je počítána jako vojenská posádka o síle 100 jednotek. Útok bude vzhledem k celkovému počtu nasazených jednotek proveden v rámci jednoho kola a útočnicko vojsko je zničeno a obránci zůstává posádka o síle 12 jednotek.

V druhé situaci má útočící vojsko sílu 125, město obránce má stejně jako v první situaci výslednou sílu 100 jednotek. Do boje je celkem nasazeno 200 jednotek, vyhodnocení probíhá v rámci jednoho kola. V situaci je vojenská posádka města zničena a útočník díky zbylým 25 jednotkám obsazuje město.

Třetí situace popisuje útok na město bez vojenské posádky, zde útočník rovnou obsazuje město. Útok na město formou dlouhodobého obléhání, případně vyhladověním města nebyl možný. Město, které bylo dobito hráčem, přešlo pod správu hráče, v případě dobytí města koalicí vojsk více hráčů záleželo na domluvě hráčů, případně bylo město vypáleno, což znamená, že bylo zničeno.

2 Návrh počítačové hry

Návrh počítačové hry vychází z původní deskové hry, ze které přebírá základní principy, upravuje je, rozšiřuje je, případně přidává nové prvky. Hra je hrána na v jednotlivých kolech, kdy hráči připravují své tahy. Návrh výsledné počítačové hry je rozdělen do fáze základní hry a fáze rozšíření, která jsou detailněji popsána v kapitole 4.

Rozdělení implementace hry do více fází umožňuje postupně hru rozšiřovat a provádět experimenty s hrou samotnou a na základě informací získaných experimenty plánovat další rozvoj návrhu hry.

2.1 Převod hry

Při převodu hry bylo bráno na zřetel, aby byly zachovány principy hry, ale aby hra zároveň byla hratelná na počítači a poskytovala hráčům prostředky pro širší herní možnosti. Kapitola se zabývá vytvořením nástroje pro komunikaci mezi hráči, na jehož základě je vytvořena diplomatická komunikace. V další části je rozebrána role města ve hře a jeho dopad na hru. Poslední část se zabývá rolí vojsk a doplňuje problematiku, jež není popsána v rámci měst.

2.1.1 Mapa hry a základní principy

Mapa hry je tvořena souvislou čtvercovou sítí polí různých typů, mapa nemá explicitně omezené rozměry, ale během hry je nelze měnit. Mapa poskytuje hráči přehled symbolizující herní svět. Vyhodnocení kola je prováděno po určeném časovém intervalu. Jsou zrušeny bonusové akce, hráči o neprovedené akce bez náhrady přicházejí.

Pole jsou ponechána podle původní hry s výjimkou pole Vysoké hory, které je přejmenováno na naleziště kovů. Herní pole nabývají jednoho z následujících typů: louky, lesy, hory, naleziště kovů, voda. Herní suroviny jsou jídlo, dřevo, kovy a platidlo. Platidlo je získáváno za ostatní suroviny v nastavení produkce měst a jsou jím pokryty potřeby vojsk.

2.1.2 Služba zasílání zpráv a diplomacie

Hráči mezi sebou mohou komunikovat pomocí herní pošty. Pošta je používána pro uzavírání spojení a vyhlášení válek zasláním speciální zprávy.

Pro uzavření spojení je nutné, aby spojení potvrdil hráč, kterému je spojení nabídnuto, při potvrzení je upraven diplomatický vztah mezi oběma hráči na spojenecký. Při vyhlášení války nepotvrzuje hráč, kterému je válka vyhlášena změnu stavu, ale může se danou situací pokusit změnit nabídnutím mírové dohody nebo spojení. U hráčů mezi kterými proběhly diplomatické akce, je jako závazná brána poslední diplomatická akce, před vyhodnocením kola. Mírová dohoda upravuje vztahy mezi dvěma hráči na neutrální. Tabulka 2.1 popisuje jednotlivé diplomatické vztahy a diplomatické akce, kterými jsou nastoleny.

Vztah	Diplomatická akce
Spojenecký	Spojenecká dohoda
Neutrální	Mírová dohoda, výchozí stav
Nepřátelský	Vyhlášení války

Tabulka 2.1 Diplomatické vztahy a akce

Ve hře je pracováno pouze s výše uvedenými smlouvami zastoupenými akcemi a jimi vzniklými vztahy, pokud hráči uzavřou jiné dohody, herní systém na ně nebude brát zřetel. Diplomatické vztahy mají dopad na chování vojsk při pohybu a boji. Záznamy z porušení různých dohod hráče, by v rozšíření mohly být zobrazeny ostatním hráčům a ovlivnit jejich ochotu s hráčem jednat. Herní systém jiným způsobem nijak nepenalizuje hráče za porušení smluv.

2.1.3 Města a jejich založení

Hráči zakládají města pomocí vojsk na polích typu louka a les. Samotné město zabírá jedno pole na mapě a jeho osmi okolí musí splňovat podmínky z Tabulka 2.2. Na volných polích pod vlivem města může hráč stavět stavby uvedené v Tabulka 2.3. Stavby uvedené v tabulce jsou do hry zavedeny z důvodu možnosti nepříznivého umístění hráče a přidání možnosti úpravy krajiny pro hráče. Osmi-okolí pole, na němž má být budova vystavěna, musí splňovat následující pravidla:

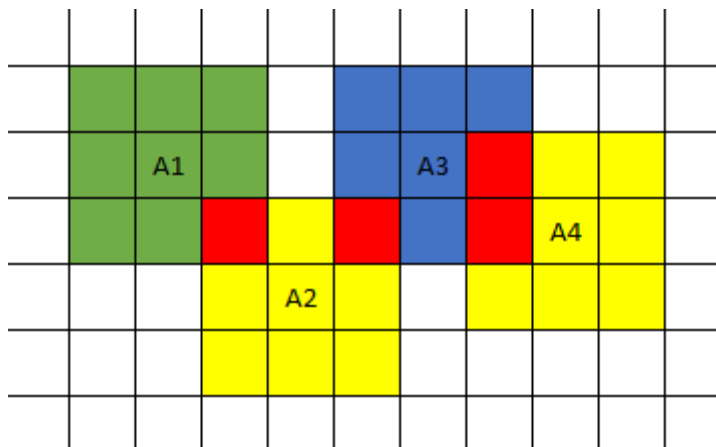
- Nesmí se nacházet v osmi-okolí jiné stavby.
- Pole může spadat do osmi-okolí města.

Do města jsou přidány městské budovy pro řízení přerozdělení přísunu surovin uvedené v Tabulka 2.4, verbování jednotek a samotného zvyšování úrovně města.

#	Podmínka	Pokud není splněna
1	Pole není pod vlivem jiného města.	Akce založení města je zrušena.
2	Na poli se nenachází vojska hráče s neutrální diplomacií	Akce založení města je zrušena.
3	Pole nepatří do osmi-okolí jiného města zakládaného v daném kole.	Akce založení propojených zakládaných měst jsou zrušeny.
4	Na poli se nenachází vojska hráče s nepřátelskou diplomacií	Založení města je odloženo na následující tah. Je řešen souboj mezi vojsky, v případě souboje více hráčů z různých polí nedochází k souboji a akce je zrušena.
5	Pole je typu louka nebo les	Akce založení města je zrušena.

Tabulka 2.2 Podmínky pro založení města

Příklad uplatnění pravidla číslo 3, pokud se překrývají akce založení města, viz Obrázek 2.1. Je dána situace kdy hráč 1 provádí akci založení města A1, hráč 2 provádí akce založení města A2 a A4, hráč 3 provádí akci založení města A3. Daná situace končí zrušením akcí A1, A2, A3 a A4 a oznámením hráčům pomocí herní zprávy od zakládajícího vojska.



Obrázek 2.1 Kolize čtyř akcí založení města.

Stavba	Typ pole	Efekt	Efekt v rozšíření
Farma	lesy, louky	Mění lesy v osmi-okolí na louky	Možnost vylepšení a vyšší zisk jednotek za kolo.
Dřevorubecká osada	lesy, louky	Mění louky v osmi-okolí na lesy	
Důlní osada	naleziště kovů	Umožňuje získávání dodatečných jednotek kovu.	

Tabulka 2.3 Stavby v okolí města

Budova	Účel
Kasárna	Najímání vojenských jednotek
Civilní budovy	Zvyšování úrovně města
Sklad	Řízení přerozdělení produkce, uložení jednotek, převod jednotek na platidlo

Tabulka 2.4 Stavby ve městě

2.1.3.1 Zvyšování úrovně města

Pro zvýšení úrovně města jsou vybrána pole, která nejsou pod vlivem jiného města. Pokud se pole nachází pod vlivem jiného města, zůstává danému městu. V případě, že více měst rozšiřuje svůj vliv na pole ve stejném kole, získává pole město s nejvyšší aktuální úrovní, při shodě je pole přiděleno losem.

Úroveň	Cena jídlo	Cena dřevo	Kola	Nová pole
2	4	4	2	4
3	8	8	4	8
4	16	16	8	16
5	20	20	16	20
6	12	12	8	12
7+	J	D	K	N

Tabulka 2.5 Ceny postupu úrovně

Do úrovně města 6 se nová pole řídí podle znázornění na Obrázek 2.2 a počty nových polí a ceny jsou uvedeny v Tabulka 2.5. Označení L_x odpovídá úrovni $x \in \{1, 2, 3, 4, 5, 6\}$. Při postupu na úroveň 7 a vyšší, jsou nová pole vybrána z polí, která nejsou pod vlivem jakéhokoliv města a jsou

v osmi-okolí polí pod vlivem města postupujícího na novou úroveň. Ceny postupu jsou vypočteny podle Rovnice 2.1. Počet nových polí N pro cenu je vyhodnocen při zadání akce a nemusí odpovídat počtu nově získaných polí v kole, kdy je akce vykonána.

