

Katedra informatiky  
Přírodovědecká fakulta  
Univerzita Palackého v Olomouci

# BAKALÁŘSKÁ PRÁCE

Beat 'em up hra v Unity frameworku



2018

Vedoucí práce: Mgr. Martin tr-  
nečka, Ph.D.

Ema Csákányiová

Studijní obor: Aplikovaná informatika,  
prezenční forma

## **Bibliografické údaje**

Autor: Ema Csákányiová  
Název práce: Beat 'em up hra v Unity frameworku  
Typ práce: bakalářská práce  
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci  
Rok obhajoby: 2018  
Studijní obor: Aplikovaná informatika, prezenční forma  
Vedoucí práce: Mgr. Martin trnečka, Ph.D.  
Počet stran: 24  
Přílohy: 1 CD/DVD  
Jazyk práce: český

## **Bibliographic info**

Author: Ema Csákányiová  
Title: Beat 'em up game in Unity framework  
Thesis type: bachelor thesis  
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc  
Year of defense: 2018  
Study field: Applied Computer Science, full-time form  
Supervisor: Mgr. Martin trnečka, Ph.D.  
Page count: 24  
Supplements: 1 CD/DVD  
Thesis language: Czech

## **Anotace**

*Práce pojednává o implementaci počítačové hry pro jednoho až dva hráče v Unity Frameworku a technologiích použitých během tohoto procesu. Je zmíněna grafická stránka hry a popsány jednotlivé skripty.*

## **Synopsis**

*This thesis describes the implementation of a computer game for one to two players in Unity Framework and technologies used in this process. In the thesis the graphics of the game are mentioned and individual scripts described.*

**Klíčová slova:** Unity framework; videohra; lokální multiplayer

**Keywords:** Unity framework; videogame; local multiplayer

Děkuji vedoucímu práce za trpělivost a vstřícný přístup.

*Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracovala samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.*

datum odevzdání práce

podpis autora

# Obsah

<b>1</b>	<b>Úvod</b>	<b>7</b>
1.1	Motivace . . . . .	7
1.2	Název hry . . . . .	7
<b>2</b>	<b>Videohra</b>	<b>8</b>
2.1	Počítačová hra . . . . .	8
2.2	Beat 'em Up žánr . . . . .	8
2.3	Zástupci žánru . . . . .	8
2.3.1	Kung-Fu Master . . . . .	9
2.3.2	Streets of Rage . . . . .	9
2.3.3	Castle Crashers . . . . .	10
<b>3</b>	<b>Použité Technologie</b>	<b>11</b>
3.1	Unity Framework . . . . .	11
3.2	C# . . . . .	11
3.3	Piskel . . . . .	12
<b>4</b>	<b>Programátorská dokumentace</b>	<b>13</b>
4.1	Unity terminologie . . . . .	13
4.1.1	Asset . . . . .	13
4.1.2	Scéna . . . . .	13
4.1.3	Game Object . . . . .	13
4.1.4	Komponenta . . . . .	13
4.1.5	Prefab . . . . .	13
4.1.6	Skript . . . . .	13
4.2	Skripty aplikace . . . . .	14
4.2.1	Player skripty . . . . .	14
4.2.1.1	PlayerMovement.cs . . . . .	14
4.2.1.2	PlayerHealth.cs . . . . .	14
4.2.1.3	PlayerAttack.cs . . . . .	14
4.2.1.4	PlayerInput.cs . . . . .	15
4.2.2	Player Ability skripty . . . . .	15
4.2.2.1	ShieldBlock.cs . . . . .	15
4.2.2.2	PiercingArrow.cs . . . . .	15
4.2.2.3	ShadowWalk.cs . . . . .	15
4.2.2.4	TripleFireball.cs . . . . .	15
4.2.3	Enemy skripty . . . . .	15
4.2.3.1	EnemyHealth.cs . . . . .	15
4.2.3.2	EnemyBrain.cs . . . . .	16
4.2.4	Menu skripty . . . . .	16
4.2.4.1	MainMenu.cs . . . . .	16
4.2.4.2	MouseDisable.cs . . . . .	16
4.2.4.3	PauseMenu.cs . . . . .	17

