



**MATEMATICKO-FYZIKÁLNÍ  
FAKULTA**  
Univerzita Karlova

## **BAKALÁŘSKÁ PRÁCE**

Jméno Příjmení

**Název práce**

Název katedry nebo ústavu

Vedoucí bakalářské práce: Vedoucí práce

Studijní program: studijní program

Studijní obor: studijní obor

Praha ROK

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne .....

Podpis autora

Poděkování.

Název práce: Název práce

Autor: Jméno Příjmení

Katedra: Název katedry nebo ústavu

Vedoucí bakalářské práce: Vedoucí práce, katedra

Abstrakt: Abstrakt.

Klíčová slova: klíčová slova

Title: Name of thesis

Author: Jméno Příjmení

Department: Name of the department

Supervisor: Vedoucí práce, department

Abstract: Abstract.

Keywords: key words

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
1.1	RTS hry blíže . . . . .	3
1.1.1	Mapy . . . . .	5
1.1.2	Jednotky . . . . .	6
1.1.3	Budovy . . . . .	9
1.1.4	Suroviny . . . . .	11
1.1.5	Vývoj technologií . . . . .	12
1.2	Uživatelé . . . . .	13
1.2.1	Tvůrci her . . . . .	14
1.2.2	Hráči her . . . . .	15
1.3	Ukázková hra . . . . .	15
1.4	Cíle práce . . . . .	16
<b>2</b>	<b>Analýza</b>	<b>18</b>
2.1	Herní engine . . . . .	18
2.2	Podporované platformy . . . . .	18
2.2.1	Mobilní platformy . . . . .	18
2.3	Formát balíčků a jejich načítání . . . . .	19
2.3.1	Základní struktura . . . . .	19
<b>3</b>	<b>Ukázková hra</b>	<b>20</b>
<b>4</b>	<b>Programátorská dokumentace</b>	<b>21</b>
<b>5</b>	<b>Uživatelská dokumentace</b>	<b>22</b>
	<b>Závěr</b>	<b>23</b>
	<b>Seznam použité literatury</b>	<b>24</b>
<b>A</b>	<b>Přílohy</b>	<b>25</b>
A.1	První příloha . . . . .	25

# 1. Úvod

Strategické hry jsou žánrem, ve kterém hráči využívají svých mentálních schopností, především taktického a strategického myšlení, pro porážku jednoho či více nepřátel. Ve většině případů se strategické hry zabývají tématem války. [1]

Žánr strategických her obsahuje mnoho poddruhů s velice rozdílnými nároky jak na hráče, tak na vývojové prostředí a na vývojáře samotného. Prvním kritériem pro rozdělení strategických her je, zda se hra odehrává jako posloupnost diskretních tahů, nebo zda se hra odehrává v přímé závislosti na ubíhajícím reálném čase. Druhým kritériem je relativní četnost a důležitost strategických rozhodnutí vůči taktickým rozhodnutím. Podle těchto dvou kritérií rozlišujeme tyto poddruhy [8]:

- *Real-time strategy* (RTS) - reálný čas, strategická rozhodnutí
- *Real-time tactics* (RTT) - reálný čas, taktická rozhodnutí
- *Turn-based strategy* (TBS) - diskretní tahy, strategická rozhodnutí
- *Turn-based tactics* (TBT) - diskretní tahy, taktická rozhodnutí

V následujících částech popíšeme rozdíly mezi těmito poddruhy strategických her a vymezíme podmnožinu, jejíž vývoj bude naše platforma podporovat.

## Real-time strategy

Cílem této práce je vytvořit platformu umožňující tvorbu Real-time strategy (RTS)<sup>1</sup> her. RTS, v překladu strategické hry probíhající v reálném čase, jsou poddruhem strategických her ve kterém se změny stavu odehrávají v přímé závislosti na změně času v reálném světě. Reakce v reálném čase jsou náročnější jak pro hráče, který je často nucen použít suboptimální strategii, tak pro hru samotnou, která musí provádět výpočet dalšího stavu v omezeném čase. Stejně tak umělá inteligence, jakožto součást hry, musí reagovat na aktivity zbylých hráčů s omezeným časem, což limituje množství dat a složitost výpočtu, které může umělá inteligence použít. Z tohoto důvodu je vývoj RTS her složitým procesem, spojujícím mnoho oborů, který se pokusíme zjednodušit vytvořením naší platformy.

Zbylé poddruhy strategických her neplánujeme v naší platformě explicitně podporovat, ale nijak nevylučujeme, že bude možné do jisté míry využít naší platformu i pro tvorbu těchto poddruhů strategických her. Zároveň je pro tvorbu RTS her výhodné znát příbuzné žánry, z kterých je možno se inspirovat a přebírat některé z jejich mechanik. Z tohoto důvodu ve zkratce popíšeme i zbylé poddruhy.

## Real-time tactics

Prvním příbuzným žánrem jsou *Real-time tactics* (RTT) hry, někdy také nazývány fixed-unit real-time hry [7], neboli hry s pevným počtem jednotek probíhající

---

<sup>1</sup>Název "real-time strategy", poprvé použitý při propagaci hry Dune II, je připisován prezidentu a spoluzakladateli Westwood Studios, *Brettu Sperrymu*. Toto studio následně využilo zkušenosti získané při tvorbě Dune II pro vývoj jedné z nejznámějších sérií RTS her, Command & Conquer.

v reálném čase. Hlavním rozdílem, odlišující RTT od RTS, je omezení strategických rozhodnutí a větším důrazem na taktická rozhodnutí a micromanagement jednotlivých jednotek. RTT hry nedovolují hráči tvorbu nových jednotek, stavbu budov či produkci surovin, hráč je tedy nucen s jednotkami, které má na začátku souboje, vyhrát celý souboj. Jedním z příkladů čistě RTT her je série Blitzkrieg. Hráč začíná každou misi s jednotkami, které si vybral před misí. Tyto jednotky jsou jediné, které bude moci v průběhu mise využít, což nutí hráče použít svých taktických schopností a maximalizovat účinnost těchto jednotek.

## Turn-based strategy

*Turn-based strategy* jsou strategické hry, ve kterých změny stavu probíhají v diskrétních tazích. Doba mezi tahy často není nijak omezena, což hráči umožňuje vymyslet optimální strategii. Oproti RTS tyto hry často omezují taktickou část problémů a umožňují hráči soustředit se výhradně na strategickou část hry, tedy plánování budov, produkci surovin, vývoj technologií a celkovou strategii pro jeho ekonomiku a armádu. Příkladem tohoto žánru je série her Civilisation[3]. Naše platforma nebude explicitně podporovat tahy, myslíme si však, že bude možné toto rozdělení do tahů vytvořit použitím platformou poskytovaných prostředků.