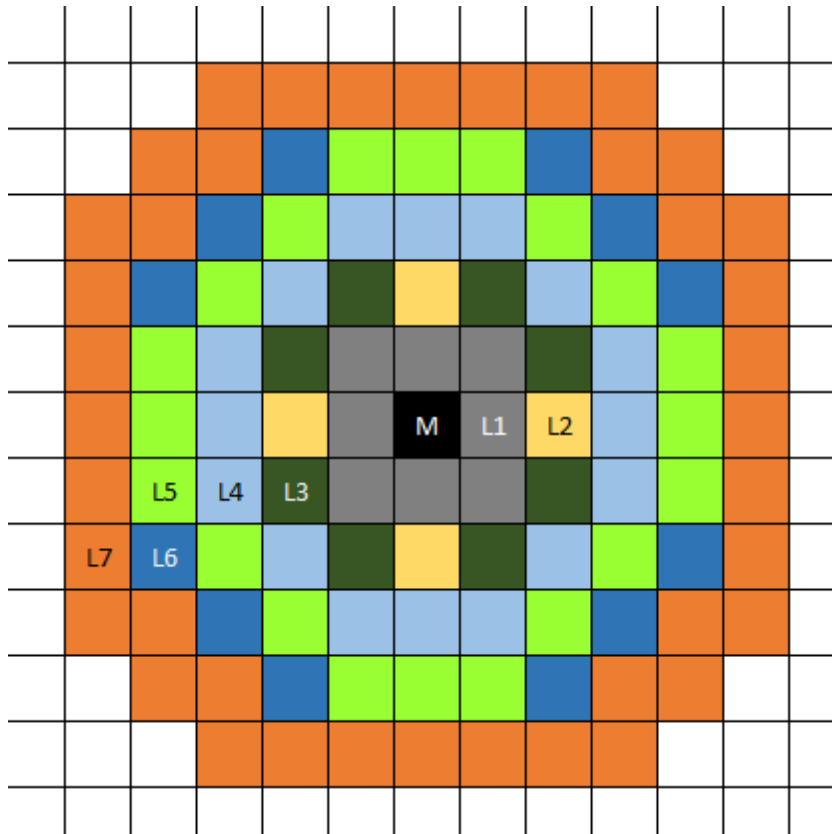
Zaokrouhlování probíhá směrem k nejmenšímu většímu celému číslu při desetinné části čísla větší nebo rovné 0,5, jinak je číslo zaokrouhleno k největšímu menšímu celému.

$$K = \text{Zaokrouhlit}(N * 1,2)$$

$$J = \text{Zaokrouhlit}(N * 1,5)$$

$$D = J$$

Rovnice 2.1 Výpočet cen



Obrázek 2.2 Rozšiřování vlivu města

V každém městě mohou být jeho vlastníkem provedeny akce popsány v Tabulka 2.6, na polích pod vlivem města může jeho vlastník nechat postavit budovu z Tabulka 2.3.

Akce	Popis akce
Verbování vojáků	Ve městě je možné najmout vojenské jednotky, počet verbovatelných jednotek je 10 jednotek za úroveň města, při ceně 2 jednotky jídla a 2 jednotky dřeva za 10 vojenských jednotek.
Nastavení produkce	V městě je možné nastavit, kolik jednotek surovin z produkce bude přeměněno na platidlo a kolik jednotek půjde do skladu.
Vylepšení města	Pokud má město v skladech dostatek surovin na rozšíření, je možné zadat akci k postupu města na následující úroveň. Suroviny jsou odečteny ze skladu.

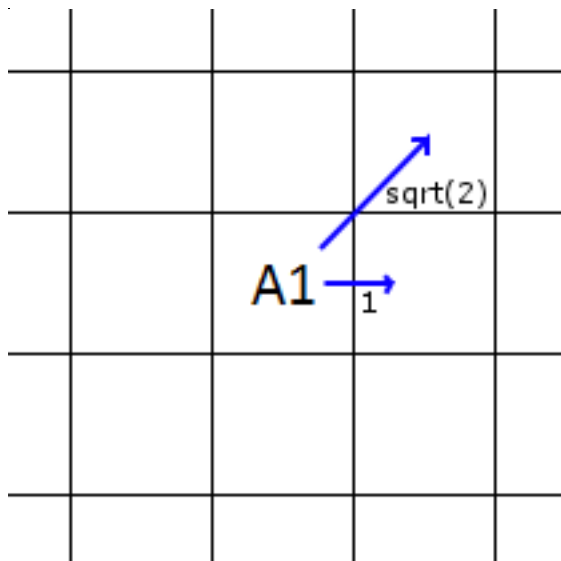
Tabulka 2.6 Akce města

2.1.4 Vojska

Pravidla pro pohyb a vyhodnocení akcí vojsk musí být upravena tak, aby akce mohly být vyhodnocovány automaticky. V základní hře jsou vojska stejně jako v původní hře tvořena pouze jedním typem jednotky, v plánovaném rozšíření herní funkcionality by měly přibýt další dva typy jednotek lukostřelci a kavalérie. Pohyb vojsk po mapě je omezen na vzdálenost tří jednotek, přičemž pohyb v horizontálním nebo vertikálním směru mezi sousedními poli stojí jednu jednotku a diagonální pohyb odmocninu ze dvou, jak je znázorněno v Obrázek 2.3. Vyhodnocení pohybu každého vojska je rozděleno v simulátoru na podle přesunů mezi sousedními poli, které tím vytvářejí simulační čas v rozmezí $t \in \langle 0; 3 \rangle$, aby odpadla nutnost detekce kolízi mezi vojsky při pohybu přímo do cílového pole. Pokud by se vojsko A1 pohybovalo diagonálně, bude při vyhodnocení času $t = 1$ nacházet ve své výchozí pozici v čase $t = \text{sqrt}(2)$ se přesune na novou pozici, pokud to pravidla umožní a je vyhodnocena akce přesunu. Stejným způsobem bude probíhat vyhodnocení pohybu s výjimkou dobývání měst a opevnění pozice vojskem. Cena diagonálního pohybu vojska je získána z délky přepony pravoúhlého trojúhelníku $A = [x, y], B = [x + 1, y], C = [x + 1, y + 1]$ vzniklého pohybem mezi souřadnicemi $[x, y] \rightarrow [x + 1, y] \rightarrow [x + 1, y + 1]$, pohyb vytváří pro hráče prvek, se kterým musí během přípravy strategie počítat. Proveditelné akce vojsky jsou uvedeny v Tabulka 2.7.

Akce	Popis akce
Pohyb vojska	Vojsko se přesunuje mezi poli, pokud se na poli během pohybu setká s nepřátelským vojskem, proběhne mezi nimi boj.
Založení města	Vojsko se během zakládání města v daném kole nemůže pohybovat, aby založilo město, pokud není akce přerušena pravidlem pro zakládání měst, je na konci kola založeno nové město.
Opevnění pozice	Vojsko, které opevňuje pozici, se v daném kole nemůže pohybovat. Opevněná pozice má rozlohu 5x5 polí, vojsko je umístěno v jejím středu. Vojsko zde má specifické chování, při kterém bojuje s postupně s každým nepřátelským vojskem, které se v opevněné oblasti nachází, dokud není zničeno nebo se v oblasti nevyskytují nepřátelská vojska.
Rozpustit vojsko	Vojsko je odebráno ze hry.

Tabulka 2.7 Akce vojsk



Obrázek 2.3 Ilustrace cen pohybu v čtvercové mapě.

Vyhodnocení bitvy vojsk probíhá v čase, kdy se nepřátelská vojska potkají na daném poli. Jsou vybrána první dvě vojska s vzájemně nepřátelskými vztahy, podle nich jsou určeny bojující strany. Primárně je jednou z bojujících stran vojsko již stojící na poli. Ostatní vojska se přidávají podle diplomatických vztahů na jednu či druhou stranu. Pokud má vojsko spojence na obou stranách, boje se neúčastní. Všechny bojové scénáře jsou vyhodnocovány okamžitě s výjimkou útoku na město, který je vyhodnocen na konci kola. Pro samotné určení vítěze a poraženého se v základní implementaci porovnají počty jednotek na obou soupeřících stranách a strana s větším vojskem vítězí. Při útoku na město musí mít má strana obránce při výpočtu koeficient $\frac{3}{2}$, útočníci musí mít přesilu a ještě jim musí zůstat více jak 15% jednotek, aby město dobyli. Ztráty jsou rozděleny mezi vojska útočící strany v poměru počtu nasazených jednotek, pokud jsou ztráty vojska větší nebo rovny 85% nasazených jednotek, vojsko je rozpuštěno. O výsledku bitvy je hráč informován prostřednictvím služby pro zasílání zpráv, kde je uveden výsledek a statistické údaje o zúčastněných stranách, vojscích a počtech jednotek, u kterých jsou uvedeny počty nasazených jednotek a zničených jednotek.

V případě, že je vojsko rozpuštěno z důvodu, že mu zbývá po boji méně jak 15% jednotek, je hráči započítána výhra, ale případná koncová akce založení města nebo obsazení města není provedena.