4.2.5	Skripty pro řízení hry . . . . .	17
4.2.5.1	GameManager.cs . . . . .	17
4.2.5.2	PlayerManager.cs . . . . .	17
4.2.5.3	EnemyManager.cs . . . . .	17
4.2.6	Camera skripty . . . . .	17
4.2.6.1	CameraMovement.cs . . . . .	17
4.2.7	Utility skripty . . . . .	18
4.2.7.1	OnLoadScene.cs . . . . .	18
4.2.7.2	ChildTrigger.cs . . . . .	18
4.2.7.3	Projectile.cs . . . . .	18
4.2.7.4	HealthBar.cs . . . . .	18
4.2.7.5	EnemySpawnTrigger.cs . . . . .	18
4.2.7.6	EndStageTrigger.cs . . . . .	18
4.2.7.7	HideStagePanel.cs . . . . .	19
4.2.7.8	ShowEndPanel.cs . . . . .	19
	<b>Závěr</b>	<b>20</b>
	<b>Conclusions</b>	<b>21</b>
	<b>A Obsah přiloženého CD/DVD</b>	<b>22</b>
	<b>Literatura</b>	<b>23</b>

# 1 Úvod

Cílem této práce bylo vytvořit počítačovou 2D hru pro jednoho až čtyři hráče s využitím herního enginu Unity v jazyku C#. Původní návrh hry podporoval hru až čtyř hráčů po síti, ale během vývoje se ukázalo, že implementace je vzhledem k nedostatečné dokumentaci nad moje schopnosti. Po dohodě s vedoucím práce byla funkčnost omezena na lokální hru dvou hráčů.

Hra je zástupcem žánru Beat ‘em up s jednoduchým příběhem a designem jednotlivých úrovní. Příběh je čistě lineární, jak bývá u toho druhu her zvykem. Ovládání primárně počítá s herním ovladačem, ale vzhledem k platformě, na kterou je hra vyvíjena, je umožněno i ovládání pomocí klávesnice.

## 1.1 Motivace

Již několik let se zajímám o herní scénu a vývoj videoher. Ještě nikdy jsem však nekusila vytvořit vlastní herní svět. Bakalářská práce mi v tomto ohledu poskytla skvělou příležitost k prvnímu pokusu o naprogramování jednoduché počítačové hry. Zvolila jsem hru pro více hráčů, protože si osobně hry užívám více, když můžu hrát se svými přáteli.

Na herní scéně se, samozřejmě, vyskytuje mnoho titulů s možností hry více hráčů. Většina také nabízí více herních hodin a kvalitnější provedení než můj projekt, ale to mě nijak neodradilo od mého záměru. Chtěla jsem pouze vytvořit krátkou, zábavnou, nenáročnou a hlavně neplacenou hru pro sebe a své blízké přátele.

## 1.2 Název hry

Hra nese název Dark Bones podle nepřátel, kteří se ve hře vyskytují. Je taky narážkou na úspěšnou sérii Dark Souls od FromSoftware [1], známou svou obtížností.

## 2 Videohra

Videohra [2] je software, který na základě interakce s uživatelským rozhraním generuje vizuální feedback na televizní obrazovce nebo monitoru počítače. Někteří lidé považují videohry za formu umění. Toto tvrzení je ovšem kontroverzní.

### 2.1 Počítačová hra

Počítačové hry [3] jsou videohry, které jsou spustitelné a hratelné na osobním počítači. Oproti konzolovým hrám obvykle běží plynuleji a s vyšší grafickou úrovní, samozřejmě ale záleží na procesoru a grafické kartě počítače, na kterém je hra spuštěna.

### 2.2 Beat 'em Up žánr

Beat 'em Up [4], v překladu doslova „Zmlát je“, žánr je specifický soubojem s neúměrně velkým množstvím nepřátel. Tradičně se odehrávají na dvourozměrné scéně s postupem pouze jedním směrem. Jednotlivé úrovně pak většinou mají na konci jednoho silnějšího nepřítele, tzv. Bosse.

Tyto hry jsou známy svou jednoduchostí. Hráč během hraní nemusí moc přemýšlet, často mu stačí mačkat jedno tlačítko a mít štěstí. Samozřejmě i v tomto žánru najdeme zástupce, kteří laťku obtížnosti zvyšují a poskytují tak hráčům určitou výzvu.