To do  
(1)

## Turn-based tactics

Turn-based tactics umožňují hráči přímé ovládání několika málo jednotek, které v každém tahu mohou provést jeden či více úkonů. Těmito úkony mohou být střelba na nepřátelskou jednotku, vyléčení přátelské jednotky, pohyb po mapě či například zničení terénu. Ukázkovým příkladem je série her X-COM. Ve hře X-COM 2 hráč vlastní až malé desítky jednotek, z kterých vybírá malou skupinu a vysílá ji na jednotlivé mise. Při misi má každá jednotka v každém tahu možnost vykonat dvě akce. Akce může být přesun o omezenou vzdálenost, použití schopnosti či útok na nepřítele. Pro normální jednotky útok ukončuje tah dané jednotky a hráč může provést tah následující.

## Shrnutí

Jak můžeme vidět, žánr strategických her zahrnuje hry s velmi rozmanitými vlastnostmi a požadavky. Z tohoto důvodu se naše práce zaměří na podporu vývoje her jednoho konkrétního poddruhu, a to RTS. Přestože naše platforma bude cílena na tvorbu tohoto poddruhu, nevylučujeme, že bude možné využít ji i pro tvorbu her spadajících do jednoho ze zbylých poddruhů strategických her.

## 1.1 RTS hry blíže

Jak bylo řečeno výše, naše platforma bude navržena pro podporu vývoje RTS her. RTS hry jsou ovšem stále příliš velká množina s příliš rozmanitými mechanikami na to, aby jedna platforma dokázala podporovat všechny možné RTS hry. Proto zde dále omezíme námi podporovanou podmnožinu RTS her.

Již od svého vzniku na konci osmdesátých let a začátku devadesátých let minulého století obsahovaly RTS hry několik konceptů, které lze nalézt v drtivé většině her tohoto žánru i dnes.

Těmito koncepty jsou:

- Výroba a ovládání jednotek s cílem ovládnutí části mapy a zničení nepřátelských jednotek a budov
- Stavba budov pro umožnění stavby nových druhů budov, jednotek či získání surovin
- Získávání surovin pro stavbu jednotek a budov
- Výzkum nových technologií

Tyto koncepty, jako základní kámen RTS her, se bude naše platforma snažit podporovat a zjednodušit tvůrcům her jejich implementaci.

Hlavní inspirací pro tvorbu platformy, a tím i pro typ her, které bude platforma nejjednodušeji podporovat, byla hra Stronghold Crusader. Na této a dalších hrách ukážeme v následujících částech blíže základní principy RTS her a námi podporované implementace těchto principů.

## Strategie vs. Taktika

Jedním z hlavních problémů RTS her je pečlivé vyvážení kombinace strategie/macromanagementu a taktiky/micromanagementu. Tyto pojmy bývají často špatně chápány a někdy zaměňovány, pokusíme se je proto konkrétně definovat. Strategii míníme rozhodnutí týkající se globálního průběhu hry. Taktika naopak zahrnuje konkrétní pozice konkrétních jednotek, jejich pohyb po mapě a spolupráci v jedné bitvě.

Mezi strategická rozhodnutí v RTS hrách patří kupříkladu které budovy hráč postaví, v jakém pořadí dané budovy postaví, které suroviny bude produkovat, které suroviny vynechá, které jednotky bude rekrutovat a které vynechá. Tato 3 rozhodnutí jsou úzce propojena, protože výběr surovin určuje budovy, které bude hráč schopen postavit a typy jednotek, které bude moci zrekrutovat. Stejně tak výběr budov určuje suroviny, které hráč může produkovat a typy jednotek, které může zrekrutovat.

Taktika/micromanagement je v RTS hrách reprezentován ovládáním jednotek, jejich přesné pozice, pohybu, směru útoku, používání schopností atd. Micromanagement lze ale vidět i v ekonomické stránce hry, kdy hráč

Nejlepším příkladem pro rozlišení pojmu strategie a taktiky je série her Total War, které kombinuje mód tahové strategie s módem real-time tactics (RTT). V jedné části hry hráč přebírá kontrolu nad celým svým národem, rozhoduje, které budovy budou ve kterých městech postaveny, které jednotky budou rekrutovány a kde budou které armády umístěny. V druhé části, při souboji nepřátelských armád, hráč přebírá kontrolu nad konkrétní armádou a ovládá jednotlivé bataliony, jejich umístění, pohyb a útoky v reálném čase.

Naše platforma nemá za cíl nijak omezovat možnosti vytvářených her v rámci jejich zaměření na taktiku či strategii. Toto rozhodnutí chceme nechat čistě na uživateli naší platformy.

To do

(3)

To do

(4)

To do

(5)



### 1.1.1 Mapy

Reprezentace herní mapy je jedním z hlavních rozhodnutí při tvorbě RTS her. Hlavní funkcí herní mapy je reprezentovat terén pro pohyb jednotek a stavbu budov. Tato funkce může zahrnovat neprostupné části mapy, několik různých druhů terénu prostupné různým druhům jednotek či části mapy ovlivňující rychlost pohybu jednotek. Při stavbě budov je pak často omezen typ terénu, na kterém je hráč schopen danou budovu postavit. Vzhledem k uzavřenosti většiny RTS her je složité zjistit, jak je v každé z nich herní svět reprezentován, z pozorovatelného vnějšího chování lze ale odvodit několik základních druhů reprezentací.

Nejstarší a nejjednodušší reprezentací je rozdělení mapy na stejně velké dlaždice. Tyto dlaždice mohou být různých tvarů, nejčastěji jsou však čtvercové či hexagonální. Příkladem takovéto reprezentace je právě hra Stronghold Crusader, podle které chceme naši platformu modelovat. Jak je vidět na 1.1, herní mapa je viditelně rozdělena na stejně velké čtvercové dlaždice. Každá dlaždice má určen typ, který určuje její vzhled, což můžeme vidět části zvýrazněné červenou barvou. Vidíme zde řady pěti dlaždic stejného typu, oddělené vždy jednou dlaždicí pouštního typu. Můžeme si všimnout, že i dlaždice stejného typu mohou mít několik různých vzhledů. Typ dlaždice je dále využíván jako omezení při stavbě budov, kde například kamenolom lze postavit pouze na dostatečném počtu dlaždic typu kámen. Toto můžeme vidět v části označené modrou barvou, kde se hráč pokouší umístit budovu, která ovšem nemůže být postavena na dlaždicích typu mokřadu. Dlaždice tohoto typu jsou proto zvýrazněny červenou barvou v rámci půdorysu budovy. Dále můžeme na ukázce vidět dlaždice s rozdílnou výškou, tvořící nedostupnou oblast mapy.



