STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
MLADÁ BOLESLAV

**DLOUHODOBÁ PRAKTICKÁ  
MATURITNÍ PRÁCE**

Michal Adámek

Mladá Boleslav 2025

STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
MLADÁ BOLESLAV

War Of civilizations (hra)

**DLOUHODOBÁ PRAKTICKÁ  
MATURITNÍ PRÁCE**

**Autor: Michal Adámek  
Studijní obor: 18-20-M/01 Informační technologie  
Vedoucí práce: Radek Pelikán**

Mladá Boleslav 2025

# Obsah

[Obsah 3](#_Toc193899248)

[1 Úvod 7](#_Toc193899249)

[2 Obsah práce 8](#_Toc193899250)

[2.1 Použitá technologie 8](#_Toc193899251)

[2.1.1 Unity 8](#_Toc193899252)

[2.1.2 Microsoft Visual Studio 2022 8](#_Toc193899253)

[2.1.3 Aseprite 9](#_Toc193899254)

[2.1.4 Beepbox 9](#_Toc193899255)

[2.2 Kamera 9](#_Toc193899256)

[2.2.1 Pohyb 10](#_Toc193899257)

[2.3 Jednotky (Units) 10](#_Toc193899258)

[2.3.1 Hledání nepřátel 13](#_Toc193899259)

[2.3.2 Boj 13](#_Toc193899260)

[2.3.3 Animace 14](#_Toc193899261)

[2.4 Věže 15](#_Toc193899262)

[2.4.1 Hledání nepřátel 15](#_Toc193899263)

[2.4.2 Boj 16](#_Toc193899264)

[2.4.3 Animace 16](#_Toc193899265)

[2.5 Pevnosti (základna hráče a nepřítele) 17](#_Toc193899266)

[2.5.1 Vytváření jednotek 17](#_Toc193899267)

[2.5.2 Obrana 18](#_Toc193899268)

[2.5.3 Evoluce 19](#_Toc193899269)

[2.6 UI 20](#_Toc193899270)

[2.6.1 Cyklus času (dne a noci) 20](#_Toc193899271)

[2.6.2 Katastrofa 21](#_Toc193899272)

[2.6.3 Mini mapa 21](#_Toc193899273)

[2.6.4 Konec hry 22](#_Toc193899274)

[2.6.5 Výroba jednotek 23](#_Toc193899275)

[2.6.6 Pořizování věže 24](#_Toc193899276)

[2.6.7 Ukazatel životů základen 25](#_Toc193899277)

[2.6.8 Ukazatel hodnot (Peníze, Zkušenosti) 25](#_Toc193899278)

[2.7 Menu 26](#_Toc193899279)

[2.7.1 Hrát (Play) 26](#_Toc193899280)

[2.7.2 Nastavení (Settings) 27](#_Toc193899281)

[2.7.3 Odejít (Exit) 28](#_Toc193899282)

[3 Závěr 29](#_Toc193899283)

[4 Přílohy 30](#_Toc193899284)

[4.1 Seznam obrázků 30](#_Toc193899285)

[4.2 Zdroje 31](#_Toc193899286)

**Prohlášení**

Prohlašuji, že jsem svou ročníkovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

Nemám závažný důvod proti zpřístupňování této ročníkové práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Mladé Boleslavi dne podpis:

# Úvod

Smyslem této práce bylo vytvořit hru, která bude podobná hře Age of War[[1]](#footnote-1) se stejným žánrem hry. Hra byla vytvořena ve vývojovém prostředí Unity Engine. Tento software jsem si vybral, protože je známý svou tvorbou mnoha různých her jak na weby, tak i samostatně na počítačích a také pro jiné operační systémy. Před začátkem maturitní práce jsem si už zkoušel vytvořit hry různého žánru. Při tvoření těch her jsem se dozvěděl hodně věcí a při tvoření maturitní práce jsem tyto zkušenosti využil, a ještě se naučil nové a lepší. Hra obsahuje většinu funkcí a podobností z původní inspirované hry, ale v průběhu jsem se snažil z větší části odklonit od té inspirace a vymýšlet vlastní funkce a vzhledy nebo témata do mé hry. Kromě toho jsem také hodně času strávil testováním hry z důvodu toho, aby bylo možné hru dohrát a aby si ji uživatel užil co nejvíce. Název hry vznikl z hlavní činnosti ve hře, kterou je válka mezi dvěma civilizacemi. Následně tato dvě slova jsem přeložil do angličtiny a vhodně jsem je spojil a vzniklo „War of Civilizations“. K vytvoření hry jsem použil obrázky, které jsem si nakreslil, a ještě vytvořil animace. Ve hře se také nachází vlastnoručně vytvořená hudba, která vznikla ze syntetických neboli uměle vytvořených zvuků.