Dalším charakteristickým znakem je možnost zahrát si ve dvou. S tím je spojen výběr jedné z mnoha různých hratelných postav, každá má typicky různé druhy útoků, své slabiny a přednosti. Samozřejmě vždy bude jedna z postav nejsilnější, ale dohrát hru by mělo být možné s jakoukoliv z nich.

### 2.3 Zástupci žánru

Žánr Beat 'em Up byl a stále je populární. V dnešní době spíše u malých vývojářských studií. Následuje výčet několika významných her zastupujících tento žánr.



### 2.3.1 Kung-Fu Master

Kung-Fu Master [5] je nejstarší hra toho žánru. Vydalo ji Nintendo roku 1984. Původně vyšla v Japonsku pod názvem Spartan X. Příběh je založený na záchraně přítelkyně hlavního hrdiny Thomase před tajemným Mr.X. Už první Beat 'em Up hru si bylo možné zahrát ve dvou. (nevím jestli přidávat obrázky, bylo by potřeba uvést i zdroj)



Obrázek 1: Ukázka ze hry Kung-Fu Master. Zdroj: [6]

### 2.3.2 Streets of Rage

Streets of Rage [7], v Japonsku známé jako Bare Knuckle, je trilogie od vydavatele SEGA, původně na konzoli Sega Mega Drive. Všechny tři díly vyšly v rozmezí let 1991 a 1994. Příběh se zaměřuje na tři mladé bývalé policisty, kteří se rozhodli bojovat proti korupci na vlastní pěst. Tradice hry pro více hráčů pokračuje i v této sérii. Hry byly přijaty s nadšením a byly několikrát znovu vydány buď jako součást kolekce klasických her, nebo jako samostatné tituly.



Obrázek 2: Výběr Postavy v Streets of Rage. Zdroj: [8]

### 2.3.3 Castle Crashers

Castle Crashers[9] je ze zmíněných nejnovější. Studio Behemoth ji poprvé vydalo na konzoli Xbox 360 v roce 2008, během let 2010 až 2015 hra vyšla i na dalších platformách, jako jsou PS3, Xbox One a Microsoft Windows. Oproti ostatním zmíněným hrám je zasazena do středověku a na každém kroku hráče provází humorné scény. Castle Crashers obsahují i hru po síti až pro čtyři hráče.



Obrázek 3: Castle Crashers multiplayer. Zdroj: [10]

## 3 Použité Technologie

### 3.1 Unity Framework

Unity [11] je herní engine, který dovoluje vyvíjet hry na téměř všechny známé platformy. Vydán byl roku 2005 jako exkluzivní engine pro OS-X. Poskytuje možnost vytvářet jak 2D, tak 3D hry. V současné době skriptovací API podporuje jazyk C#, dříve však uživatelé mohli programovat i v Boo nebo JavaScriptu. JavaScript byl podporován až do roku 2017.

Engine má několik verzí. Verze Personal, kterou používám pro svou hru, je zcela zdarma. Pokud ale přesáhnete jako uživatel Unity určitý příjem, musíte zaplatit za verzi Pro. Pro verze poskytuje více funkcionality a také 20% slevu na veškeré assety z Unity Asset Store. Dále existují verze Plus a Enterprise.

Vzhledem k rozsáhlé funkcionalitě, kterou Unity poskytuje už i v Personal verzi, chvíli trvá, než se v aplikaci zorientujete. Velkou roli hraje takzvaná Drag and Drop funkcionalita, která umožňuje samotné psaní kódu značně omezit. Nevýhodou ovšem je, že nemáte kontrolu nad tím co se děje. Často dochází k chybám, které nemají s kódem vůbec nic společného, ale zaseknete se nad nimi na několik hodin, protože hledáte chybu ve svém kódu. Přitom často stačí jen někde zaškrtnout správné políčko.

Unity jsem si vybrala na základě podpory práce ve 2D prostředí.

### 3.2 C#

C# [12] je vysokoúrovňový, silně typovaný programovací jazyk vyvinutý firmou Microsoft. Nepotřebuje dopřednou deklaraci metod a podporuje objektové programování. Nepodporuje vícenásobnou dědičnost, ale tuto funkcionalitu zastupují Rozhraní. Programátor se také nemusí starat o správu paměti, jelikož C# obsahuje Garbage Collector. Další výhodou je přenositelnost zdrojového kódu.