Obrázek 1.1: Ukázka dlaždic ve hře Stronghold Crusader

Na každé dlaždici může být postavena až na výjimky nejvýše jedna budova a stát nejvýše jedna jednotka. Při pohybu jednotek není tato vlastnost dodržována, lze tedy jednotky přesouvat přes dlaždice na kterých již jiná jednotka stojí.

Naše platforma bude podporovat rozšířenější verzi tohoto druhu mapy, ve které nebudeme vynucovat limity na počty jednotek a budov na jedné dlaždici. Tato omezení budou přenechána pro implementaci tvůrcem her využívajících naši platformu a bude vytvořeno rozhraní pro co nejjednodušší implementaci těchto limitů.

Naše platforma bude podporovat mapu s těmito vlastnostmi:

M1: Terén rozdělený na čtvercové dlaždice.

Další  
druhy  
(6)

M2: Dlaždice s různou výškou.

M2: Dlaždice různých typů.

M3: Možnost dotazovat se na jednotky nacházející se na dlaždici.

M4: Možnost dotazovat se na budovy postavené na dlaždici.

### 1.1.2 Jednotky

Jednotky jsou základním nástrojem hráče pro boj s nepřítelem. Pohybem po mapě, poškozováním ostatních jednotek a ničením budov jednotky umožňují hráči vést souboj s protivníkem, získat strategickou výhodu a následně vyhrát hru.

#### Pohyb

Hlavním odlišujícím prvkem jednotek od budov je jejich schopnost pohybu. Pohyb jednotky je určen jejím typem, kde každý typ jednotek může procházet jinými typy terénu, pohybovat se nad terénem, na vodě či pod vodou. Naše platforma bude podporovat pohyb jednotek kdekoli nad terénem, navíc bude tvůrcům poskytnuta komponenta umožňující chůzi po terénu, neboť je to nejčastější způsob pohybu.

Pohyb jednotek je nejčastěji řízen hráčem, ať už na úrovni příkazů jednotlivým jednotkám, tak na úrovni slučování jednotek do skupin a ovládání těchto skupin. Naše platforma bude umožňovat jak ovládání jednotlivých jednotek, tak celých skupin. Dále umožníme vývojáři přidat složitější ovládání, například slučování do permanentních formací a následné ovládání těchto formací.

V některých hrách existují také jednotky, které se mohou své schopnosti pohybu vzdát a stát se budovu, buď dočasně, nebo trvale. Příkladem takového jednotky/budovy mohou být budovy Nočních elfů ze hry Warcraft 3 [?]. Tyto entity jsou stavěny jako budovy, tedy jsou umístěny do světa a postaveny jednotkou, stejně jako u všech ostatních ras. Následně je ale hráči umožněno, pomocí speciální ability těchto budov, změnit je dočasně na jednotky a pohybovat s nimi či je dokonce použít pro boj. V tomto módu ale ztrácí všechny funkce budov, tedy není možné je použít pro sběr surovin či produkci jednotek. Následně je možné znovu je zakořenit, čímž získávají zpět své funkce budovy. Naše platforma by měla takového jednotky/budovy také podporovat.

To do  
(7)

#### Umělá inteligence

Ve velké části RTS her jsou jednotky schopny do určité míry autonomního rozhodování bez zásahu hráče, od střelby na cíl, který se ocitne v jejich dostřelu, po vyhledání krytu, pokud jsou pod palbou.

Jako příklad jednoduché umělé inteligence jednotek můžeme vzít hry Starcraft a Warcraft od společnosti Blizzard. Zde se jednotky chovají velice předvídatelně, splňují přesně hráčovi rozkazy a nedělají nic navíc, což umožnilo hře Starcraft II vytvořit jednu z prvních masivních e-sport scén na světě. [4]

Dobrým příkladem jednotek s vysokou autonomií je série Company of Heroes, kde jednotky automaticky vyhledávají krytí, rozutečou se, pokud jsou pod palbou dělostřelectva, a v případě příliš velkých ztrát utečou z boje. Tato autonomie má

To do  
(8)

ale svou cenu, a to v nepředvídatelnosti chování jednotek. Při jednoduché umělé inteligenci jednotek je hráč schopen předvídat jejich chování a využít ho pro svůj prospěch. Naopak při složité umělé inteligenci, jako právě v případě *Company of Heroes*, je hráč často nucen provést více pokusů při vydávání rozkazu, protože není schopen jednoduše odhadnout chování jednotky. Tato skutečnost činí hry často realističtější, protože simuluje chování reálných vojáků, kteří rozkaz interpretují a implementují podle svého, není ale vhodná pro souboje více hráčů, a už vůbec ne více hráčů na profesionální úrovni.

Platforma bude umožňovat vykonat libovolný kód v rámci každého výpočtu stavu každé jednotky, bude tedy pouze na vývojáři, zda se budou jednotky chovat jednoduše a předvídatelně, nebo zda budou vykonávat složité, avšak nepředvídatelné úkony bez hráčova vědomí. Pro ulehčení vývoje bude platforma poskytovat předpřipravené komponenty, umožňující základní úkony jako střelbu na cíl, pohyb po mapě a útok na blízko.

## Produkce

Jednou z vlastností definujících RTS hry je možnost produkce nových jednotek. Existuje několik systémů produkce jednotek, úzce svázaných se systémem surovin v dané hře. (viz. 1.1.4) Od kontinuální produkce, kde hráč zvolí produkováné jednotky a suroviny jsou spotřebovávány v průběhu produkce, po diskrétní produkci, kde hráč musí vlastnit všechny suroviny potřebné pro výrobu dané jednotky při začátku produkce a všechny suroviny jsou odečteny v jeden okamžik. Kontinuální systém umožňuje hráči naplánovat produkci armády v předstihu, i když v daném okamžiku nevlastní dostatečné suroviny. Naopak při diskrétní produkci je hráč nucen čekat do chvíle, kdy má všechny suroviny, a až poté může začít s produkcí. Naše platforma se pokusí podporovat oba systémy. Bude záležet pouze na tvůrci hry, jak se k surovinám a produkci jednotek zachová a který z těchto systémů bude implementovat.