# Obsah práce

## Použitá technologie

### Unity

Unity je herní engine a vývojové prostředí vytvořené společností Unity Technologies. Vývojové prostředí dokáže vytvořit různé aplikace například hry, které dokáže tvořit ve 2D i 3D prostředí a od roku 2007 podporuje vytváření multiplayerových her s použitím protokolu UDP. Je využíváno pro vývoj aplikací na počítače, konzole, mobily a také weby. Vytvořené aplikace lze vytvořit pro daný operační systém. V dnešní době dokáže vytvořit aplikaci pro Android, iOS, PS4, PS5, WebGL, Universal Windows Platform, tvOS. Vývojové prostředí je složeno z různých klíčových komponentů: Vykreslovací engine, Fyzikální engine, Skriptování, Zvukový engine.[[2]](#footnote-2)

### Microsoft Visual Studio 2022

Microsoft Visual Studio 2022 je vývojové prostředí (IDE), které je vyvíjeno společností Microsoft, pro vytváření konzolových aplikací a také aplikací s grafickým rozhraním. Visual Studio 2022 podporuje různé programovací jazyky, jako jsou C/C++, C#, F#, Python, Ruby, Visual Basic, JavaScript s TypeScriptem a mnoho dalších. Samotné vývojové prostředí obsahuje integrované ladění kódu a také zahrnuje nástroje pro tvorbu aplikací s grafickým rozhraním (GUI).[[3]](#footnote-3)

### Aseprite

Aseprite je bitmapový grafický editor od společnosti Igara Studio, napsaný v programovacím jazyce C++. Tento software je především zaměřen na tvorbu pixelových obrázků a následnou animaci. Disponuje širokou škálou nástrojů pro úpravu samotného pixelového obrázku a také obsahuje nástroje pro export a import obrázků. V programu je možné si organizovat své kreslicí pole za pomoci vrstev pro každý obrázek (sprite).[[4]](#footnote-4)

### Beepbox

Beepbox je webová aplikace, která je zaměřena na vytváření nebo pouze navrhování hudby na základě syntetizovaných zvuků. Tato aplikace je veřejně přístupná a nepřivlastňuje si žádné výtvory, které vzniknou za použití této aplikace. Umožňuje exportovat hudbu ve formátu .mp3, .wav, .mid. Také podporuje formát .json, díky kterému si může načíst hudbu do webového editoru a pokračovat v úpravách hudby. Samotný editor dále podporuje široké spektrum zvuků a umožňuje úpravu syntetizovaných zvuků. Dále umí změnu tempa, hlasitosti, rytmu, tónu a dokonce velikost pole, kam můžeme zaznamenávat zvuk. Samotný editor ještě umožňuje na webu ukládat lokálně poslední editovanou hudbu.[[5]](#footnote-5)

## Kamera

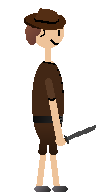
Ve hře byla prvně vytvořena logika pro kameru, a tedy v této části se bude nacházet logika, která je použita pro kameru. Kamera je použita v 2D prostředí s tím, že hráč se může pohybovat do dvou směrů (levá, pravá).

### Pohyb

Pohyb kamery se dělí na klávesový a na základě pohybu myši. V Unity se nachází přímo předdefinovaná funkce, která se postará o klávesový vstup uživatele. Výhodou je to, že nemusíme nastavovat jednotlivá tlačítka a existuje tedy zkrácený příkaz do kterého napíšeme „horizontal“, což nám určuje, pro jakou osu nastavujeme vstup pro uživatele. V tomhle případě nastavujeme osu x, takže uživatel se může pohybovat tím, že zmáčkne na klávese „A“ – doleva, „D“ – doprava nebo také může využívat šipky. Pro pohyb kamery za pomoci pozice myši se už přednastavená funkce nenachází. Na základě toho, že můžou být pouze dva směry pohybu (doleva, doprava), tak se obrazovka rozdělí na dvě pole podle velikosti aktuální šířky a výšky obrazovky. Dle pozice myši lze zjistit, jaké části obrazovky (pole) se myš dotýká a podle toho, zda bude myš na levé části obrazovky tak se kamera bude pohybovat doleva. Kamera má také nastavené hranice, za které se nemůže dostat. Hranice obrazovky jsou nastaveny dynamicky a upravují se podle aktuální velikosti obrazovky. Hlavním smyslem bylo, aby na každém zařízení byl vidět počáteční bod na stejném místě. Nastavení velikosti funguje tak, že se získá aktuální velikost obrazovky, která se vydělí dvěma a převede se z pixelů na jednotky ve vývojovém prostředí. Tím získáme poloviční velikost obrazovky a můžeme určit, kde bude hranice kamery. Díky tomu bude na každém zařízení počáteční bod stejně umístěn.

## Jednotky (Units)

V této části se bude nacházet logika použitá pro vytváření bojových jednotek ve hře a jejich funkce. Jednotky se dělí do tří tříd. První třída je „soldier“, která je charakteristická tím, že vlastní meč, který lze použít pouze na blízko. Druhou třídou je „ranger“, která se liší tím, že může ubírat životy na dálku a vlastní pouze zbraň na dálku, její velkou nevýhodou je, že má málo životů, ale na druhou stranu je hodně silná. Poslední třídou je „tank“, tato třída je charakteristická tím, že má hodně životů, ale zase neubírá hodně životů a vlastní štít, který je používán jako zbraň na blízko.



Obrázek : Soldier 3.úroveň (vlastní zdroj)

Obsah obrázku kreslené, klipart, umění

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek : Ranger 2.úroveň (vlastní zdroj)



Obrázek : Tank 1.úroveň (vlastní zdroj)

Pohyb jednotek je hodně podobný pohybu kamery, ale s tím, že se jednotka pohybuje konstantně do jednoho směru. Bojové jednotky také mají přidanou před nastavenou funkci pro fyziku neboli gravitaci a pohyb Rigidbody 2D a pro kolize s dalšíma jednotka je nastavena další před vytvořená funkce Box Collider 2D.