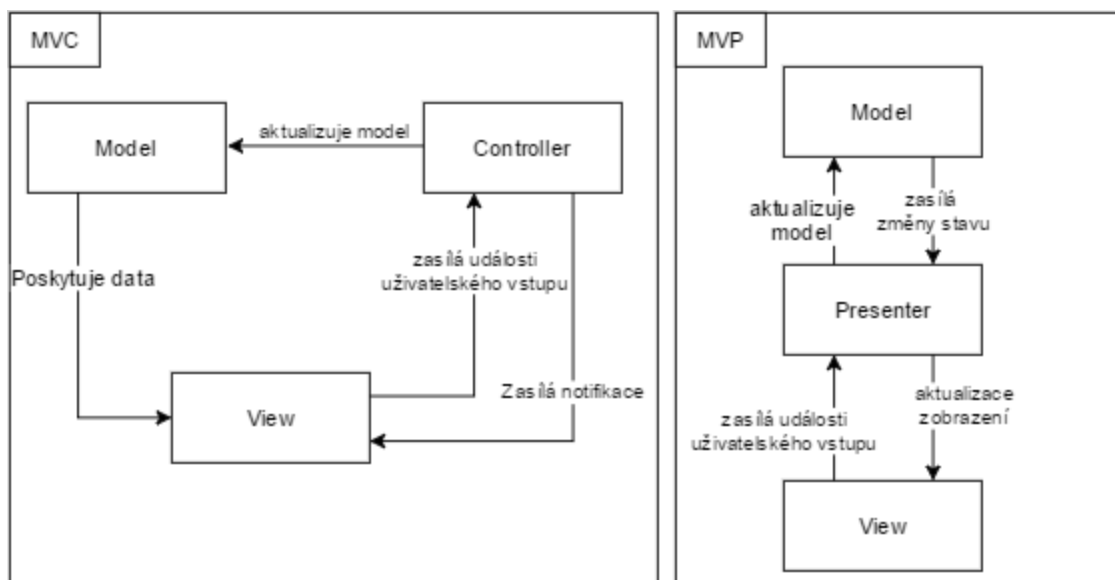
3 Implementace

Kapitola se zabývá implementací hry za použití návrhových vzorů. Pro vytvoření databáze je použito nástroje objektově relačního mapování a její návrh je popsán ER diagramem. V další části je uvedeno rozvrstvení aplikace na základě návrhových vzorů, návrhem struktury databáze. Je představen návrh uživatelského rozhraní. V poslední části se zabývá návrhem simulátoru hry, jež implementuje pravidla a provádí akce při posunech mezi koly.

3.1 Návrhové vzory

Návrhové vzory jsou praxí ověřené a používané obecné případy řešení častých problémů návrhu software. V objektově orientovaném programování se zabývají tvorbou objektů a interakcí mezi nimi. Poskytují zobecněné řešení problému formou šablon, které jsou aplikovatelné na reálné problémy. V knize [2] jsou uvedeny základní návrhové vzory a způsob postupu identifikace správného návrhového vzoru pro daný problém. Návrhové vzory a diagramy popsané UML uvedené v kapitole vychází z uvedené publikace a z internetových stránek [4]. UML je grafický jazyk pro vizualizaci, specifikaci a navrhování software, dále se využívá pro dokumentační účely.

Základní návrhové vzory jsou děleny na vzory pro tvorbu objektů, strukturální vzory a vzory pro chování objektů. Vzhledem k rychlosti vývoje trhu se software řešeními vznikají nové pokročilejší návrhové vzory, případně nové vzory vzniklé úpravou již existujících. Příkladem může být návrhový vzor *model-view-controller (MVC)* a z něj vycházející *model-view-presenter (MVP)*, kde se oba návrhové vzory zabývají oddělením zobrazení (*view*) a business logiky aplikace (*model*). Příklad uvedený v Obrázek 3.1 znázorňuje rozdíl mezi MVC a MVP s pasivním zobrazením, kde MVP odděluje odpovědnosti jednotlivých vrstev užitých pro zobrazení dat.



Obrázek 3.1 Zjednodušené srovnání MVC a MVP

V implementaci práce jsou použity návrhové vzory na platformě .NET Framework, která má komponenty založené na základních, pokročilých a vlastních návrhových vzorech.

3.1.1 MVVM: Model-View-ViewModel

Vývoj aplikací s uživatelským rozhraním, předpokládá zobrazení dat uživateli a změnu dat na základě akcí provedených uživatelem. Návrhový vzor *model-view-viewmodel* [7] odděluje data a logiku aplikace v modelu od jejich zobrazení ve *view* pomocí *viewmodel*, podobně jako návrhový vzor mvp. Viewmodel komunikuje se zobrazením pomocí notifikací, pro které bývá používán návrhový vzor *observer*. Zobrazení zasílá aktualizace dat přes *databinding*, pro uživatelské akce bývá použita implementace návrhového vzoru *command*.

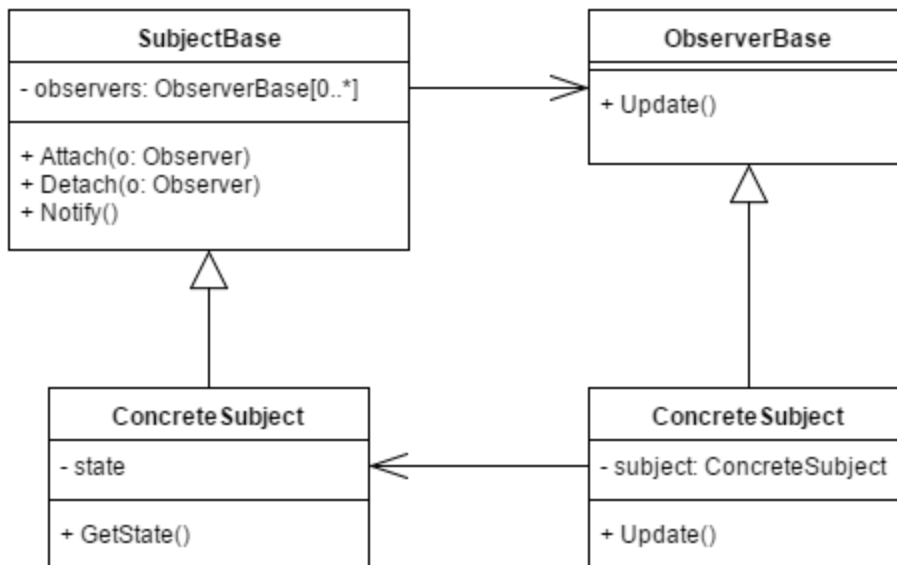
Použití návrhových vzorů v rámci model-view-viewmodel, je uvedeno v závislosti na platformě .NET Framework a implementace hry vychází ze vzorů pro aplikace využívající XAML.



Obrázek 3.2 Zjednodušený příklad MVVM vzoru

3.1.2 Observer

Observer je návrhový vzor pro chování objektů, který definuje způsob komunikace mezi nimi. *Subject* je pozorovaný objekt, u kterého jsou sledovány jeho změny pomocí *observer*, který je pozorovatelem.

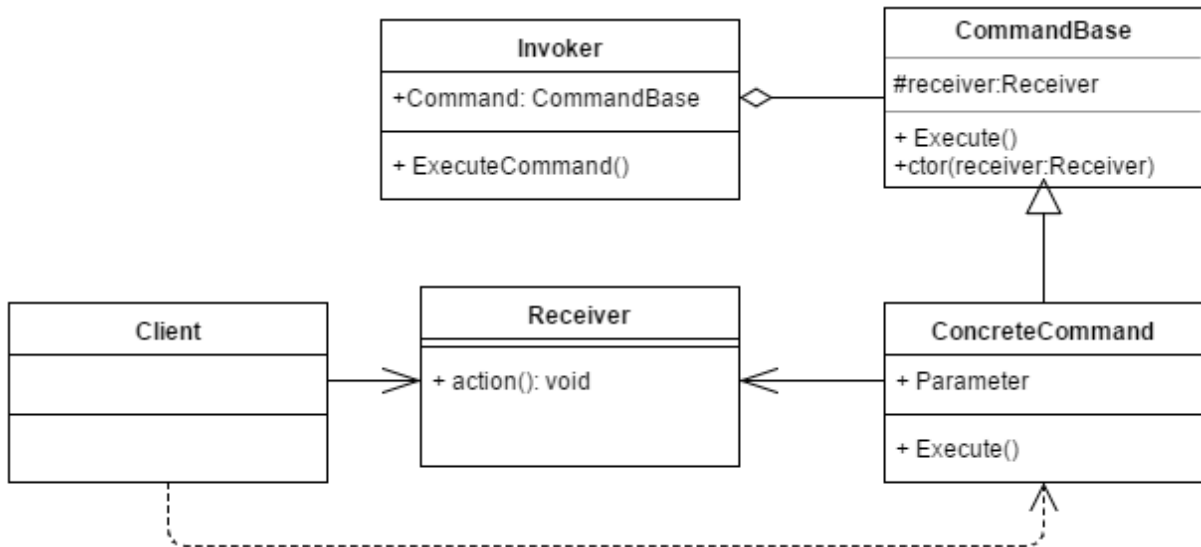


Obrázek 3.3 Návrhový vzor observer

V platformě .NET Framework jsou pro jazyk C# připravena rozhraní `IObservable<T>` a `IObserver<T>` pro implementaci návrhového vzoru. Platforma poskytuje zjednodušené použití návrhového vzoru pomocí `delegate` a `event`. Události jsou používány v pozorovaném objektu, pro registraci delegovaných metod pozorovatelem. Pozorovaný objekt invokes delegované metody v událostech. Delegát uvozený klíčovým slovem `delegate` vytváří předpis pro delegovanou metodu, kde definuje návratový typ a parametry metody. Příloha 1 názorně ukazuje v jazyce C# 6 použití delegace metod a událostí s využitím implementace návrhového vzoru `command` třídou `RelayCommand`.

3.1.3 Command

Návrhový vzor command patří do kategorie behaviorálních vzorů, slouží pro uložení akce, která má být vykonána, ale sám ji neimplementuje. Diagram na Obrázek 3.4 popisuje návrhový vzor.