Počet jednotek je často limitován, jak pro účely vyvážení hry, tak pro omezení zátěže hardwaru. Z hlediska vyvážení síly jednotek umožňuje limit na počet jednotek předejít tzv. "Zergu", kdy hráč vytvoří obrovské množství levných jednotek, které následně převálcují jakýkoli odpor. Z hlediska hardwarové náročnosti je účel limitu vcelku zřejmý, protože každá jednotka zabírá určité množství paměti a výpočetního výkonu. *Stronghold Crusader* omezuje počet jednotek na 1000 pro každého hráče. Toto omezení se jeví především jako limit na hardwarovou náročnost hry. Naše platforma žádné explicitní limity nestanovuje, avšak v uživatelské dokumentaci pro vývojáře budeme silně doporučovat stanovení limitů na počet budov, jednotek a projektilů. Za tímto účelem umožníme vývojáři při vytvoření každé jednotky, budovy či projektilu učinit rozhodnutí, zda je vytvoření možné a případně toto vytváření zrušit.

Hráč často začíná s malým počtem jednotek, jejichž účelem je zamezit tzv. "Rush" strategii, ve které je cílem vytvořit co nejrychleji co možná nejvíce levných jednotek a zničit nepřítele ještě před tím, než je schopen začít produkovat své jednotky. Ve hře *Stronghold Crusader* hráč začíná každou hru s několika lučišníky a kopiníky, jejichž počet je určen v nastavení před začátkem hry. Toto bude v naší platformě umožněno přidáváním jednotek v rámci editace mapy, případně bude tvůrce hry schopen umožnit hráči určit počty jednotek před začátkem hry pomocí grafických elementů v uživatelském rozhraní a následně při začátku hry

vytvořit požadované množství jednotek. Tyto jednotky bude poté hráč vlastnit již na počátku hry.

## Boj

V drtivé většině RTS her mají jednotky tzv. “hit pointy”, zkráceně *HP*, které určují počet zásahů, které může jednotka obdržet než bude zabita. S každým zásahem jsou poté tyto *HP* odečítány a v okamžiku, kdy je jednotka poškozena na 0 *HP* je zabita. Naše platforma bude tento systém samozřejmě podporovat, ale nebude ho nijak explicitně vyžadovat, bude tedy tvůrci umožněno použít jakýkoli jím implementovaný systém.

Jednotky mohou obdržet poškození z mnoha zdrojů, nejčastěji však útokem z blízka (tzv. *meele*) či z dálky (tzv. *ranged*). Útok na blízko je omezen dosahem, rychlostí útoků a velikostí uděleného poškození. Naše platforma bude podporovat komponentu poskytující útoky na blízko právě s těmito parametry. Útok na dálku lze rozdělit do dvou typů, tzv. *hit-scan* a *projektily*. První typ je reprezentován například laserovými zbraněmi, které v okamžiku výstřelu urazí celou vzdálenost, dokud nenarazí na terén či nějakou entitu (budovu či jednotku). Druhý typ v okamžiku výstřelu vytvoří projektil, který se v průběhu času pohybuje herním světem, dokud také nenarazí na terén či nějakou entitu. Naše platforma bude podle předlohy Strongholdu Crusader podporovat především projektilové útoky. Za tímto účelem bude vytvořena komponenta umožňující střelbu projektilů, dále projektily samotné, simulace jejich letu a především možnost výpočtu pro střelbu na pohyblivý cíl. Hit-scan útoky nebudou přímo podporovány, mělo by však být umožněno tvůrci hry tento typ útoků implementovat manuálně.

## Shrnutí požadavků

Naše platforma bude podporovat jednotky s těmito vlastnostmi:

- J1: Pohyb jednotek volně kdekoliv nad terénem.
- J2: Podpora pohybu po terénu.
- J3: Ovládání jednotek a skupin jednotek.
- J4: Rozšiřitelnost o složitější ovládání.
- J5: Podpora jednoduché i složité umělé inteligence v podobě vykonání libovolného kódu.
- J6: Podpora diskrétní i kontinuální produkce jednotek.
- J7: Přidávání jednotek při editaci mapy.
- J8: Přidávání jednotek při startu hry.
- J9: Podpora systému hit-pointů.
- J10: Útoky na blízko i na dálku.
- J11: Projektily.

### 1.1.3 Budovy

Stavba budov představuje jednu z hlavních prezentací hráčovi strategie. Podle postavených budov lze často vcelku přesně odhadnout, jakou strategii hráč zvolil, čímž je umožněno nepřítelům reagovat a adaptovat svou strategii odpovídajícím způsobem. Při volbě strategie lze ale narazit na problém, kdy je hráč nucen zvolit svou strategii před tím, než nalezne protivníky a tedy před tím, než může vidět jejich strategii. Tento problém je velmi výrazný při tzv. “rock-paper-scissors” strategiích, kde strategie 1 poráží strategii 2, strategie 2 poráží strategii 3 a strategie 3 poráží strategii 1. V tuto chvíli hra degeneruje v loterii, zda hráč náhodně vybere správnou strategii porážející tu vybranou nepřítelem.[1] Jedním z řešení tohoto problému je co nejmenší rozdíl v síle prvních úrovní technologie, což umožní hráčům reagovat a změnit svoji strategii před tím, než je rozdíl mezi jejich silami neúnosně velký. [6] Dalším možným řešením, použitým ve Stronghold Crusader, je neskryvat před hráčem nepřítelovu strategii. Toto řešení lze použít v různé míře, od odhalení nepřátelských budov po úplné odkrytí celé mapy, tedy pozice všech jednotek i budov, ať už přátelských, nepřátelských či neutrálních. Naše platforma použije toto poslední řešení, kdy budeme hráč vidět všechny jednotky a budovy všech hráčů.

Reference  
(10)

Budovy mají ve hrách mnoho funkcí, mezi které patří například:

- Produkce jednotek
- Vylepšování jednotek
- Produkce surovin
- Uskladnění surovin
- Obrana
- Stavba budov
- Zkoumání technologií

Naše platforma se pokusí co nejvíce zjednodušit implementaci těchto funkcí poskytnutím programátorského rozhraní pro umístění budov a jednotek do herního světa, přidání a odebrání surovin hráči či střelbu projektilů. Dále umožníme tvůrci her přístup ke grafickému rozhraní, do kterého bude možné umístit okna, tlačítka a další elementy, které následně použitím programátorské rozhraní budou schopné implementovat všechny zmíněné funkce budov.

Jak již bylo řečeno v sekci o jednotkách 1.1.2, naše platforma bude nechávat volbu mezi kontinuální a diskrétní produkcí jednotek na tvůrci hry. Budeme se tedy snažit do co největší míry podporovat oba tyto způsoby.

#### Obrana

Obrana bude podporována v podobě komponent, které umožní budovám jak útok na blízko, jako v případě pastí ve hře Stronghold Crusader, tak na dálku, jako například obranné věže ve hře Warcraft 3.