Obsah obrázku text, snímek obrazovky, software, Multimediální software

Popis byl vytvořen automaticky

Obrázek : Funkce Rigidbody 2D (vlastní zdroj)

Obsah obrázku text, snímek obrazovky, software, Multimediální software

Popis byl vytvořen automaticky

Obrázek : Funkce Box Collider 2D (vlastní zdroj)

Jednotky také mají funkci toho, že jakmile před sebou uvidí nepřítele či spojence, tak se zastaví. To funguje na základě příkazu, který pomyslně vytvoří okolo dané jednotky kruh a jakmile do daného kruhu se dostane cílový objekt s konkrétním layerem[[6]](#footnote-6), tak se vrátí hodnota „true“ neboli, že je to pravdivé. Podle podmínek se následně určí, že pokud tedy v tom kruhu je cílový objekt, tak se jednotka zastaví. Podmínky těchto kruhů jsou aplikovány pro kolize mezi spojenci, mezi protivníky, a ještě mezi základnami. Velikost jednotlivých kruhů pro kontrolu kolize se mění na základě jejich funkce. Nejmenší kruh kontroluje, zda není před jednotkou spojenec a zda se může jednotka pohybovat bezstarostně vpřed. Poslední kruh se velikostně mění na základě typu jednotky, pokud by se jednalo o „soldiera“ nebo „tanka“ tak ti mají stejnou velikost kruhu, ale „ranger“ má největší průměr kruhu. Je to hlavně udělané proto, aby mohl bojovat na větší vzdálenost. Díky tomuto kruhu můžou jednotky detekovat nepřítele či základnu.

Obsah obrázku snímek obrazovky, umění

Popis byl vytvořen automaticky

Obrázek : Kolize jednotek (vlastní zdroj)

### Hledání nepřátel

Hledání jednotek je funkce, bez které by se tato hra nedokázala nazývat hrou. Tato část byla nejnáročnější a vznikalo tu hodně problémů. Po několika pokusech a strávených hodinách byla vytvořena funkce, která dokáže najít každého nepřítele. Funkce funguje na základě layerů a kruhu, který detekuje objekty uvnitř na základě jejich layeru. Každá jednotka má proměnnou pro ukládání layerů nepřítele. Hned jakmile se k jednotce přiblíží nepřítel s touto proměnnou, tak se splní podmínka ve funkci pro hledání protivníků, která přiřadí skript danému objektu nepřítele a začne se spouštět funkce pro boj.

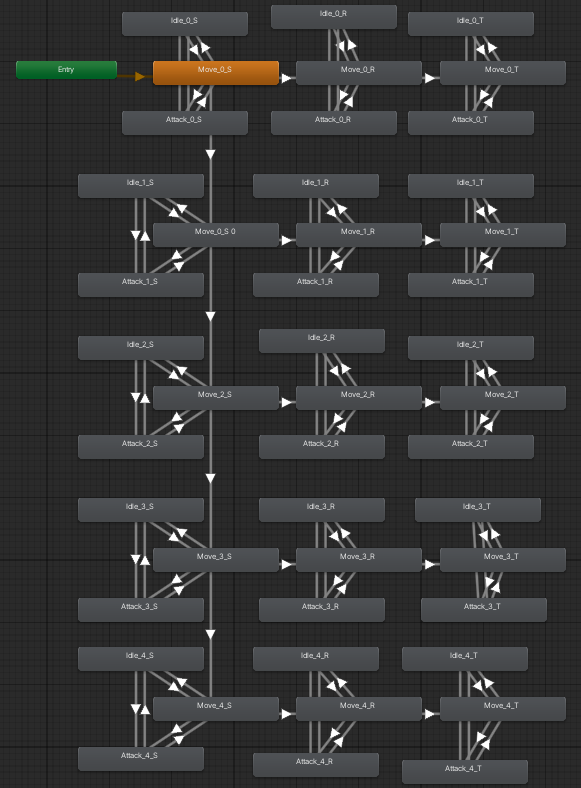
Hledání základen je o trošičku jednodušší a to tím, že můžeme každým jednotkám předem nastavit objekt se skriptem a budeme kontrolovat pouze to, zda jednotka v kruhu má objekt s layerem základny, pokud to bude pravda tak se spustí funkce pro boj.

### Boj

Funkce pro ubírání životů je další hodně důležitá funkce, bez které by se tato hra nedala nazývat hrou. Díky této funkci můžeme ubírat protivníkům a základnám životy. Při nalezení nepřítele či základny se spustí funkce pro boj, která funguje tak, že nalezenému objektu nepřítele ubere aktuální počet životů. Předtím, než ubere životy, čeká na nabití útoku, a po jeho plném nabití jednotka udeří protivníka, pokud bude čekání jakýmkoliv způsobem narušeno celá funkce se zruší a nikomu se neubere nic.

### Animace

Veškeré obrázky, které tvoří animace jednotek, byly vytvořeny v programu Aseprite, přičemž byly vytvořeny vlastnoručně. Animace fungují na principu střídání obrázků v čase, který si můžeme sami nastavit. V Unity jsou animace uspořádány do Animator Controlleru, ve kterém se animace přepínají na základě podmínek a vztahů mezi ostatními animacemi. Každý objekt ve vývojovém prostředí může mít aktivní vždy jen jednu animaci, ale může obsahovat více animací. Každý typ jednotky má tři základní animace, které se větví podle aktuální úrovně jednotky.