C# hrál velkou roli i ve výběru engine, který jsem nakonec použila, protože se mi v něm nejlépe pracuje a téměř všechny ostatní projekty programuji v C#, pokud to zadání projektu dovoluje.

### 3.3 Piskel

Piskel je bezplatná webová aplikace pro editování a vytváření spritů (jednotlivých obrázků, ze kterých se skládá animace). Poskytuje základní nástroje pro práci s pixelovou grafikou a také náhled na animaci, kterou právě vytváříte. Hotové sprity ukládá ve formě png.

Piskel jsem zvolila právě pro jeho specializaci na pixelové obrázky a animaci.



Obrázek 4: Uživatelské rozhraní aplikace Piskel

## 4 Programátorská dokumentace

### 4.1 Unity terminologie

#### 4.1.1 Asset

Jakákoliv média nebo data, která jsou použitelná ve hře nebo jiném druhu projektu. Asset [13] může být vytvořen přímo v Unity, příkladem může být *Animator Controller*, *Audio Mixer*. Druhá možnost je vytvoření z vnějšího souboru jednoduchým přetáhnutím do příslušné složky. Takto v Unity můžeme použít vlastní obrázky nebo 3D modely.

#### 4.1.2 Scéna

Scéna [14] obsahuje prostředí a menu hry. Jednotlivé scény jsou v podstatě úrovně. Přidáváním dekorací, překážek a prostředí do scény vývojář tvoří hru. V jeden okamžik může být zobrazena pouze jedna scéna.

#### 4.1.3 Game Object

Game Object [15] je fundamentální objekt v Unity reprezentující postavy, objekty a části scény. Samostatně nemají širokou funkcionalitu, ale fungují jako kontejnery pro komponenty, které funkcionalitu zajišťují.

#### 4.1.4 Komponenta

Komponenta [13] je část Game Objectu, která zajišťuje funkcionalitu. Jeden Game Object může obsahovat jakékoliv množství komponent. Unity obsahuje velké množství již předdefinovaných komponent, ale za použití skriptů je možné definovat vlastní.

#### 4.1.5 Prefab

Prefab [16] je typ assetu, který uloží celý Game Object i se všemi jeho komponenty. Prefab potom slouží jako šablona, ze které je možné vytvořit jakékoliv množství původního Game Objectu.

#### 4.1.6 Skript

Skript [17] zajišťuje funkcionalitu hry od reakce na vstup, přes řízení hry a tvorbu grafických efektů až po implementaci umělé inteligence nepřátel.

## 4.2 Skripty aplikace

Skripty zajišťující funkcionalitu aplikace jsou rozděleny do několika kategorií pro lepší orientaci v kódu. Je potřeba mít na paměti, že skripty v `Unity` neodpovídají třídám ve standardním projektu. V některých případech jedné třídě odpovídá až několik skriptů.

Všechny skripty v mé aplikaci dědí ze třídy `MonoBehaviour`. Tato třída poskytuje metodu `Update`, která je volána každý `frame`.

### 4.2.1 Player skripty

Tyto skripty určují funkcionalitu hratelných postav, která je záměrně rozdělena do malých, samostatných skriptů. Tento způsob omezuje redundanci kódu a zároveň umožňuje jednoduše zajistit odlišnou hratelnost pro všechny čtyři postavy.

#### 4.2.1.1 PlayerMovement.cs

Na základě vstupu od uživatele je určen směr pohybu a nastavena nová pozice hráče na scéně. Skript zároveň zajistí spuštění animace pohybu. Obsahuje proměnnou `speed`, která určuje rychlost pohybu hráče.

#### 4.2.1.2 PlayerHealth.cs

Tento skript nereaguje na vstup od uživatele. Zajišťuje správu životů a znemožnění ovládnutí postavy v případě smrti. Obsahuje mimo jiné metodu `TakeDamage`, která odečte od momentálního zdraví příslušnou hodnotu. Dále obsahuje proměnné s informacemi o počtu životů, maximální hodnotě zdraví a momentální hodnotě zdraví.