Dále v rámci funkce obrany umožňují v některých hrách budovy jednotkám pohybovat se po nich. Jak můžeme vidět na obrázku ?? ze hry Stronghold Crusader, jednotky mohou být umístěny na hradbách, věžích, bránách či na tvrzi. Jak bylo řečeno v části o jednotkách, naše platforma bude umožňovat neomezený pohyb jednotkami, tedy i ve vzduchu, na budovách či skrz budovy. Protože se pohyb po budovách vyskytuje ve velkém množství RTS her, poskytne platforma možnost rozšíření terénu o plochu budov a komponentu umožňující chůzi po těchto plochách.

Umístění na budovách poskytuje jednotkám ochranu před nepřátelskými jednotkami útočícími na blízko, které se musí nejdříve dostat do blízkosti cíle, než zaútočí.

V některých hrách, kde bohužel Stronghold Crusader není jednou z nich, poskytuje vyvýšení nad terén jednotkám větší dostřel. Tento efekt nemusí být omezen pouze na vyvýšení pomocí budov, ale lze ho dosáhnout už při rozdílných výškách terénu, na kterém je umístěn střelec a jeho cíl, obecněji na rozdílu výšky pozice střelce a cíle, pokud se střelec nemusí pohybovat přímo po terénu. Naše platforma bude podporovat realistické chování projektilů splňující tuto vlastnost.

Budovy, podobně jako jednotky, mají ve většině her určitý počet tzv. "hit pointů", které určují počet zásahů, které může budova obdržet před tím, než bude zničena. Navíc oproti jednotkám mohou budovy často obdržet poškození pouze od omezené podmnožiny jednotek, nejčastěji pouze obléhacích strojů. Stejně jako u jednotek přenecháme systém poškození na tvůrci hry. Naše platforma pouze umožní budově reagovat na zásah projektilem či zbraní, ať už snížením svých *HP*, nebo ignorováním daného útoku v případě že přišel od jednotky či projektilu, který danou budovu nemůže poškodit.

zkontrolovat  
(11)

## Stavba budov

Budovy mají často restrikce, které je hráč nucen splnit před stavbou budovy. Tyto restrikce mohou sahát od reliéfu a typu terénu, přes existenci jiných budov ve stejném místě, po vlastnictví určitého množství surovin či typu jednotek. Naše platforma umožní tvůrci před stavbou budovy zjistit stav všech těchto typů restrikcí a případně vetovat stavbu budovy.

Jednou z restrikcí je výzkum určité technologie či stavba určité předcházející budovy<sup>1.1.5</sup>. Tímto způsobem jsou budovy uspořádány do postupně se zlepšujících úrovní, které hráč v průběhu času odemyká. Každá z úrovní obsahuje řadu rozdílných budov, umožňujících zvolit různé strategie. Platforma bude umožňovat tvůrci postupné zpřístupňování budov a jednotek, čímž bude tvůrce schopen implementovat postupné zkoumání nových technologií.

Existující hry využívají několik možností, jak hráči poskytnout zpětnou vazbu o splnění restrikcí při stavbě budovy. Jednoduší možností, použitou ve hře Stronghold Crusader, je zobrazení půdorysu v různých barvách podle splnění restrikcí. Další, složitější možností je zobrazení průhledného či jinak upraveného modelu budovy na místech, kde ji nelze postavit. Naše platforma bude podporovat jednodušší způsob, tedy zobrazení půdorysu budovy v různých barvách podle požadavků tvůrce hry.

## Shrnutí požadavků

Naše platforma bude podporovat tyto vlastnosti:

- B1: Stavbu budov v herním světě.
- B2: Komponenty pro útok budov na blízko i na dálku.
- B3: Rozšiřitelnost dostupného terénu v herní mapě o prostor na budovách.
- B4: Podpora zvýšení dostřelu při umístění jednotky na vyvýšený terén, například budovu.
- B5: Stavba budov při editaci mapy.
- B6: Stavba budov při startu hry.
- B7: Podpora systému hit-pointů.
- B8: Tvůrcem definované reakce na obdržení zásahu budovou.
- B9: Zobrazení půdorysu v různých barvách pro zpětnou vazbu splnění restrikcí.
- B10: Kontrolu požadavků při stavbě budovy.
- B11: Přístup ke grafickému rozhraní, možnost zobrazení elementů hráči.

### 1.1.4 Suroviny

“Resource management”, tedy management surovin, je přítomný ve všech hrách žánru RTS již od jeho vzniku. Od koření v Dune II, přes zlato a dřevo ve Warcraft 3, po všechny typy surovin ve hře Stronghold, získávání surovin je jednou z hlavních motivací konfliktu v RTS hrách.

Systémy surovin lze rozdělit podle způsobu získávání a počtu typů surovin.

Podle způsobu získávání můžeme systém surovin rozdělit na

1. Aktivní získávání surovin
2. Pasivní získávání surovin

Při aktivním získávání surovin existuje ovladatelná herní entita, která svým pohybem mezi pozicemi na mapě přináší suroviny. Tento pohyb může být ovládán hráčem, ale nejčastěji dokáže pracovat jednotka samostatně. Příkladem může být Warcraft 3 [2], kde speciální jednotky získávají dřevo a zlato přenášením z lesů/dolů do hráčovy hlavní budovy. Při pasivním získávání přibývají suroviny bez akcí entit, pouze díky vlastnictví určité části mapy nebo druhu budovy. Zdroj bývá nekonečný nebo skoro nekonečný, poskytující suroviny do obsazení nebo zničení zdroje. Každý z těchto stylů podporuje jinou strategii kontroly mapy.

Naše platforma bude podporovat jak pasivní, tak aktivní získávání surovin. Bude pouze na tvůrci, v jakém okamžiku budou suroviny přičteny, ať už v závislosti na čase nebo na pohybu určitých jednotek.

## Shrnutí požadavků

Naše platforma bude podporovat tyto vlastnosti:

S1: Podporovat aktivní i pasivní získávání surovin.

S2: Poskytnout rozhraní pro přidání a odebrání surovin hráči.

### 1.1.5 Vývoj technologií

Volba vyzkoumaných technologií důležitou součástí strategické části RTS her.

Technologie jsou často uspořádány ve stromové struktuře či orientovaném acyklickém grafu (DAG), kde vyzkoumání technologie v rodičovském uzlu odemkává technologie následujících uzlech. Příklad takového uspořádání můžeme vidět v ukázce ze hry *Civilisation V* [3]1.2, kde vidíme počátek stromu technologií. V této hře jsou technologie uspořádány do DAGu, začínajícího v jednom kořeni. Můžeme vidět žluté vrcholy, značící vyzkoumané technologie, dále zelené vrcholy, značící technologie, které mají splněné všechny předky a mohou být vyzkoumány, černé technologie, které bude možné začít zkoumat po odemčení všech předků, a nakonec červené technologie, značící technologie v dalším věku. Dále můžeme vidět hrany spojující závislé technologie.