Obrázek : Animace jednotek (vlastní zdroj)

První animace je „movement“. Tato animace zobrazuje chůzi jednotky a spustí se tehdy, pokud není před ní žádná překážka ve formě spojence, nebo pokud nevidí před sebou nepřítele či základnu. Další animací je „idle“, která zobrazuje jednotku čekající, než bude moci jít dál nebo než bude moci zaútočit na nepřítele nebo jeho základnu. Spustí se tehdy, když před sebou uvidí překážku ve formě spojence. Poslední hlavní animací je „attack“. Díky této animaci vidíme, jak se jednotka napřahuje, a po daném čase bouchne svou zbraní a ubere nepříteli životy. Spustí se pouze tehdy, pokud uvidí jednotka nepřítele nebo nepřátelskou základnu.

## Věže

V této části bude popsána logika použitá pro fungování věží, které mohou být koupeny hráčem za finanční prostředky. Hlavním účelem věže je ubránit základnu, aby utrpěla co nejmenší poškození a při nejlepším nedostala žádné. V každé úrovni základy se nabízí jiný druh věže s jinými vlastnostmi. Každá věž se dělí na dvě části. První částí je nehybná součást zobrazující tělo věže. Druhou částí je pohyblivá část zobrazující převážně dělo. Jedinou výjimečnou věží je první úroveň, protože se skládá ze dvou nehybných částí. Věže mohou být používány nepřítelem, který ji koupí nebo vylepší ve chvíli, když jeho základna bude napadena. Po útoku na jeho základnu se zapne časovač, který nastavuje čas pro vyrobení věže. Náhodný výběr délky časovače pro vytvoření věže se pohybuje mezi jednou a pěti sekundami. Každá vyrobená věž se bude nacházet nad základnou toho kdo si jí vyrobil. Základna může vlastnit pouze jednu věž.

### Hledání nepřátel

Hledání jednotek je funkce, která nám umožňuje najít každého nepřítele s obrovskou přesností. Funguje na základě layerů a kruhu, který znázorňuje zorné pole věže. Jakmile nějaký objekt označený nepřátelským layerem vstoupí do zorného pole dané věže, se tomu objektu přiřadí skript protivníkovy jednotky. Pokud daný objekt bude mít přiřazený skript, tak se může spustit funkce pro boj. V daném kruhu se může nacházet více jednotek a tím pádem tam dochází ke komplikaci. Problém nastává v tom, že věž může útočit pouze na jednu nepřátelskou jednotku. Pokud by se tedy nacházelo více jednotek v zorném poli, tak může útočit na špatného nepřítele. Hlavní myšlenkou je, aby útočil na toho nejbližšího a díky funkci, která vypočítá vzdálenost mezi věží a objektem můžeme určit prvního nepřítele po kterém bude věž útočit. Zda věž nedetekuje žádnou přítomnost nepřátelských jednotek přesune se do výchozí pozice.

### Boj

Funkce pro boj věže funguje na základě funkce pro hledání nepřítele, bez které by nikdy nebyla bojová funkce spuštěna. Při úspěšném nalezení nepřítele se věž začne otáčet na nepřítele. Hned, jakmile věž bude natočena na protivníkovu jednotku, začne střílet projektily na aktuální pozici jednotky. Pokud se projektil dotkne cíle, tak mu na základě získaného skriptu z funkce pro hledání nepřítele ubere životy podle aktuální úrovně věže. Po výstřelu věž čeká, než bude znovu nabita a bude moci zaútočit znovu. Celou dobu, co se věž nabíjí, stále sleduje první nepřátelskou jednotku.

### Animace

Animace fungují na základě úrovně věže, která byla zakoupena. Úroveň věže se nastaví v okamžik, kdy byla zakoupena a přepíše se na aktuální úroveň základny. Podle úrovně věže se také nastaví vzhled projektilů, které věž vystřelí při útoku na nepřátelské jednotky. Každá úroveň věže a projektilu má svůj vlastní Animator Controller, který obsahuje jeden obrázek vytvořený v programu Aseprite. Veškeré obrázky pro věže a projektily byly vytvořeny vlastnoručně. Jedinou výjimkou je poslední úroveň věže, která se liší tím, že obsahuje více animací pro vyjádření svého aktuálního stavu. Prvním vyjádřením stavu věže nastává v moment, kdy věž neuvidí žádného nepřítele na obzoru a je nabitá. Pokud budou předchozí podmínky splněny, věž bude mít zapnutá zelená světla. Další vyjádření stavu věže nastává v moment, kdy je nepřítel v zorném poli věže. Tím pádem se světla nastaví na červená. Posledním světlem, které vyjadřuje stav, je žluté. Toto světlo zobrazuje, že věž nevidí žádného nepřítele a vrací se do výchozí polohy.

Obsah obrázku snímek obrazovky, diagram, text, design

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek : Animátor těla (vlastní zdroj)

Obsah obrázku snímek obrazovky, diagram, řada/pruh, text

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek : Animátor zbraně (vlastní zdroj)

## Pevnosti (základna hráče a nepřítele)

Tato část bude obsahovat logiku použitou pro základny hráče a nepřítele. Základny jsou objekty, které se nehýbou a stojí na stejném místě. Každá základna má své životy, jakmile jedna ze základen bude mít méně než nebo rovno nule, je hra u konce. Základny dokážou vytvářet jednotky a vylepšovat svou úroveň. Každé vylepšení úrovně zobrazuje vylepšení éry.