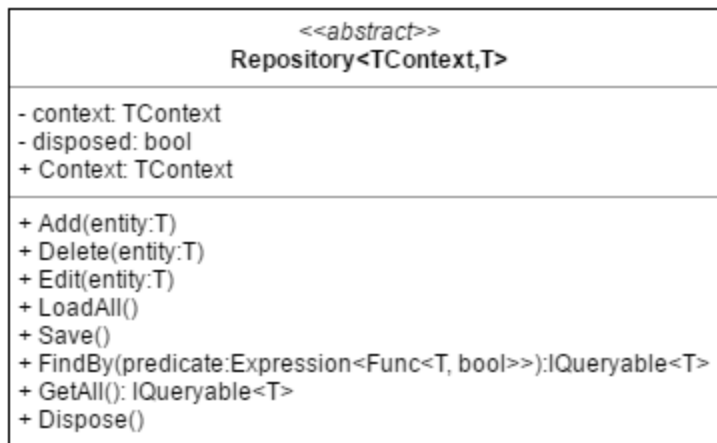


Obrázek 3.4 Návrhový vzor Command

Klient (Client) je třída, která vytváří instance Command a propojuje je s instancemi třídy Receiver, která definuje metodu `action()`, která je zavolána, když je jeden nebo více Command invokován. V .Net Framework je podpora návrhového vzoru zavedena v komponentách uživatelského rozhraní a v kódu pomocí rozhraní `ICommand`. Pro implementaci rozhraní je vhodné použít `RelayCommand`, kde je příkazu v konstuktoru předán delegát na metodu akce, která má být vykonána a predikát pro vyhodnocení spustitelnosti akce.

3.1.4 Repository

Návrhový vzor [8] pro přístup k datům, který má za úkol řešit centralizaci kódu pro přístup k datům a tím odstranit možné duplicity. Přináší podpory silně typovaných objektů. Zlepšuje udržitelnost a čitelnost kódu oddělením business logiky aplikace od logiky služeb pro přístup k datům.

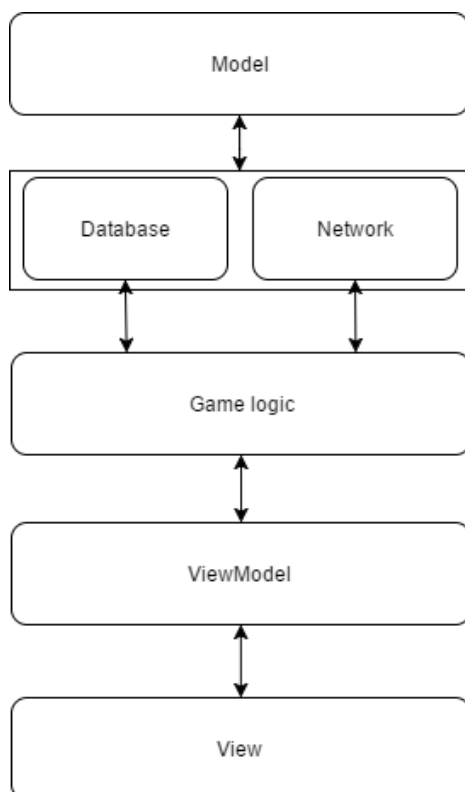


Obrázek 3.5 Použitá implementace návrhového vzoru Repository

Důvod pro použití návrhového vzoru v řešení je odstínění přístupu k perzistentním datům pomocí Entity Framework od vyšších vrstev aplikace. V implementaci je využita modifikace uvedená v diagramu na Obrázek 3.5. Abstraktní generická třída `Repository<TContext, T>`. Třída používá generický datový typ `TContext`, který požaduje konstruktor bez parametrů a je poděděn z základové třídy `DbContext`, která je implementována v Entity Framework a slouží pro připojení k databázi. Generický typ `T` musí splňovat, že je uvozen klíčovým slovem `class`. Třída si při své inicializaci vytvoří kontext pro práci s databází. Metoda `Dispose()` je volána v momentu, kdy instance `Repository` není dále potřeba a má být uvolněna z paměti, volání metody je pro uvolnění povinné a opomenutí může působit problémy se zahlcením paměti a při ukončování běhu aplikace. Jedna konkrétní implementace `repository` umožňuje přístup k jednomu typu objektů daného typu `T`.

3.2 Návrh aplikace

Aplikace je rozdělena do vrstev podle schématu na Obrázek 3.6. Vrstvy reflektují návrhový vzor model-view-viewmodel, síťová komunikace je navržena podle architektury klient server. V modelu je objektová reprezentace relací databáze pro objektově relační mapování, model zároveň definuje základní třídy a rozhraní pro celé řešení. Přístup k databázi a síťové komunikaci zajišťuje vrstva služeb, kde jsou umístěny kontexty pro mapování databáze na objekty a implementace serveru a klienta. Ve vrstvě pro herní logiku je reprezentace mapy, objektů na ní umístěných a simulátor pro vyhodnocení kola. Herní logika je zapouzdřena do vrstvy `ViewModel`, která zprostředkovává výměnu dat mezi zobrazením a vrstvami blíže modelu. Zobrazení prezentuje informace uživateli v lidsky čitelné formě, transformuje informace poskytované vrstvou `ViewModel` do uživatelského rozhraní. Stará se o vykreslení mapy, herních objektů a menu pro potřeby hráče.



Obrázek 3.6 Ilustrace identifikující vrstvy aplikace s využitím návrhového vzoru mvvm

3.2.1 Návrh databáze

Návrh databáze je tvořen pro objektové relační mapování, opisuje objektový model a upravuje relace pro jednoduché mapování na objekty. Návrh databáze je popsán ER digramem na Obrázek 3.9. Entitní množina *Game* tvoří identifikaci pro entity vyskytující se v jedné hře, *Player* identifikuje vlastníka nebo vykonavatele propojených entitních množin ostatních entit ve hře. Entitní množina *City* reprezentuje města, každé město má vlastníka z množiny hráčů a je zařazeno do jedné hry z množiny her. Atributy *X* a *Y* identifikují pozici na mapě, pro identifikaci města hráčem, má jím přiřazené jméno. Entitní množina *InfluencePosition* reprezentuje pole pod vlivem měst. Entitní množina *Structure* reprezentuje stavby, které spadají pod dané město. Atributy *X* a *Y* určují pozici stavby na mapě a *Type* určuje typ budovy ve hře. Entitní množina *Army* reprezentuje vojska, kde je každé vlastněno nějakým hráčem, patří do jedné hry a může podporováno městem. Pro jednodušší identifikaci hráčem jsou vojska pojmenována a podobně jako města mají souřadnice *X* a *Y* pro identifikaci polohy na mapě, počet jednotek v daném vojsku je určen atributem *Units*. Entitní množina *DiplomacyStatus* definuje vztahy mezi hráči v rámci jedné hry.

Entitní množina *GameAction* reprezentuje akce prováděné hráči v rámci hry. Atribut *Discriminator* určuje koncový objektový typ pro mapování. Mapovací nástroj určuje, které vlastnosti entity budou použity pro vytvoření mapovaného objektu. Entitní množina *Messages* je navržena podobným způsobem pro různé typy zpráv zasílaných mezi hráči během jedné hry.

V diagramu na Obrázek 3.9 jsou uvedeny pouze nejdůležitější entitní množiny a vazby z důvodu využití generování databáze z kódu aplikace pomocí Entity Framework[3]. Většinu vazebních tabulek a cizích klíčů je Entity Framework schopen určit podle kódu, jinak je nutné explicitně definovat, jakým způsobem mají být objekty mapovány, příkladem je vazba *m* ku *n*.

3.2.2 Návrh uživatelského rozhraní

Uživatelské rozhraní je vytvořeno na podsystému .NET Framework *Windows Presentation Foundation*, zkráceně WPF, které používá značkovací jazyk XAML pro popis elementů zobrazovaných elementů a je rozděleno na obrazovku pro navigaci v hlavním menu a obrazovku pro zobrazení hry. Obrazovka pro hlavní menu je tvořena pohledem pro zobrazení samotného menu a pohledy pro jednotlivé položky menu. Obrazovka pro vykreslení hry je rozdělena do tří částí. Oblast pro vykreslení zmenšené mapy, oblasti pro zobrazení aktuálního herního menu a oblasti pro vykreslení herního obsahu. Rozložení oblastí je ukázáno v **Chyba! Nenalezen zdroj odkazů.**

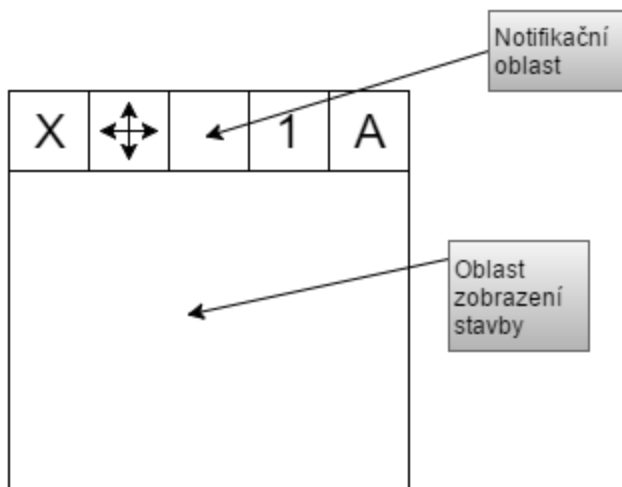
Mapa herního plánu je vyobrazena v oblasti pro herní obsah, tvoří ji síť čtvercových dlaždic, které reprezentují jednotlivá pole na mapě. Notifikační oblast je dokována na vrchní část dlaždice s výškou jedné pětiny délky strany dlaždice, je rozdělena na pět částí, směrem zleva je první oznámení pro hráče, zda jeho vojsko chrání pole. Druhé pole notifikační oblasti oznamuje hráči, zda se přes pole pohybuje alespoň jedno jeho vojsko, třetí pole je ponecháno volné. Čtvrté a páté pole oznamují počet jednotlivých vojsk na jednom poli, kde na je čtvrtém poli počet vojsk a na pátém ikona symbolizující vojsko. Oblast pro zobrazení stavby zabírá zbytek pole a v případě, že se na poli nachází stavba tak ji zobrazí, jinak zůstává prázdné. Struktura dlaždice je rozkreslena na **Chyba! Nenalezen zdroj odkazů.** celý návrh je ukázán na snímku obrazovky z běhu aplikace na Obrázek 3.11.