Další možné uspořádání je několik disjunktních stromů technologií, kdy je hráč nucen zvolit jeden z těchto stromů. Toto uspořádání můžeme vidět ve hře *Company of Heroes* [? ], kde jsou hráči dostupné tři vzájemně výlučné cesty, každá zaměřená na jinou oblast boje. Každá z těchto cest je dále rozdělena na dvě větve postupně se zlepšujících technologií.

Každé větvení v grafu technologií představuje možné rozhodnutí hráče, kterou z větví se hráč vydá a které technologie odemkne. Toto rozhodnutí jsou jedním z hlavních projevů hráčovi strategie.

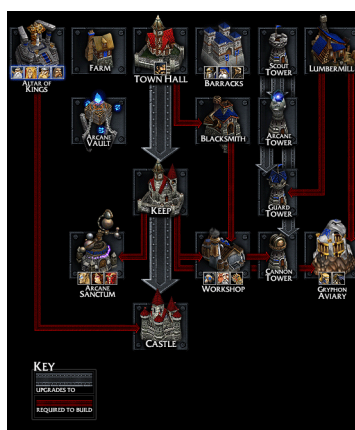
Vyzkoumání technologie může mít mnoho různých efektů. Nejčastějším efektem bývá odemknutí nového typu jednotek nebo budov. Další možností je změna vlastností již vlastněných jednotek nebo budov. V neposlední řadě pak může vyzkoumání technologie odemknout nové schopnosti nebo kouzla, které hráč může následně použít při taktických soubojích.



Obrázek 1.2: Výřez stromu technologií ze hry *Civilisation V*



Explicitní strom technologií, viditelný na 1.2, se v RTS hrách vyskytuje spíše výjimečně. Nejčastěji je odemykání nových typů jednotek a budov umožněno stavbou určitého typu budovy nebo dosažení určitého stupně vylepšení již existující budovy. Jako příklad můžeme vzít Warcraft 3 [2], kde závislosti budov na stupních vylepšení a existenci jiných budov tvoří strom technologií. Graf tvořený těmito závislostmi můžeme vidět na obrázku 1.3. Zde vidíme stupně vylepšení budov, reprezentované šedými šipkami, a závislosti budov na ostatních budovách a stupních jejich vylepšení. Aby byl hráč schopen postavit budovu, musí vlastnit všechny budovy na kterých je tato budova závislá v požadovaném nebo lepším stupni vylepšení. Ve hře je tento graf reprezentován požadavky zobrazenými při najetí myši na ikonu zamčené budovy.



Obrázek 1.3: Budovy a závislosti mezi nimi tvořící obdobu stromu technologií ve hře Warcraft 3

Jak bylo řečeno v sekci o budovách 1.1.3, naše platforma bude umožňovat tvůrci postupné odemykání budov a jednotek, což umožní tvůrci her implementovat výzkum nových technologií. Pro technologie měnící vlastnosti typů jednotek bude pouze na tvůrci, aby změnil logiku hry a chování umělé inteligence v závislosti na vyzkoumaných technologiích.

## Shrnutí požadavků

Naše platforma bude podporovat tyto vlastnosti:

T1: Umožnit postupné odemykání budov a jednotek.

T2: Změny chování a vlastností jednotek v průběhu hry.

## 1.2 Uživatelé

V předešlé sekci jsme popsali druh her, který bude naše platforma podporovat. V této sekci popíšeme požadavky na chování platformy z pohledu všech druhů našich uživatelů.

### 1.2.1 Tvůrci her

Prvním druhem uživatele budou tvůrci her, používající naši platformu pro zjednodušení své práce a vyřešení problémů opakujících se ve většině RTS her. Tvůrce samozřejmě nemusí být pouze jeden člověk, vzhledem ke složitosti RTS her existuje při jejich tvorbě mnoho rolí, které požadují velmi různorodé schopnosti a mohou být plněny více lidmi. Příkladem rolí může být 3D umělec tvořící modely a animace, producent hudby, tvůrce umělé inteligence, programátor herní logiky a nakonec člověk, který toto vše integruje dohromady.

#### Programátor

Naším hlavním cílem je umožnit tvůrcům umělé inteligence a programátorům herní logiky použít všechny jazyky .NET Frameworku, především C#, pro jejich práci. Z programátorského pohledu bude naše platforma sloužit jako knihovna poskytující tyto funkce:

1. Udržovat aktuální stav hry a umožňovat dotazy na tento stav.
2. Vytvářet nové jednotky, budovy a projektily.
3. Registrovat si metody, které budou zavolány při určitých událostech v průběhu hry.
4. Upravovat terén.
5. Vykreslovat terén, jednotky, budovy, projektily a grafické rozhraní.
6. Ukládat a načítat stav hry.
7. Implementace pro základní problémy jako pohyb po terénu, let projektilů, výpočet úhlu střelby projektilů. .
8. Ovládání kamery.

To do  
(15)

Protože implementace vykreslování a grafického rozhraní by byla nad rámec jedné bakalářské práce, využije naše platforma pro tyto účely existující herní engine UrhoSharp. Pro programátory následně poskytneme přístup k relevantním částem UrhoSharp enginu.

Platforma bude poskytovat ovládání kamery v několika módech. Tato funkcionalita bude potřebná již pro implementaci editoru, poskytneme ji tedy i tvůrcům her. Prvním módem bude klasická RTS top-down kamera, jakou můžete vidět na ukázce ze hry Company of Heroes ???. V této hře je oproti jiným starším RTS hrám jako Warcraft 3 nebo Age of Empires 3 s kamerou možné rotovat a přibližovat, což nám přijde jako velice atraktivní. Toto bychom rádi podporovali i v naší platformě. Druhým módem kamery bude tzv. “free-float”, kdy kamera volně “létá” nad terénem. Třetím módem bude sledování jednotky, kdy se kamera bude chovat jako v prvním módu, její pohyb bude ale řízen pohybem jednotky.

## Moddeři

Pro integraci vytvořených jednotek, budov, dlaždic a jejich modelů, textur a logiky umožní naše platforma vytvořit *balíček*, obsahující vše vytvořené programátory a umělci. Tento balíček poté platforma umožní přidat a načíst v libovolné instalaci platformy.

Platforma bude také sloužit jako editor úrovní, které budou využívat logiku, jednotky, budovy a typy terénu dodané pomocí balíčků. Vytvořené úrovně bude následně možné uložit zpět do balíčku a distribuovat spolu s balíčkem do dalších instancí naší platformy.