### Vytváření jednotek

Hráč a nepřítel dokážou vytvářet tři typy jednotek. Každý typ jednotek stojí jiné množství peněz a také mají jiné vlastnosti. Když se jednotka vytvoří neboli když se časovač výroby jednotek naplní, tak se vytvoří jednotka, které se přidá hodnota vrstvy do proměnné „army Type“ a vytvoří se přednastavený objekt, který bude mít tag[[7]](#footnote-7) základny, která ho vyrobila. Objektu neboli jednotce se ještě nastaví proměnná, která bude určovat jakým směrem se bude pohybovat. Pro hráče je to hodnota „left“ a pro nepřítele to je „right“ Vrstva určuje, jaký typ jednotky to je, aby se poznalo, jaké bude mít vlastnosti a také animace. Tag nám udává, zda je jednotka přátelská či nepřátelská.

Nepřítel ve výrobě má značnou výhodu. Výhoda je ta, že nepotřebuje žádné finanční prostředky pro vyrábění jednotek. Tím pádem nemůžou mu nikdy dojít náklady na výrobu. Systém výběru jednotek k výrobě je náhodný a vybírá se mezi „soldierem“ a „rangerem“, záleží, jakou má hráč nastavenou obtížnost hry. Podle obtížnosti hry se nepříteli nastavuje po kolika vyrobených jednotkách vytvoří protiútok, který může být zneužit hráčem. Zneužít ho může tím, že se nepříteli bude vyrábět jednotka „tank“ a po ní se vyrobí jednotka „ranger“. Při výrobě „tanka“ může využít času a v ten moment zaútočit na nepřítelovu základnu. Samotný protiútok nepřítele je velice účinný, protože má za cíl zničit co nejvíce jednotek hráče.

### Obrana

Základna má obranné prvky ve formě věží, které si musí hráč zakoupit za finanční prostředky. Věž mu poté bude sloužit do konce hry nebo do té doby, dokud neprodá danou věž za účelem si třeba zakoupit lepší nebo získat část peněz z koupě zpět. Věž bude sloužit v té formě, že bude útočit na nepřátelské jednotky, které se přiblíží k základně hráče. Věže nemůžou být zničeny jednotkami a nemají žádnou kapacitu munice. Dalšími obrannými prvky základny jsou jednotky, které slouží hráči na obranu a zároveň pro útok na nepřátelskou základnu. Pokud by se stalo, že se jednotka nepřátelské strany přiblíží k základně hráče a zjistí se, že před nepřátelskou jednotkou není žádná obranná jednotka hráče, tak se základně budou ubírat životy.



Obrázek : Životy základny (vlastní zdroj)

### Evoluce

Základna má pět úrovní neboli evolucí. Když jednotky hráče porazí jednotky nepřítele, získají zkušenosti, které se započítávají hráči, jakmile hráč dosáhne určitého počtu zkušeností, získá novou evoluci. Hráč vidí zkušenosti ve formátu procent. Po získání požadovaných zkušeností pro vylepšení úrovně si hráč může dobrovolně vybrat, zda bude chtít vylepšit svou úroveň, pokud se rozhodne počkat, žádné zkušenosti neztratí. Veškeré přebytečné zkušenosti se ukládají do proměnné se zkušenostmi a při vylepšení se mu pouze odečte potřebná hodnota pro vylepšení jedné úrovně. Veškeré evoluce mají stejný požadavek na vylepšení. Evoluce zvyšuje maximální životy základny, přičemž se vypočítá procentuální nárůst, kolik životů měla v předchozí úrovni a tyto procentuální hodnoty se aplikují na nové maximální životy. Životy procentuálně zůstanou, ale hodnotně budou životy větší. Vylepšení základny také upravuje vzhled základny a úroveň jednotek. Úroveň jednotek se projevuje vizuální změnou a zvýšením životů a poškozením, které jednotka způsobuje.

Základna nepřítele má stejný počet úrovní, ale nevylepšuje se na základě zkušeností nasbíraných za porážení jednotek hráče. Nepřítel si vylepšuje základnu dle zkušeností hráče. Když hráč dosáhne určitého počtu zkušeností nebo pokud by se stalo, že hráč bude o jednu úroveň napřed, tak se nepříteli automaticky spustí odpočet, který trvá patnáct sekund. Po uplynutí času se nepříteli vylepší základna na vyšší úroveň.

## UI

Tato část obsahuje logiku použitou pro UI neboli uživatelské rozhraní. Rozhraní ukazuje důležité hodnoty jako jsou například životy, peníze, zkušenosti, ale také nám umožňuje interagovat s hrou pomocí tlačítek. Díky tlačítkům může hráč vyrábět jednotky, vyvolat katastrofu, vylepšovat svou úroveň, interagovat s minimapou.