#### 4.2.1.3 PlayerAttack.cs

Skript `PlayerAttack.cs` na základě vstupu od uživatele spustí animaci útoku. Díky `Animation Events`, které `Unity` poskytuje, je možné zavolat metodu během určité části animace. Toho tento skript využívá a obsahuje metody pro provedení jak útoku na blízko, kterým disponují dvě ze čtyř hratelných postav, tak provedení útoku na dálku.

I když všechny postavy využívají stejný skript, jejich útok je odlišný. Při útoku na dálku je vytvořen nový `GameObject` pod názvem `Projectile` letící směrem, kam se postava dívá. V případě kolize projektilu je jemu příslušný `GameObject` zničen a projektil zmizí. Při útoku na blízko se pouze aktivuje na dobu animace zbraň postavy. V obou případech při kolizi zbraně nebo projektilu s nepřítelem je příslušnému nepříтели zaslána zpráva `TakeDamage`. O tuto funkcionalitu se stará metoda `OnTriggerEnter`.

Skript obsahuje příslušné proměnné s informacemi o velikosti poškození a případně rychlosti projektilu.

#### 4.2.1.4 **PlayerInput.cs**

Skript `PlayerInput.cs` pouze uchovává informaci o ovládání postavy. Obsahuje čtyři řetězce s názvy tlačítek, které slouží pro ovládání pohybu a útoku postavy.

#### 4.2.2 **Player Ability skripty**

Každá ze čtyř postav má krom útoku také speciální schopnost. Tyto schopnosti jsou natolik odlišné, že pro každou existuje samostatný skript. Schopnost se po použití na určitou časovou dobu deaktivuje.

##### 4.2.2.1 **ShieldBlock.cs**

Skript `ShieldBlock.cs` na krátký okamžik zabrání poškození postavy. Tohoto efektu je dosaženo dočasným přepsáním tagu, který určuje zda nepřátelská zbraň zašle zprávu `TakeDamage`.

##### 4.2.2.2 **PiercingArrow.cs**

Skript `PiercingArrow.cs` vytvoří projektil, který při první kolizi nezmizí, ale pokračuje ve své trase. Při každé kolizi s nepřítelem je zaslána zpráva `TakeDamage`.

##### 4.2.2.3 **ShadowWalk.cs**

Skript `ShadowWalk.cs` na několik sekund zabrání poškození postavy. Jako ve skriptu `ShieldBlock.cs` je po dobu trvání přepsaný tag objektu, čímž je zabráněno přijmutí poškození.

##### 4.2.2.4 **TripleFireball.cs**

Skript `TripleFireball.cs` vytvoří tři projektily, které letí v různých úrovních. Z každé kolize s nepřítelem je uděleno standardní poškození. Projektily po kolizi zmizí.

#### 4.2.3 **Enemy skripty**

Oproti skriptům řídícím hratelné postavy jsou skripty pro nepřátele delší a méně rozčleněné. Zároveň se jedná o nejtěžší skripty na implementaci, jelikož zahrnují jednoduchou jednoúčelovou umělou inteligenci.

##### 4.2.3.1 **EnemyHealth.cs**

Skript `EnemyHealth.cs` je společný všem typům nepřátel a funguje velmi podobně jako `PlayerHealth.cs`. Obsahuje metodu `TakeDamage`, která je

volána při kolizi se zbraní nebo projektilem. Tento skript také zabráňuje pohybu nepřítele po jeho smrti, kdy je přehrána animace smrti a následně je `GameObject` příslušný nepříteli zničen.

#### 4.2.3.2 `EnemyBrain.cs`

Skript `EnemyBrain.cs` řídí akce nepřítele na základě vzdálenosti od jeho cíle. Cíl je jeden z aktivních hráčů. Pokud nepřítel cíl nemá, pohybuje se v náhodných směrech po mapě, než cíl získá. Tuto funkcionalitu zajišťuje metoda `RandomMove`.

Cílem se hráč stane v případě, kdy ho nepřítel uvidí. Toho je docíleno aktivní oblastí v okolí nepřítele, která při kolizi s hráčem tohoto hráče určí jako cíl příslušného nepřítele. Po této akci je komponenta s touto funkcí deaktivována.