První prototyp hry byl vytvořen na platformě XNA Framework[4], které měla poskytovat jednotné prostředí pro implementaci her na Xbox, Windows Phone a desktopové operační systémy. Grafický návrh prototypu je ukázán na snímku obrazovky v Obrázek 3.10, od vývoje na XNA Framework bylo upuštěno z důvodu zrušení vývoje na novějších operačních systémech společnosti Microsoft. V prototypu bylo pracováno s nástroji pro platformu, tudíž většina kódu je nepoužitelná,

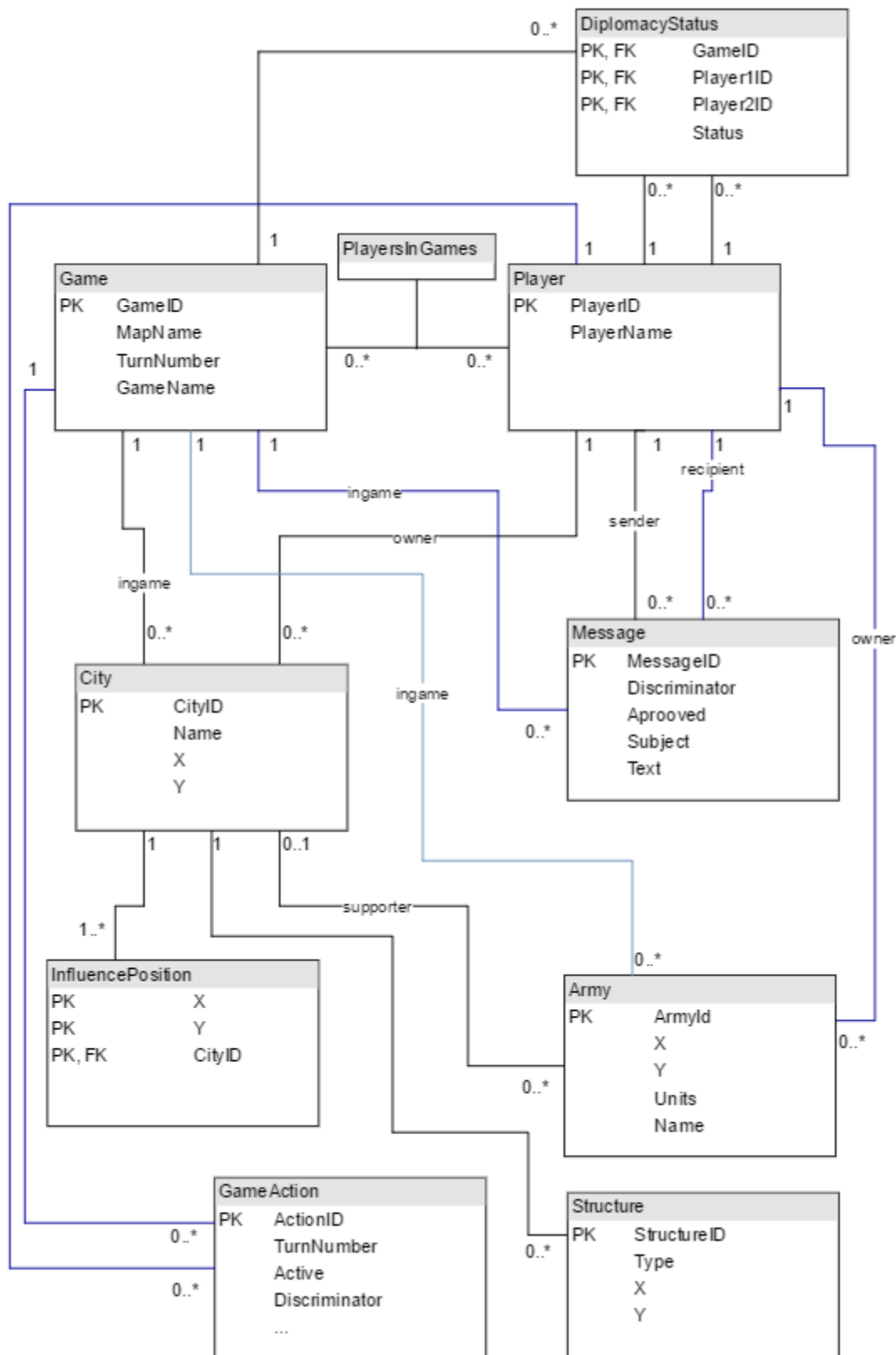
ovšem v aktuální implementaci jsou použity principy z reprezentace mapy a vizualizace herních objektů. U mapy je použito posouvání polí mapy pod vizualizačními dlaždicemi.



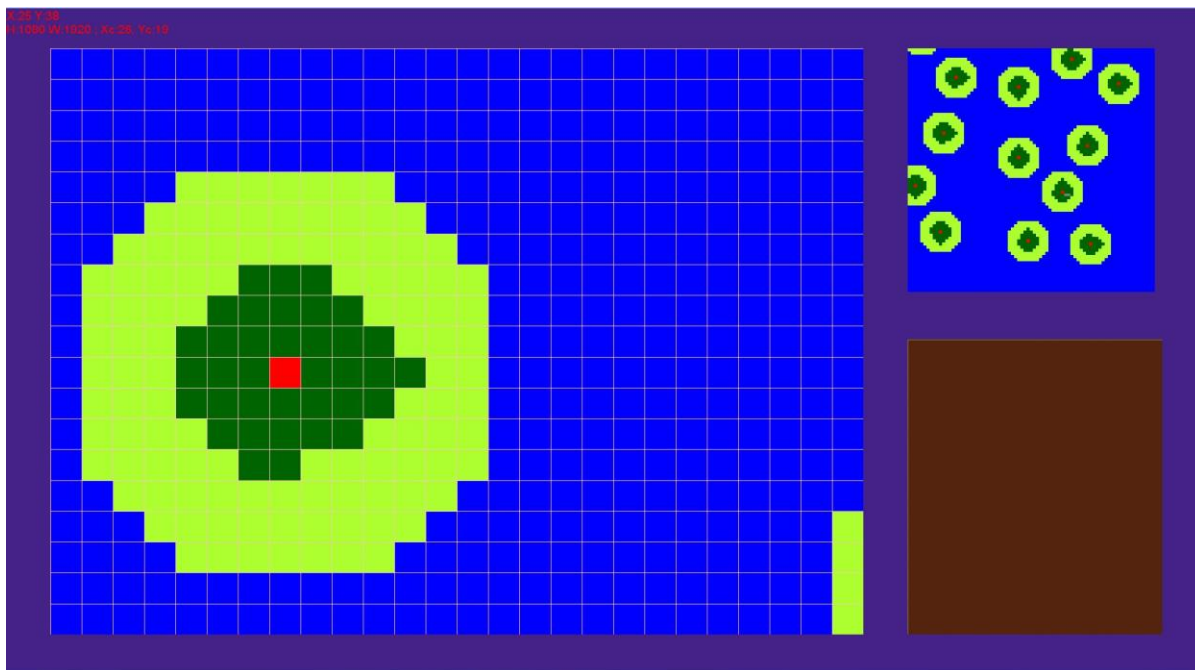
Obrázek 3.7 Návrh rozložení kontejnerů v uživatelském rozhraní