### Cyklus času (dne a noci)

Funkce starající se o simulování času ve hře tím, že se střídá den s nocí. Den je zobrazen sluncem a dalším prvkem je hlavně to, že je krajina nádherně barevná a světlá, zatímco noc je zobrazena měsícem, který přivádí ztmavení krajiny a přidává atmosféru noci bez využití prvků pro světlo ve vývojovém prostředí. Cyklus času zobrazuje pohyb objektu, kterým je slunce nebo měsíc. Tyto objekty se pohybují ve tvaru elipsy a mají nastavenou hodnotu, která zobrazuje souřadnice na ose x (pojmenování podle kartézské soustavy souřadnic), ohraničenou minimální a maximální hodnotou. Pozice y části objektu se dopočítává na základě aktuální pozice x. Pozici y je ohraničena tím, že nemůže být nikdy záporná a tím pádem se objekt bude pohybovat pouze v kladné části elipsy. Střed elipsy se nachází v bodech (0;0). K pozici x objektu se přičítá hodnota s každým dalším snímkem a díky funkci se dopočítá pozice y na základě předpisu funkce elipsy. Čas může být ovlivněn hráčem stisknutím pravého tlačítka na myši, kterými může pohybovat objektem po x ose. Při narušení pohybu objektu ze strany hráče se podle x souřadnice dopočítá y souřadnice objektu na základě předpisu funkce elipsy.

Obsah obrázku měsíc, snímek obrazovky, kreslené

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek : Den, slunce (vlastní zdroj)

Obsah obrázku měsíc, kreslené

Obsah vygenerovaný umělou inteligencí může být nesprávný.

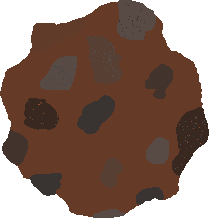
Obrázek : Noc, měsíc (vlastní zdroj)

#### Změna jasu okolí

Obrázek objektu se změní, pokud se jeho x-ová pozice dostane na svou maximální hodnotu na ose x. Pokud se tak stane, objekt se přesune na minimální pozici na ose x a bude mít změněný obrázek. V případě, že předtím bylo slunce tak po přesunutí bude nahrazeno měsícem, a to funguje i naopak. Při dni, když by bylo objektem slunce tak se při dosažení bodu, který je poblíž maximální hodnoty, začne okolí ztmavovat a v opačném případě, pokud by byl objektem měsíc tak se na stejném místě stane zase to, že se pozadí začne rozjasňovat.

### Katastrofa

Katastrofa slouží hráči jako zvýhodnění oproti nepříteli zejména v protivníkově výrobě. Zvýhodnění nastává v tom, že hráč si může díky katastrofě získávat peníze tím, že zabijí nepřátelské jednotky za pomoci katastrofy, která se dobijí po určitém čase a nestojí ho žádné finance. Katastrofa funguje na základě jednotlivých kruhů, které se náhodně vytvoří nad mapou bojového pole. Kruhy se vytvářejí jednotlivě a obsahují komponentu Rigidbody 2D, která přidává kruhu gravitaci. Pokud se kruh dotkne nepřátelské jednotky, tak se díky dalšímu komponentu jménem Box Collider 2D, najde objekt a k němu se přidá skript nepřítel. Pokud se předchozí funkce splní, tak se kruh zničí a odečte nepříteli poškození, které se každou úrovní konstantně zvyšuje. Samotné kruhy vypadají jako meteority, které využívají další komponentu Trail Render pro zvýraznění ohnivého ocasu za meteoritem. Další funkcí, která se vykonává při vytváření meteoritů při zapnuté katastrofě je, že se meteority náhodně otáčejí.



Obrázek : Meteorit (vlastní zdroj)

### Mini mapa

Tento ukazatel zmenšené mapy celého herního pole nás informuje hlavně ohledně toho, jakou část mapy hráč aktuálně vidí. Samotná mapa se aktualizuje při pohybu klávesnice nebo myší. Funguje na základě hraničních bodů kamery, které se nastavují dynamicky podle aktuální velikosti obrazovky. Minimapa je vytvořena přes posuvník, ve kterém se nastaví minimální hodnota na nulu, která zobrazuje obrazovku úplně na levé straně. Druhá strana se nastaví podle vzdálenosti levého a pravého hraničního bodu kamery. Následně vypočítaná vzdálenost mezi body se dosadí do maximální hodnoty posuvníku a pokud tedy kamera bude na pravé straně neboli na konci mapy, tak v posuvníku to bude zobrazovat maximální hodnotu. Minimapa je pro hráče interaktivní z toho pohledu, že s posuvníkem může pohybovat a tím pádem pohybuje s kamerou bez využití klávesnice nebo bodů posunu pro myš. Minimapa se nachází v pravé části obrazovky, konkrétně pod tlačítky pro výrobu jednotek nebo ukazatele informací ohledně věže.

### Konec hry

Hra skončí, pokud jedna ze základen bude poražena druhou. V případě naplnění této situace se hra ukončí a ukáže se výsledek, kdo vyhrál hru a podle toho se uzpůsobí oslava. V případě výhry hráče se bude oslavovat vítězným ohňostrojem, který je vytvořen pomocí komponenty Particle System. V opačném případě se nic nebude oslavovat a pouze se ukáže, že hráč prohrál hru. Při zjištění výsledků se veškeré jednotky na bitevním poli přestanou pohybovat a začnou skákat do vzduchu. Jednotky tímto gestem naznačují oslavu nad ukončením dlouhodobé války mezi dvěma civilizacemi a toto gesto se uskuteční v obou případech dohrání hry.