V okamžiku, kdy má nepřítel svůj cíl, rozhoduje se, jakou akci provede. Pokud je jeho cíl daleko, pomocí metody `FollowTarget` se k němu přiblíží na dosah útoku. V okamžiku, kdy je na dosah a může útočit, zaútočí na hráče a nastaví se odpočet do dalšího útoku podle proměnné `attackCD`. Jestliže útočit nemůže, protože odpočet ještě neskončil, pohybuje se v náhodných směrech kolem svého cíle.

Rozlišné chování různých druhů nepřátel je zajištěno nastavením jiných hodnot příslušných proměnných. Nepřátelé pak mohou útočit z různé vzdálenosti a v různých intervalech.

#### 4.2.4 Menu skripty

Menu skripty mají na starost pohyb uživatele v menu a správné zobrazování všech menu prvků

##### 4.2.4.1 `MainMenu.cs`

Skript `MainMenu.cs` sdružuje všechny funkce potřebné pro navigaci v hlavním menu hry. Každá z funkcí je navázána na některé z tlačítek v menu. Velká část funkcionality menu ale v tomto skriptu není kvůli výše zmíněné `Drag and Drop` podstatě `Unity`. Ta umožňuje přepínat mezi panely a deaktivovat tlačítka v menu bez potřeby psát příslušné funkce.

##### 4.2.4.2 `MouseDisable.cs`

Skript `MouseDisable.cs` blokuje kurzor a pohyb myši. Menu lze tedy ovládat pouze přes klávesnici nebo ovladač. Dále také nastavuje aktivní tlačítka na každém panelu, aby nedošlo k situaci, kdy uživatel nemůže menu ovládat.



#### 4.2.4.3 PauseMenu.cs

Skript `PauseMenu.cs` podobně jako `MainMenu.cs` sdružuje metody pro navigaci v menu, které se zobrazí během hry po zmáčknutí klávesy `Escape` nebo `Start` na ovladači. Toto menu pozastaví hru. Znemožní pohyb jak hráčům, tak nepřátelům. Hráči mají možnost hru znovu spustit nebo se vrátit do hlavního menu.

#### 4.2.5 Skripty pro řízení hry

Tyto skripty zajišťují plynulý postup hrou. Zahrnují funkcionalitu jako uchování informací o aktivních hráčích, seznam aktivních nepřátel ve hře nebo načítání nových scén.

##### 4.2.5.1 GameManager.cs

Skript `GameManager.cs` uchovává informace o spuštěné hře a přepíná mezi aktivními scénami. Nejdůležitější informace je uložena v proměnné `coop`. Ta určuje, jestli uživatel hraje sám nebo ve dvou. Podle toho se hra chová odlišně.

##### 4.2.5.2 PlayerManager.cs

Skript `PlayerManager.cs` uchovává informace o aktivních hráčích a zajišťuje jejich objevení na nové scéně. Pokud všem aktivním hráčům klesne hodnota zdraví na nulu, hra skončí a uživatelé budou vráceni do hlavního menu. Jestliže jeden hráč zemře a druhý dokončí úroveň, v další úrovni bude mrtvý hráč vrácen do hry.

##### 4.2.5.3 EnemyManager.cs

Skript `EnemyManager.cs` zajišťuje vytvoření nepřátel na aktivní scéně. Kombinace nepřátel pro hru jednoho hráče se liší od kombinace pro lokální multiplayer. Uchovává seznam aktivních nepřátel a jestliže tento seznam není prázdný, blokuje kameru.

#### 4.2.6 Camera skripty

Skripty ovládající pohyb kamery. Tyto skripty určují, co uživatelé uvidí během hraní.

##### 4.2.6.1 CameraMovement.cs

Skript `CameraMovement.cs` sleduje jednoho z aktivních hráčů v závislosti na jejich poloze na ose `x` v herním světě. Kamera se pohybuje pouze jedním směrem a proto není možné vracet se zpět probíhající úrovní. Pokud jsou na scéně aktivní nepřátelé, kamera přestane sledovat svůj cíl.

## 4.2.7 Utility skripty

Tyto různorodé skripty zajišťují další funkcionalitu potřebnou pro bezproblémový postup hrou. Nemusí nutně patřit specifickému `Game Objectu`.