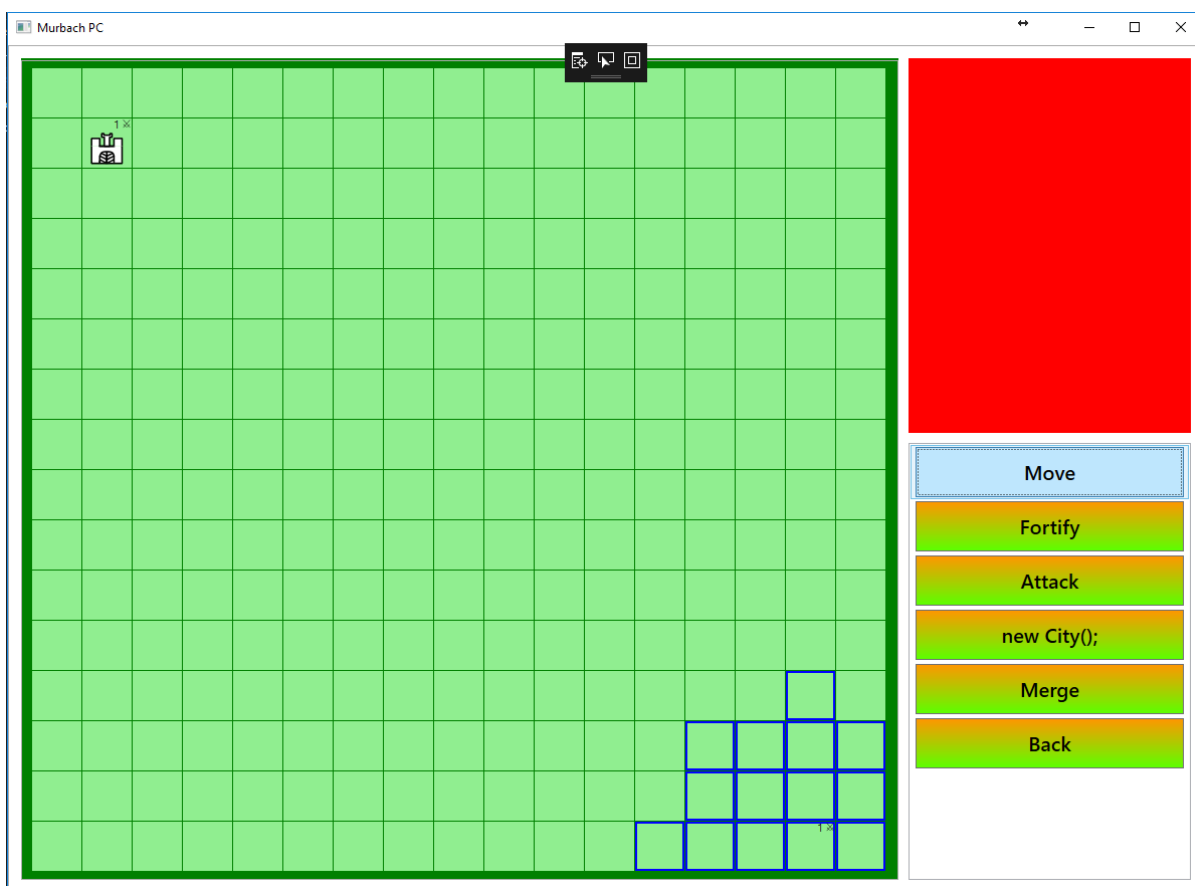
Obrázek 3.8 Zevrubný návrh dlaždice



Obrázek 3.9 Zjednodušený ER diagram hlavních entit



Obrázek 3.10 Prototyp návrhu uživatelského na XNA Framework



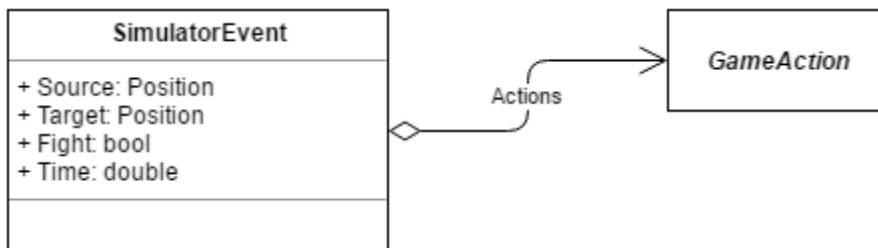
Obrázek 3.11 Grafický návrh herního přehledu

3.2.3 Návrh simulátoru

Simulátor slouží pro řešení vyhodnocení akcí zadaných hráči, je inspirován principy pro diskretní simulaci. Pro každé kolo je inicializována nová simulace, jejíž výsledky mění aktuální stav modelu. Časová osa simulačního modelu je vytvořena převedením maximální vzdálenosti, kterou může za kolo urazit vojsko, v případě základní implementace je maximální čas $max_t = 3$. Každá akce má určeno pořadové číslo kola, ve kterém má být vykonána.

Ve vyhodnocovaném kole jsou použity akce s číslem daného kola. Během inicializace simulace jsou vyhodnoceny diplomatické akce a jsou upraveny diplomatické vztahy hráčů. Pro automatickou detekci kolizí akcí je použita datová třída `SimulatorEvent` popsaná na Obrázek 3.12. Vlastnosti `Source` a `Target` identifikují pole, na kterých jsou akce provedeny, `Fight` označuje, zda se během vykonání některé z akcí má odehrát bitva mezi vojsky, `Time` reprezentuje čas, kdy mají být akce provedeny. Akce, které jsou na mapě statické, jsou umístěny do kolekce pro statické akce, během inicializace simulace. Kolize statických akcí jsou vyhodnoceny během inicializace simulace, pokud akce nemohou být nadále provedeny, jsou zrušeny. Akce, u kterých dochází k pohybu vojenských jednotek na mapě, jsou seskupeny do vlastní kolekce, ve které je chování jejich prvků závislé na simulačním čase.

Při detekci kolize jsou identifikována všechna zúčastněná vojska. Poté jsou určeny soupeřící strany a proveden boj, případně je boj zrušen a hráči jsou informováni o výsledku. Pokud je kolizí útok na město, je boj naplánován na čas konce simulace. Po ukončení simulace jsou městům připsány získané suroviny, je rozšířen jejich vliv, pokud k tomu v kole existovala příslušná akce a nebyla zrušena. Pro práci s kolekcemi je využita knihovna `LINQ`[11], která usnadňuje práci s kolekcemi a zpřehledňuje kód. `LINQ` umožňuje psát přímo dotazy podobné integrovanému `sql` přímo do kódu nebo využívat extension metod nad rozhraním `IEnumerable<T>`, kterým je předáván predikát pro jejich vyhodnocení případně vykonaná akce nebo funkce.



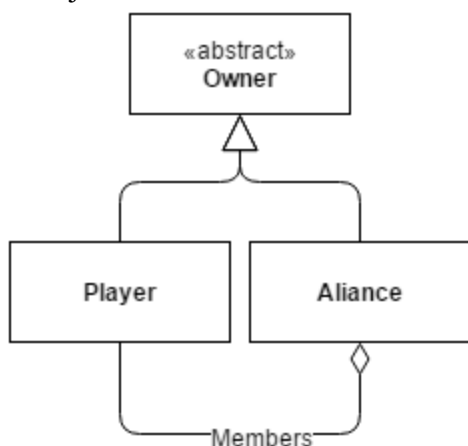
Obrázek 3.12 Digram třídy `SimulatorEvent`

4 Zpětná vazba a plán vývoje hry

Zpětná vazba byla získávána během vývoje a implementace hry, v omezeném kruhu lidí. Některé poznatky jsou přímo zařazeny do návrhu hry, jiné jsou plánovány v rámci dalšího vývoje hry. Součástí dalšího vývoje hry je implementace fáze rozšíření uvedených v jejím návrhu.

4.1 Plánovaná rozšíření

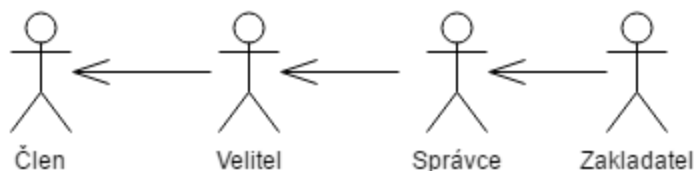
Plánovaným rozšířením je možnost hráčů se sdružovat do aliancí, které budou vystupovat jako další poddědný typ vlastníka (Owner) znázorněného na Obrázek 4.1. členové aliance budou dostávat pod správu alianční města a související vojska. Aliance bude rozlišovat několik druhů rolí členů uvedených v Tabulka 4.1, dědičnost rolí je na Obrázek 4.2 Dědičnost rolí. Členové budou moci alianci věnovat svá města, za podmínky, že jim zbyde minimálně jedno město a město věnované alianci může být vráceno zpět pouze hráči, který jej věnoval. Aliance může věnovat hráči alianční město, které bylo založeno vojskem aliance.



Obrázek 4.1 Digram pro model aliance.

Druh členství/role	Popis
Zakladatel	Zakladatelem je jeden hráč v rámci jedné aliance, dokud je mu členství přiřazeno nemůže opustit alianci. Pokud je posledním členem aliance může alianci rozpustit, jinak může roli předat jinému členovi. Přiřazuje a odebrá roli správce.
Správce	Hráč může přijímat a vylučovat členy do aliance, udělovat a odebrat roli velitele, může spravovat všechna města a vojska. Pokud je v alianci víc jak jeden správce, mohou správci hlasovat o změně zakladatele, přidělení nebo odebrání role správce jinému hráči.
Velitel	Hráči mohou být svěřena vojska a města, se kterými bude provádět akce. Jedno město nebo vojsko může spravovat více hráčů.
Člen	Členství v alianci zajišťuje mezi hráči spojenecké diplomatické vztahy, hráč může být členem pouze jedné aliance v rámci jedné hry.

Tabulka 4.1 Role v alianci



Obrázek 4.2 Dědičnost rolí

Dalším plánovaným rozšířením je přidání jednotek dalších dvou typů vojenských jednotek a rozdělení na pěchotu, střelce a kavalérii, které jsou popsány Tabulka 4.2. Rozšíření ekonomické stránky hry spočívá ve vylepšení staveb v okolí města a přidání přístavu a lodí. Přidání lodí je rozděleno na fázi přidání transportních lodí a následně naplánování fáze přidání válečných námořních plavidel. Všechna plavidla předpokládají, obsazení vojenskou posádkou nebo její přepravu. Přidání dalších typů vojenských jednotek vyžaduje úpravu systému vyhodnocení bitev.

Typ jednotky	Popis jednotky
Pěchota	Základní vojenská jednotka
Střelci	Vojenská jednotka s bonusem proti pěchotě, spočívajícím v pobití části pěchoty před bojem na zteč.
Kavalerie	Jednotka má bonus proti střelcům, v redukci poškození před ztečí.

Tabulka 4.2 Rozdělení typů vojenských jednotek

4.2 Rozšíření na základě zpětné vazby

Během konzultací návrhu hry a tvorby prototypu, byly zaznamenány požadavky na nástroj pro tvorbu mapy, přidání počítačem řízených hráčů, zapracování hexagonální mapy. Bylo zmíněno i uživatelské rozhraní aplikace a možnost uživatelského nastavení barev a textur.

4.2.1 Tvorba mapy

Jedním z hlavních požadavků je vytvoření zabudovaného editoru map. Funkce editoru jsou popsány v Tabulka 4.3.