#### Ohňostroj

Funkce pro zobrazování ohňostroje při výhře hráče funguje na základě komponenty Particle System a skládá se ze čtyř hlavních částí, kterými tvoří efekt raketky s následným výbuchem. První částí je samotné tělo, které je ohraničeno místem odkud může být vystřeleno do vzduchu. Tělo je zobrazeno barevným kruhem. Druhou částí je ocas rakety při výstřelu, který zobrazuje kouř za raketou. Ten je zobrazen menšími kruhy, které mají podobnou barvu a následují první část. Třetí částí je výbuch první části neboli raketky, který se rozdělí na více kruhů, které se vzdalují od výbuchu první části rakety, která v tomto kroku s druhou částí zmizí. Poslední částí je znovu ocas, ale tentokrát je součástí jednotlivých kruhů při třetí částí. Po krátkém intervalu zmizí i třetí část a tím zmizí i čtvrtá část. Proces se opakuje a může být vypuštěno více takových raketek najednou a vytváří efekt ohňostroje. Jejich barva je náhodná v barevném intervalu.

Obsah obrázku kreslené, Dětské kresby, snímek obrazovky

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek : Ohňostroj (vlastní zdroj)

### Výroba jednotek

Pro výrobu jednotky se používají tlačítka, na které uživatel klikne pro vyrobení konkrétní jednotky, která je vyobrazena na daném tlačítku svou charakteristikou nebo názvem. Každé tlačítko obsahuje informace ohledně aktuální úrovně a také kolik je potřeba finančních prostředků k její výrobě.



Obrázek : Tlačítka pro výrobu jednotek (vlastní zdroj)

Při výrobě více jednotek je vidět, kolik se plánovaně bude vyrábět jednotek a také kolik má hráč ještě místa pro plánovanou výrobu jednotek. Pokud by uživatel překročil maximální limit plánované výroby, tak se uživateli ukáže chyba, že nemá dostatek místa pro plánovanou výrobu jednotek. Chyba také funguje pro ukazatel, zda má uživatel dostatek financí. Tím pádem, pokud by neměl a chtěl by vyrobit jednotku, tak se mu chyba zobrazí s tím, že nemá dostatek finančních prostředků.

Funkce pro výrobu jednotek funguje na základě dvou proměnných, které mají datový typ queue[[8]](#footnote-8). Díky těmto proměnným můžeme ukládat frontu v pořadí, ve které byly postupně přidané jednotky do výroby. První proměnná funguje na ukládání hodnot postupně a druhá hodnota funguje pro jednotky, které se vyrobí, ale zpozdí se jejich nasazení. To funguje na základě vyrobení jednotky a pokud bude mít hráč zastavenou výrobu, tak se jednotky začnou přidávat do druhé proměnné. Při kliknutí na otevření fronty se jednotky začnou nasazovat do boje. Obě proměnné mají dohromady limit na maximálně pět jednotek. Nasazení jednotek, které jsou ve druhé proměnné, trvá půl sekundy.



Obrázek : Fronta pro budoucí výrobu jednotek (vlastní zdroj)

Pro zobrazení aktuální výroby jednotek ve smyslu, kolik času ještě zbývá do dokončení výroby jedné jednotky nám slouží grafický ukazatel výroby. Jedná se o grafické vyplňování obdélníku na základě procent podle toho, kolik času ještě zbývá k vyrobení jednotky, která je na řadě ve výrobě. Pokud je obdélník prázdný, tak se jednotka teprve začala vyrábět. Pro opačný případ se vyrobí jednotka a obdélník se vyprázdní. Každá jednotka má jiný čas pro výrobu. Nejkratší čas pro vyrobení jednotky má „soldier“ a naopak ten nejdelší čas má „tank“, což je hlavně zapříčiněno jeho vlastnostmi.



Obrázek : Grafický ukazatel výroby jednotek (vlastní zdroj)

### Pořizování věže

Tlačítko pro pořízení věže je ve tvaru bílého kříže, který připomíná operátor pro sčítání. Tlačítko se nachází nad hráčovou základnou a funguje na základě najetí myši. Uživatel si může pořídit vlastní věž za určitou finanční částku, která se každou úrovní zvyšuje podle základní hodnoty peněz, vynásobenou aktuální úrovní jeho základny. Při najetí na bílý kříž se uživateli zobrazí informační tabulka. Konkrétně se zobrazí na místě, kde se nacházejí tlačítka pro vyrábění jednotek. Tabulka obsahuje informace ohledně nákupní ceny, poškození, poloměr kružnice zobrazujícího zorné pole věže a rychlosti střílení projektilů. Po koupení věže se místo bílého kříže zobrazí věž aktuální úrovně základny a při najetí se na místě věže zobrazí červený koš. Tabulka bude obsahovat stejné informace, ale místo kupní ceny se tam bude nacházet finanční hodnota při prodání věže, která bude o dvacet pět procent z původní částky nižší. Pokud by hráč prodal věž, tak se mu snížená hodnota peněz vrátí a místo věže se tam bude znovu ukazovat bílý kříž.