### 4.2.7.1 `OnLoadScene.cs`

Skript `OnLoadScene.cs` uchovává informaci o místech, na kterých má skript `PlayerManager.cs` vytvořit hráče po načtení nové herní scény. Tento skript je komponenta hlavní kamery, protože je zajištěno, že se vyskytuje na každé nové načtené scéně, a tím pádem vždy dojde k vytvoření hráčů.

### 4.2.7.2 `ChildTrigger.cs`

Skript `ChildTrigger.cs` zajišťuje, že se provede správná akce při kolizi různých `Triggerů` náležících jednomu `Game Objectu`. Metoda `OnTriggerEnter` zašle rodiči objektu, ke kterému je tento skript připojen, zprávu. Ta je dále zpracována na základě tagů objektu, který kolidoval s `triggerem` a samotného `triggeru` jenž zprávu zaslal.

### 4.2.7.3 `Projectile.cs`

Skript `Projectile.cs` reaguje na kolizi projektilu, ať už je vytvořen hráčem, nebo nepřítelem. Tato funkcionalita je zajištěna pomocí porovnávání tagů kolidujících objektů.

### 4.2.7.4 `HealthBar.cs`

Skript `HealthBar.cs` mění `sprity` objektu znázorňujícího zbývající zdraví hráče na základě hodnoty zbývajícího zdraví. Zdraví není znázorňováno přesně, ale pouze orientačně.

### 4.2.7.5 `EnemySpawnTrigger.cs`

Skript `EnemySpawnTrigger.cs` uchovává informace o tom, jaký typ nepřítele a v jakém množství se má na scéně vyskytovat. Po kolizi hráče s objektem, na kterém je tento skript, zašle zprávu objektu se skriptem `EnemyManager.cs`. Ten už se postará o vytvoření potřebného počtu nepřátel daného typu.

### 4.2.7.6 `EndStageTrigger.cs`

Skript `EndStageTrigger.cs` zobrazí hráči na konci úrovně zprávu o dokončení úrovně. Po uplynutí určité časové doby je načtena požadovaná scéna určená proměnnou `index`.

#### **4.2.7.7 HideStagePanel.cs**

Skript `HideStagePanel.cs` deaktivuje panel s informací o právě načtené úrovni. Když je panel aktivní, hráči nemůžou ovládat své postavy. Tento skript obnovuje kontrolu hráčů nad jejich postavami.

#### **4.2.7.8 ShowEndPanel.cs**

Skript `ShowEndPanel.cs` podobně jako `HideStagePanel.cs` manipuluje s viditelností panelu zobrazujícího informace o průběhu hry. Tento skript je volán, když všichni aktivní hráči zemřou. Je zobrazena zpráva o neúspěšném konci hry a následně po uplynutí určité časové doby je načtena scéna s hlavním menu.

## Závěr

Výsledkem bakalářské práce je počítačová hra pro jednoho až dva hráče vytvořená v Unity Frameworku s použitím skriptovacího jazyku C#. Hra obsahuje tři úrovně s odlišným prostředím a čtyři postavy s rozdílnou hratelností. Bohužel se mi nepovedlo zadání práce splnit kompletně. Proto v aktuální verzi chybí hra po síti až pro čtyři hráče.

Hra po síti pro mě byla od začátku náročnou výzvou. Vzhledem k mým malým zkušenostem v této oblasti a neúplné dokumentaci k HLAPI které Unity poskytuje, jsem nebyla schopna tuto část implementovat. Do budoucna bych ovšem chtěla aplikaci o tuto funkcionalitu rozšířit. Další plány zahrnují přidání nových lokací a několika unikátních postav, stejně jako nových a chytřejších nepřátel.

## Conclusions

The outcome of the bachelor thesis is a computer game for one to two players made in Unity Framework developed in C#. The game includes three stages with different environment and four characters with unique playstyle. Unfortunately, I wasn't able to fulfill the aim of the thesis completely. Therefore this version doesn't include online multiplayer for up to four players.

For me, online multiplayer was challenging from the very beginning. Considering my little experience in this area and insufficient Unity's HLAPI documentation, I wasn't able to implement this feature. I would like to add this feature in the future development of this application. Other plans include adding new locations and unique characters as well as new and more intelligent enemies.