Pro editaci mapy poskytne platforma několik základních nástrojů a umožní tvůrcům modifikovat nástroje či přidávat své vlastní. Základní nástroje by měli umožnit editaci terénu mapy (změnu typu terénu na všechny tvůrce definované typy a editaci výšky dlaždic) a přidávání všech druhů jednotek a budov do mapy.

Formát ukládání úrovní bude definován otevřeně pomocí prostředků nezávislých na programovacím jazyce, čímž chceme umožnit tvorbu separátních editorů map produkujících úrovně v námi používaném formátu.

### 1.2.2 Hráči her

Z pohledu hráče se bude platforma chovat jako instalovatelná aplikace, která hráči umožní za běhu přidávat balíčky a následně využít jejich obsah.

Při běhu platforma umožní plný přístup k nastavení UrhoSharp enginu, tedy k nastavení rozlišení, Vsync, triple buffer a dalších. Dále umožní správu balíčků, tedy přidávání, odebírání a spouštění. Při spuštění balíčku umožní platforma hráči výběr z existujících map, jak pro hraní, tak pro editaci. Dále platforma umožní vytvoření a editaci úplně nové mapy.

Při tvorbě nové mapy či editaci existující mapy mít bude hráč přístup ke všem jednotkám, budovám a typům terénu, ke kterým mu dají editační nástroje specifikované tvůrcem balíčku přístup. Následně bude hráči umožněno mapu uložit, a to přepsáním zdrojové mapy, kterou hráč načel k editaci, nebo vytvořením nové mapy pod novým jménem.

## 1.3 Ukázková hra

Pro ukázkou bude vytvořen balíček s jednoduchou hrou demonstrující možnosti naší platformy. Ukázková hra bude zároveň sloužit jako referenční příklad použití naší platformy.

Ukázková hra bude obsahovat několik jednotek, demonstrujících tyto vlastnosti:

1. Jednoduchou a složitou umělou inteligenci
2. Plně automatické jednotky, neovladatelné hráčem
3. Jednotky útočící na dálku
4. Jednotky útočící na blízko
5. Aktivní získávání surovin

6. Pohyb po terénu
7. Pohyb nad terénem (létání)
8. Rozdílnou rychlost pohybu různých jednotek
9. Rozdílnou přístupnost částí mapy pro různé jednotky
10. Pohyb po budovách

Demonstrovanými vlastnostmi budov budou:

1. Restrikce na místo stavby
2. Neprostupnost budov pro některé jednotky
3. Produkce surovin budovami
4. Přidání plochy nebo části plochy budovy jako přístupný terén pro určité typy jednotek

Jako obecné vlastnosti bude ukázková hra demonstrovat:

1. RTS mód kamery
2. Volný pohyb kamery
3. Sledování jednotky kamerou
4. Tvůrcem definované prvky v uživatelském rozhraní
5. Minimapu

Balíček obsahující ukázkovou hru bude demonstrovat tyto vlastnosti:

1. Tvorbu vlastních úrovní za použití jednotek, budov a typů terénu obsažených v balíčku.
2. Ukládání a načítání hry.

## 1.4 Cíle práce

Cílem této práce je vytvořit platformu pro vývoj 3D RTS her pro jednoho hráče za použití herního engine UrhoSharp, umožňující vývojářům vytvářet hry jako separátně distribuované balíčky, které bude poté koncový uživatel schopen přidat do naší platformy nainstalované na uživatelské počítači a použít je pro hraní dodaných úrovní či tvorbu svých vlastních.

Při tvorbě hry bude umožněno tvůrci použít jazyky frameworku .NET pro vytvoření Umělé inteligence jednotek, budov a nepřátelských hráčů, pro vytvoření další logiky hry a pro přidání nástrojů do editoru map.

Požadované vlastnosti platformy:

1. Podporované vlastnosti jednotek:
  - (a) Pohyb jednotek ( $J_1$ ,  $J_2$ )

- (b) Ovládání jednotek (J3, J4)
  - (c) Umělá inteligence, definice chování (J5)
  - (d) Produkce jednotek (J6)
  - (e) Přidávání jednotek jako součást mapy (J7, J8)
  - (f) Podpora systému hit-pointů (J9)
  - (g) Útoky na blízko i na dálku (J10, J11)
2. Podporované vlastnosti budov:
- (a) Stavba budov v herním světě (B1, B9, B10)
  - (b) Podpora obraných budov (B2, B4)
  - (c) Rozšiřitelnost dostupného terénu o plochu budov (B3)
  - (d) Přidávání budov jako součásti mapy (B5, B6)
  - (e) Zničitelnost budov (B7, B8)
  - (f) Produkce jednotek, surovin, stavba budov pomocí budovy (B11)
3. Podpora surovin:
- (a) Přidávání a odebírání libovolného počtu surovin (nejen celočíselných) (S1, S2)
4. Podpora výzkumu technologií:
- (a) Umožnit postupné odemykání dostupných jednotek a budov, umožnit změny chování jednotek za běhu (T1, T2)
5. Vlastnosti pro tvůrce balíčků:
- (a) Platforma musí umožňovat přidávání balíčků za běhu, obsahujících nové typy jednotek, budov, dlaždic, projektilů a hráčů spolu s jejich modely, texturami a AI.
  - (b) Platforma musí umožňovat použití přidaných balíčků pro tvorbu map a uložení vytvořených map do balíčku použitého pro jejich tvorbu.
  - (c) Editor map musí být rozšiřitelný o nástroje z balíčku.
  - (d) Herní grafické rozhraní musí umožňovat tvůrci přidávat vlastní okna, tlačítka a další prvky.
6. Vlastnosti pro koncového hráče:
- (a) User interface pro stolní počítače, umožňující vybírání balíčků, map a oponentů, dále načítání a ukládání her, a nastavování zobrazení hry.
  - (b) Herní user interface musí obsahovat minimapu, poskytující hráči přehled o větší části mapy než kterou vidí vlastní kamerou.
  - (c) Ovládání kamery umožňující klasický top-down pohled, volné poletování kamery po mapě a následování jednotky.
  - (d) Ukládání a načítání hry.

## 2. Analýza

V první kapitole jsem specifikovaly náš cíl, tedy implementaci platformy nad enginem UrhoSharp umožňující tvorbu RTS her a jejich distribuci. V této kapitole popíšeme problémy při implementaci platformy a jejich možná řešení.

### 2.1 Herní engine

Jak jsme psali v Cílech práce (viz 1.4), naším cílem je vytvořit platformu za použití herního enginu UrhoSharp. “UrhoSharp je multiplatformní 3D a 2D engine který může být použit pro tvorbu animovaných 3D a 2D scén za použití modelů, materiálů, světel a kamer”[? ], jak říká úvodní stránka dokumentace enginu. Jak už název napovídá, UrhoSharp je .NET binding pro Urho3D engine [? ], což je opensource engine implementován v C++.