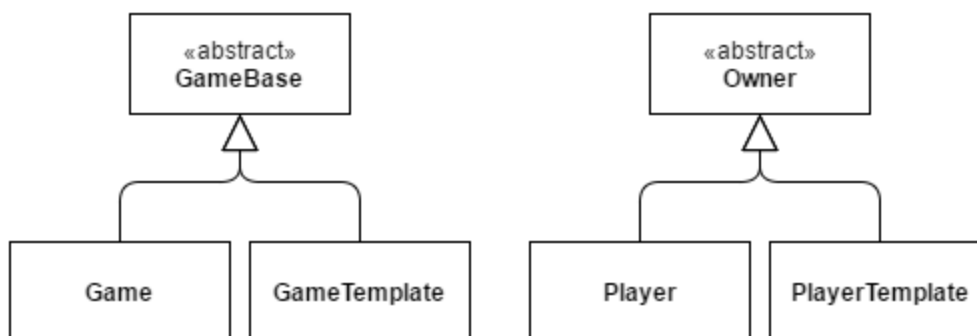
Funkce	Popis
Tvorba nových map	Vytvoření nové mapy o daných rozměrech s jedním typem herních polí, pro vytvoření mapy uživatelem.
Generování terénu	Vytvoření nové mapy nebo segmentu v existující mapě na základě uživatelem zadaných parametrů. Funkčnost potřebuje algoritmus pro generování terénu.
Úprava existujících map	Poskytnout uživateli možnost úpravy existující mapy, zvětšení a zmenšení velikosti mapy.
Vytvoření herní scénáře	Uživatel má možnost na existující mapě vytvořit města, vojska a další herní objekty

Tabulka 4.3 Výpis jednotlivých funkcí pro tvorbu mapy

Funkce tvorbu a úpravu map by mohla využívat upravené rozhraní pro vyobrazení herní mapy s novými kontextovými menu. Pro generování terénu by mohlo být vhodné použít již existující volně dostupné řešení nebo použít existující algoritmy a zakomponovat je do řešení.

Vytvoření herního scénáře by zasáhlo do modelu aplikace. V modelu by se přidaly šablony pro hru a hráče podle diagramu v Obrázek 4.3, ze kterých by se následně generovala hra. Při využití

objektově relačního mapování z návrhu aplikace, budou změny zakomponovány do databáze na základě modelu a související entitní množiny budou upraveny diskriminátorem.



Obrázek 4.3 Návrh na úpravu modelu

4.2.2 Vylepšení uživatelského rozhraní

Identifikaci objektů na mapě lze vylepšit jasným odlišením vlastních a cizích herních objektů, při plánování pohybu vojska barevně oddělit zvýraznění polí vůči kolům. Vytvoření grafického uživatelského rozhraní odpovídajícího dnešním hrám. Možnost vytvoření vlastních textur pro hru uživatelem.

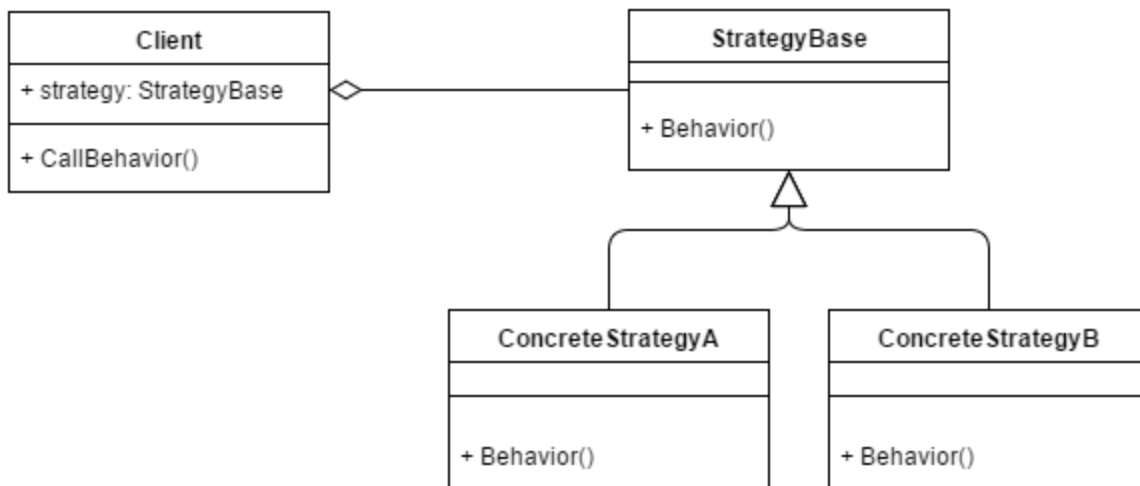
Lepší identifikaci objektů na mapě lze provést použitím více druhů dlaždic a umístěním notifikačních ikon do rohů dlaždice a zvětšit je. Pokud se na poli mapy nenachází budova, je možné umístit na dlaždici místo budovy grafickou reprezentaci symbolizující přítomnost vojska. Odlišení vlastních a cizích herních objektů, lze vyřešit barevným rozlišením ikon na mapě.

Estetické preference uživatelů se většinou liší, z toho důvodů je v rámci vylepšení vzhledu aplikace přínosné poskytnout prostor pro vytvoření vlastního vzhledu, formou změn barev aplikace a vytvořením vlastních textur. Osobní nastavení hry by mohlo být uloženo, buď přímo u klienta, nebo také centralizovaně na serveru s čímž souvisí povedení analýzy udržitelnosti.

4.2.3 Přidání hexagonální mapy

S hexagonální mapou bylo experimentováno v deskové hře, ale z důvodu změn v pravidlech, tvorby a pochopení mapy hráči byla zavržena. V počítačové hře přináší zjednodušení pohybu, nemusí se řešit pohyb rovně nebo diagonálně, cena pohybu může počítána pouze mezi sousedními poli, které mají společnou stranu. Dané rozšíření vyžaduje vytvoření nových pravidel hry na základě existujících pravidel. Nevýznamnější změnou je vytvoření nových pravidel pro rozšiřování vlivu města, úpravy v navigaci při tvorbě cesty vojska a vyhodnocování akcí vojska. Z implementačního hlediska by bylo nutné přetvořit návrh aplikace a zavést do něj návrhový vzor *abstract factory* [9][10][2] (abstraktní továrna), případně *strategy*, pro výběr komponent a vyhodnocovacích algoritmů podle typu mapy.

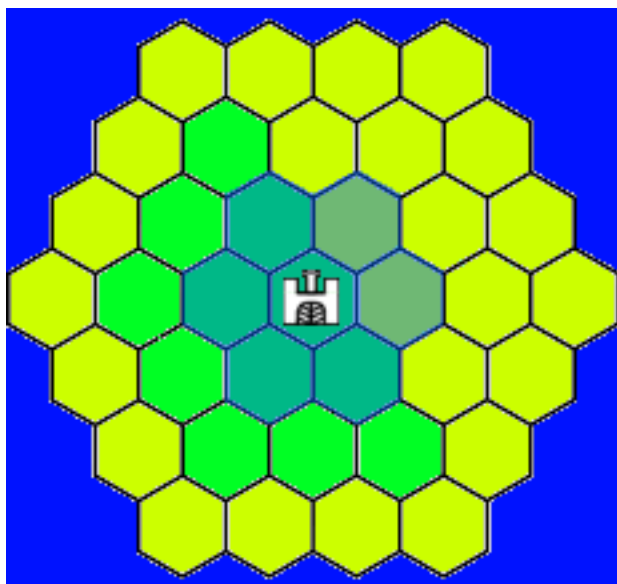
Strategy je návrhový vzor pro chování, kde je definováno rozhraní (*interface*) volání společné pro všechny konkrétní algoritmy, přínosem je možnost zaměnitelnosti jednotlivých algoritmů za běhu aplikace. Pro jeho použití bude nutné provést analýzu, zda je opravdu potřebný.