## A Obsah přiloženého CD/DVD

### **bin/**

Aplikace Dark Bones spustitelná přímo z CD/DVD. Adresář obsahuje všechny soubory potřebné ke spuštění aplikace.

### **doc/**

Text práce ve formátu PDF a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu.

### **src/**

Kompletní zdrojové kódy k aplikaci.

### **readme.txt**

Instrukce pro spuštění hry.

## Literatura

- [1] FROMSOFTWARE. © 1994-2018 FromSoftware, Inc. All right reserved. [online]. [cit. 2018-08-09]. Dostupné z: <https://www.fromsoftware.jp/ww/>
- [2] WIKIPEDIA. Video Game [online]. [cit. 2018-08-09]. Dostupné z: [https://en.wikipedia.org/wiki/Video\\_game](https://en.wikipedia.org/wiki/Video_game)
- [3] WIKIPEDIA. PC Game [online]. [cit. 2018-08-09]. Dostupné z: [https://en.wikipedia.org/wiki/PC\\_game](https://en.wikipedia.org/wiki/PC_game)
- [4] WIKIPEDIA. Beat 'em Up [online]. [cit. 2018-08-09]. Dostupné z: [https://en.wikipedia.org/wiki/Beat\\_%27em\\_up](https://en.wikipedia.org/wiki/Beat_%27em_up)
- [5] WIKIPEDIA. Kung-Fu Master [online]. [cit. 2018-08-09]. Dostupné z: [https://en.wikipedia.org/wiki/Kung-Fu\\_Master\\_\(video\\_game\)](https://en.wikipedia.org/wiki/Kung-Fu_Master_(video_game))
- [6] WIKIPEDIA. File:Kung fu master mame.png [online]. [cit. 2018-08-09]. Dostupné z: [https://en.wikipedia.org/wiki/File:Kung\\_fu\\_master\\_mame.png](https://en.wikipedia.org/wiki/File:Kung_fu_master_mame.png)
- [7] WIKIPEDIA. Streets of Rage [online]. [cit. 2018-08-09]. Dostupné z: [https://en.wikipedia.org/wiki/Streets\\_of\\_Rage](https://en.wikipedia.org/wiki/Streets_of_Rage)
- [8] WIKIPEDIA. File:Streets of Rage Genesis Character select.png [online]. [cit. 2018-08-09]. Dostupné z: [https://en.wikipedia.org/wiki/File:Streets\\_of\\_Rage\\_Genesis\\_Character\\_select.png](https://en.wikipedia.org/wiki/File:Streets_of_Rage_Genesis_Character_select.png)
- [9] WIKIPEDIA. Castle Crashers [online]. [cit. 2018-08-09]. Dostupné z: [https://en.wikipedia.org/wiki/Castle\\_Crashers](https://en.wikipedia.org/wiki/Castle_Crashers)
- [10] BIFUTEKI. All things awesome [online]. [cit. 2018-08-09]. Dostupné z: <http://www.bifuteki.com/wp-content/uploads/2010/09/texture051.jpg>
- [11] WIKIPEDIA. Unity (game engine) [online]. [cit. 2018-08-09]. Dostupné z: [https://en.wikipedia.org/wiki/Unity\\_\(game\\_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))
- [12] WIKIPEDIA. C Sharp (programming language) [online]. [cit. 2018-08-09]. Dostupné z: [https://en.wikipedia.org/wiki/C\\_Sharp\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))
- [13] UNITY MANUAL. Glossary [online]. [cit. 2018-08-09]. Dostupné z: <https://docs.unity3d.com/Manual/Glossary.html>
- [14] UNITY MANUAL. Scenes [online]. [cit. 2018-08-09]. Dostupné z: <https://docs.unity3d.com/Manual/CreatingScenes.html>
- [15] UNITY MANUAL. GameObject [online]. [cit. 2018-08-09]. Dostupné z: <https://docs.unity3d.com/Manual/class-GameObject.html>

[16] UNITY MANUAL. Prefabs [online]. [cit. 2018-08-09]. Dostupné z:  
<https://docs.unity3d.com/Manual/Prefabs.html>

[17] UNITY MANUAL. Scripting [online]. [cit. 2018-08-09]. Dostupné z:  
<https://docs.unity3d.com/Manual/ScriptingSection.html>