To do  
(16)

### 2.2 Podporované platformy

Jak bylo zmíněno v předešlé sekci, je námi používaný herní engine multiplatformní. Bohužel každá z platforem má určitá specifika a restrikce, jak obecně, tak v rámci enginu, které nás nakonec vedli k podpoře pouze platformy Windows.

#### 2.2.1 Mobilní platformy

Při implementaci pro mobilní platformy existuje několik problémů.

##### Zobrazení a ovládání

Na mobilních platformách je mnohem bližší vztah mezi GUI, tedy grafických uživatelským rozhraním, a ovládáním. Oproti platformě PC, ať už Windows či Linux, je zde jediným možným vstupem dotyková obrazovka. GUI musí tedy sloužit jak pro zobrazení informací hráči, tak pro získání vstupu od hráče.

Druhým problémem je velikost obrazovky, která je u drtivé většiny mobilních zařízení mnohem menší než u PC.

Oba tyto problémy nás vedou k separátní implementaci uživatelského rozhraní pro mobilní zařízení. Pro tuto implementaci jsme v naší práci připravily základní kostru, upustili jsme ovšem od konečné implementace z důvodu nedostatku času.

##### JIT vs. AOT

Dalším rozdílem, tentokrát s rozdílným chováním i mezi různými mobilními platformami, je jejich chování ke kódu. Na systémech Windows, Linux či Android je CIL kód přeložen za běhu pomocí JIT kompilátoru do instrukcí procesoru a následně vykonán. Oproti tomu systém *iOS* zakazuje namapování stránek paměti zároveň pro zápis a pro vykonání, čímž znemožňuje jakýkoli JIT překlad. Z tohoto důvodu musí být všechny aplikace pro systém *iOS* překládány tzv. AOT, tedy před tím, než jsou distribuovány k uživateli, přímo do instrukční sady cílového procesoru.[5] Tato skutečnost znemožňuje jednoduché nahrání assembly pomocí

reflexe a nutila by nás k složitějšímu řešení. Tvůrce balíčku by musel své kódy přeložit pro všechny možné architektury, a naše platforma by se následně při běhu musela podle platformy, na které běží, rozhodovat, kterou z assembly nahrát. Toto je jeden z důvodů, proč jsme upustily od implementace podpory mobilních zařízení.

## Souborové systémy

### 2.3 Formát balíčků a jejich načítání

Důležitou součástí implementace naší platformy je systém balíčků pro distribuci vytvořených her. Tyto balíčky obsahují všechny součásti hry, od modelů a textur, přes logiku a umělou inteligenci, po mapy a úrovně vytvořené tvůrcem hry. Všechny tyto součásti musí naše platforma být schopna načíst za běhu a použít jak pro tvorbu nových map, tak pro hraní již existujících.

#### 2.3.1 Základní struktura

Pro implementaci načítání balíčků musíme definovat strukturu, kterou budou balíčky splňovat, a podle které bude platforma určovat závislosti a typy souborů.

První možností je založit strukturu balíčku na adresářové struktuře, kde každý balíček bude tvořen jedním adresářem obsahujícím další pevně specifikované podadresáře. Každý z podadresářů by obsahoval jeden z typů zdrojů, tedy 3D modely, textury, popis jednotek nebo skripty.

Pro popis typů jednotek, budov, projektilů, dlaždic a nepřátel jsme se inspirovali v existujících hrách, ať už *Civilisation V* nebo *Kerbal Space Program*, a využili jsme XML soubor pro definici závislostí. Tento XML soubor

Každý balíček je reprezentován jedním XML souborem. Tento soubor má formát daný pevným schématem a tento formát je kontrolován při každém načítání.

### 3. Ukázková hra



## 4. Programátorská dokumentace

## 5. Uživatelská dokumentace

# Závěr

# Seznam použité literatury

- [1] ADAMS, E. (2009). *Fundamentals of Game Design*. New Riders, 2 edition. ISBN 0-321-64337-2.
- [2] BLIZZARD ENTERTAINMENT, INC. Warcraft 3. <https://playwarcraft3.com/en-us/>. Accessed:2019-03-18.
- [3] FIRAXIS GAMES, INC. (2010). Civilisation V. <https://civilization.com/civilization-5/>.
- [4] LANTZ, J. (2008). Opinion: The Evolution of the Modern RTS. [http://www.gamasutra.com/php-bin/news\\_index.php?story=18326](http://www.gamasutra.com/php-bin/news_index.php?story=18326).
- [5] O'BRIEN, L., SCHONNING, N., DUNN, C., UMBAUGH, B. a PAKALA, Y. (2017). iOS App Architecture. <https://docs.microsoft.com/en-us/xamarin/ios/internals/architecture>. Accessed:2019-04-08.
- [6] OXEYE GAME STUDIO (2008). RTS Game-play Part 3: Build Options. <http://www.oxeyegames.com/rts-game-play-part-3-build-options/>.
- [7] WALKER, M. H. (2004). Strategy Gaming: Part II. <https://web.archive.org/web/20070129065355/http://archive.gamespy.com/articles/february02/strategy02/>. [Online; seen 2019-04-01].
- [8] WALKER, M. H. (2004). Strategy Gaming: Part V – Real-Time vs. Turn-Based. <https://web.archive.org/web/20081201113359/http://archive.gamespy.com/articles/february02/strategygames05/index.shtm>. [Online; seen 2019-04-01].

## A. Přílohy

### A.1 První příloha

## To do...

- ☐ 1 (p. 3): popsat Civilisation
- ☒ 2 (p. 4): Možná vynechat Možná vynechat
- ☐ 3 (p. 4): citace
- ☐ 4 (p. 4): citace
- ☐ 5 (p. 4): možná graf následujícího, jak se ovlivňují
- ☐ 6 (p. 5): Další druhy popsat další druhy implementací
- ☐ 7 (p. 6): obrázek
- ☐ 8 (p. 6): obrázek s chováním
- ☒ 9 (p. 8): Přeuspořádat Lépe uspořádat sekci o budovách
- ☐ 10 (p. 9): Reference Reference na rock-paper-scissors
- ☐ 11 (p. 10): zkontrolovat zkontrolovat, že opravdu Stronghold Crusader není jednou z nich
- ☒ 12 (p. 10): zničitelnost
- ☒ 13 (p. 10): různé druhy poškození
- ☒ 14 (p. 10): restrikce na umístění
- ☐ 15 (p. 14): more
- ☐ 16 (p. 18): porovnání s dalšími enginy