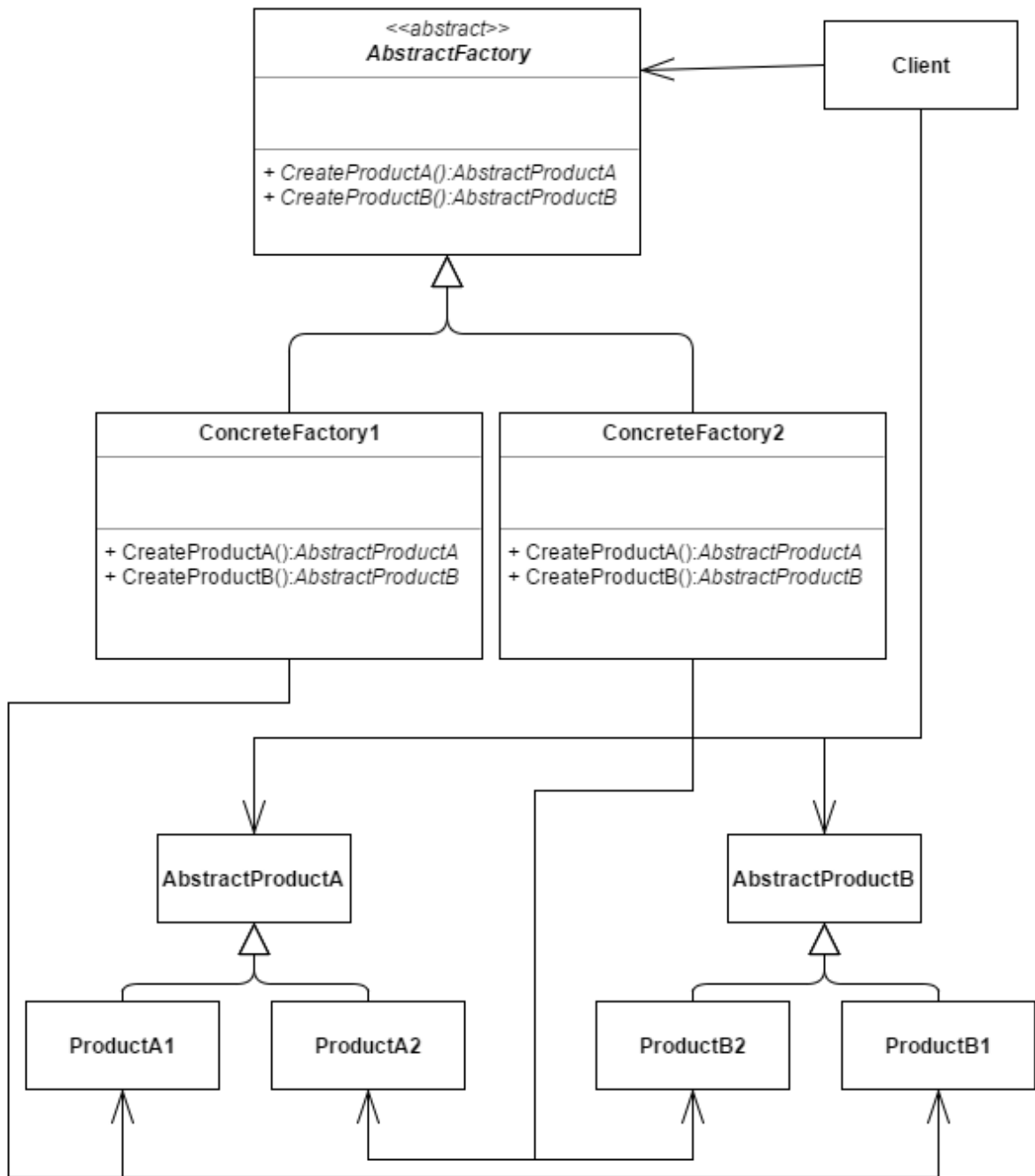
Obrázek 4.4 Návrhový vzor strategy

Abstraktní továrna je tvořivý návrhový vzor, který vytváří rozhraní třídu továrny, jejíž konkrétní implementace tvoří konkrétní produkty (`ProductA1`, `ProductB1` ...), popisuje rozhraní abstraktních produktů (`AbstractProduct`). Návrhový vzor je popsán diagramem na Obrázek 4.6. Implementace založená na abstraktní továrně, není závislá na konkrétní implementaci jednotlivých produktů a poskytuje prostor pro jednodušší záměnu použitých cílových komponent.

U hexagonální mapy by abstraktní továrna implementovala konkrétní továrny pro čtvercovou mapu a pro hexagonální. Obrázek 4.5 nastiňuje možný grafický návrh zobrazení hexagonální mapy, modrá pole znázorňují hranici mapy, žlutá pole louky a zelená lesy. Ikona hradu symbolizuje město a namodralé překrytí polí vymezuje vliv daného města. Přínosným rozšířením je možnost obsazování jednotlivých polí a přidání obchodu se sférou vlivu. Hráči by tak mohli vyměňovat pole mezi městy, jejichž hranice jsou společné, přičemž by sféra vlivu musela zůstat souvislá. Byla by přidána diplomatická akce obchodu s územím, kde by hráči dohodli cenu v podobě vyměněných polí a surovin.



Obrázek 4.5 Návrh zobrazení hexagonální mapy



Obrázek 4.6 Abstract factory

4.2.4 Přidání počítačem řízených hráčů

Počítačem řízení hráči by mohli mít přínos v podobě hry jednoho hráče proti nim, případně dalšího hráčsky zajímavého prvku v normální hře mezi hráči. Přínosným experimentem s umělou inteligencí může být vytvoření rozhraní pro uživatelskou implementaci v interpretovaném jazyce.

5 Závěr

Práce se zabývala deskovou hrou Murbach. Proces získávání původních pravidel hry byl problematický z důvodu časového odstupu mezi hraním deskové hry a tvorbou této práce, rovněž vzhledem k faktu, že pravidla hry nebyla nikdy formálně popsána, ani dochována v písemné podobě. Pravidla deskové hry byla rekonstruována za pomoci konzultací s některými členy oddílu Dvojkaři, kteří se této hry účastnili. Neúplnost některých pravidel je dotvořena v návrhu počítačové hry, která se hlavně zabývá pravidly a principy správy měst a vojsk.

V implementaci jsou využity používány metody softwarového inženýrství. Pro návrh implementace jsou využity diagramy, které jsou vyobrazeny pomocí jazyka UML. Jsou rozebrány stěžejní návrhové vzory pro implementaci, která vychází ze vzoru model-view-viewmodel. Databáze pro uložení stavu hry byla vytvořena pomocí nástroje Entity Framework, vygenerováním objektově relačního mapování ze zdrojového kódu v jazyce C#. Je rozebrán návrh uživatelského rozhraní.

Byla navržena plánovaná rozšíření pro další rozvoj hry. Každé rozšíření je plánováno jako samostatná fáze. Bylo řešeno přidání dvou typů vojenských jednotek a následné rozdělení vojenských jednotek na pěchotu, střelce a kavalerii. Dalším herním rozšířením bylo přidání přístavů a zavedení transportních lodí, na základě tohoto kroku byl nastíněn plán pro přidání námořních bitev a válečného námořnictva. Plánovaným rozšířením bylo také přidání aliancí do hry jako dalšího přínosného prvku.

Na základě informací získaných ze zpětné vazby byla navržena další rozšíření hry. Byl rozebrán nástroj pro tvorbu a úpravy uživatelských map, který by mohl být rozšířen o tvorbu herních scénářů. Byly navrženy úpravy uživatelského rozhraní a rozpracována možnost umožnění uživatelům upravit vzhled aplikace. Byl identifikován možný dopad rozšíření v podobě zavedení hexagonální mapy do hry a provedena zevrubná analýza použitelných implementačních postupů. Byla také nastíněna možnost přidání počítačem řízených hráčů a možnost vytvoření rozhraní pro uživatelskou implementaci.

Literatura

- [1] Junák – Český skaut středisko Dvojka Brno, z.s.
URL: <http://www.dvojkabrno.cz>
- [2] GAMMA, Erich. Design patterns: elements of reusable object-oriented software. Reading, Mass.: Addison-Wesley, c1995. ISBN 0201633612.
- [3] Entity Framework
URL: [https://msdn.microsoft.com/en-us/library/gg696172\(v=vs.103\).aspx](https://msdn.microsoft.com/en-us/library/gg696172(v=vs.103).aspx)
- [4] XNA Framework
URL: <https://msdn.microsoft.com/en-us/library/bb203940.aspx>
- [5] BlackWasp [online]. [cit. 2016-05-16]. Dostupné z: <http://www.blackwasp.co.uk/>
- [6] Murbach Abbey. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-05-16]. Dostupné z: https://en.wikipedia.org/wiki/Murbach_Abbey
- [7] The Repository Pattern [online]. [cit. 2016-05-16]. Dostupné z: <https://msdn.microsoft.com/en-us/library/ff649690.aspx>
- [8] The MVVM Pattern [online]. [cit. 2016-05-16]. Dostupné z: <https://msdn.microsoft.com/en-us/library/hh848246.aspx>
- [9] Abstract Factory [online]. [cit. 2016-05-16]. Dostupné z: <http://www.oodeesign.com/abstract-factory-pattern.html>
- [10] Abstract Factory [online]. [cit. 2016-05-16]. Dostupné z: <http://www.blackwasp.co.uk/AbstractFactory.aspx>
- [11] LINQ (Language-Integrated Query) [online]. [cit. 2016-05-16]. Dostupné z: <https://msdn.microsoft.com/en-us/library/bb397926.aspx>

Seznam příloh

Příloha 1. Použití event, delegate a command

Příloha 2. DVD

Příloha 1. Použití event, delegate a command

```
public class MainMenu : IDisposable
{
    public delegate void EntryMenuDelegate();

    public event EntryMenuDelegate CreateGameEvent;
    public event EntryMenuDelegate JoinGameEvent;
    public event EntryMenuDelegate ExitGameEvent;

    public ObservableCollection<MenuItem> EntryMenu { get; }
        = new ObservableCollection<MenuItem>();

    private void CreateMenu()
    {
        EntryMenu.Add(new MenuItem()
        {
            Text = "Create Game",
            ActionCommand = new RelayCommand(() => CreateGameEvent?.Invoke())
        });
        EntryMenu.Add(new MenuItem()
        {
            Text = "Join Game",
            ActionCommand = new RelayCommand(() => JoinGameEvent?.Invoke())
        });
        EntryMenu.Add(new MenuItem()
        {
            Text = "Exit Game",
            ActionCommand = new RelayCommand(() => ExitGameEvent?.Invoke())
        });
    }

    public void Dispose()
    {
        CreateGameEvent = null;
        JoinGameEvent = null;
        ExitGameEvent = null;
    }
}
```

```

public class GameViewModel : NotifyObject
{
    private int selectedViewPage;

    public MainMenu Menu { get; } = new MainMenu();

    public int SelectedViewPage
    {
        get { return selectedViewPage; }
        set { Set(ref selectedViewPage, value); }
    }

    public GameViewModel()
    {
        Menu.CreateGameEvent += Menu_CreateGameEvent;
        Menu.JoinGameEvent += Menu_JoinGameEvent;
        Menu.ExitGameEvent += Menu_ExitGameEvent;
    }

    private void Menu_CreateGameEvent()
    {
        SelectedViewPage = 1;
    }

    private void Menu_JoinGameEvent()
    {
        SelectedViewPage = 2;
    }

    private void Menu_ExitGameEvent()
    {
        Menu.Dispose();
        Application.Exit();
    }
}

```