Obsah obrázku symbol, hrob, hřbitov

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek : Symbol pro koupení věže (vlastní zdroj)

Obsah obrázku kreslené, snímek obrazovky, Karmín, pixel

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek : Symbol pro prodání věže (vlastní zdroj)

### Ukazatel životů základen

Pro vidění aktuálních životů základen nám slouží grafický ukazatel životů nebo numerické ukazatele. Grafické ukazatele jsou dva. Pro hráče se ukazatel nachází na levé části obrazovky a pro nepřítele se ukazatel nachází na pravé části obrazovky. Logika grafických životů funguje na základě procent životů. Dle procent se naplní obdélník, který považujeme za ukazatel. Obdélník se zbarvuje podle toho, kolik má základna životů. V případě maximálních životů bude obdélník zelený a v opačném případě bude červený.



Obrázek : Životy základny s 80% (vlastní zdroj)



Obrázek : Životy základny s 20% (vlastní zdroj)

### Ukazatel hodnot (Peníze, Zkušenosti)

Logika pro získávání a následné zobrazení hodnot peněz a zkušeností. Hodnoty se získávají z jednotek, které zemřou. Konkrétně před tím, než jednotka zemře, přidá do proměnné s penězi a zkušenostmi určitou hodnotu a poté se jednotka odstraní. Po získání hodnot se aktualizuje textové pole, ve kterém se zobrazuje daná proměnná. Každé textové pole s konkrétní proměnnou je označeno obrázkem, který definuje název proměnné.

Získávání peněz se oproti získávání zkušeností výrazně liší z důvodu vyvážení hry. Zkušenosti hráč získává v konstantní míře a mění se pouze hodnota u typů jednotek, zatímco u peněz je to jiné. Hráč získává peníze na základě aktuální úrovně protivníka a také podle typu jednotky, kterou porazil. Peněžní hodnota závisí na úrovni protivníka. Počítá se na základě první hodnoty daného typu jednotky, která se vynásobí aktuální úrovní té jednotky. Tato hodnota se ještě vynásobí procentuálním koeficientem, který mírně snižuje finální zisk peněz.



Obrázek : Ukazatel hodnot zkušeností a peněz (vlastní zdroj)

## Menu

Tato část bude obsahovat logiku použitou pro vytvoření menu, které obsahuje tři tlačítka. Každé tlačítko obsahuje jinou funkci a logiku.

Pozadí hlavní nabídky je doplněno obrázkem, který je ve formátu .gif, ale vývojové prostředí Unity tento formát nepodporuje a tím pádem je tento obrázek ručně animovaný a skládá se z více obrázků. Obrázek obsahuje hrad, který má zobrazovat hráčovu pevnost. Tato pevnost za svými pevnými hradbami brání bednu plnou zlaťáků a drahokamů. Před hradbami z okolního světa a na hradbách se nachází hradní stráž, která brání hrad před hrozbami ve světě.

### Hrát (Play)

První tlačítko má název „Hrát“. Toto tlačítko zaručuje, že jakmile ho hráč stiskne, tak ho to přesměruje do nastavení hry, kde se nastavuje obtížnost. Podle které se bude nepřítel chovat jinak a bude se mu měnit interval, kdy bude zranitelný. Čím je obtížnost lehčí, tím bude mít hráč více možností zaútočit na nepřátelskou základnu. V opačném případě se nepřítel bude starat o to, aby měl hráč co nejméně šancí na útok. Po nastavení obtížnosti se v menu bude nacházet další tlačítko, a to pustí hráče do hry. Po stisknutí se hra načte do scény, ve které je samotná hra. Tlačítko obsahuje předem vytvořenou funkci, která se stará o přechod mezi jinými scénami ve vývojovém prostředí.

**public** **void** Play()

{

SceneManager.LoadScene("GameScene");

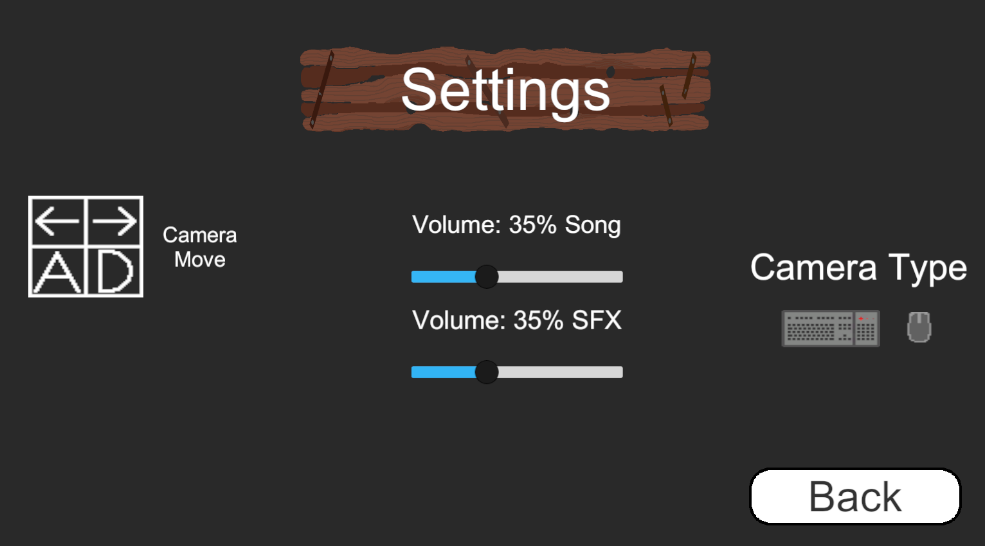
GameScript.isGameOver = **false**;

}

Zdrojový kód č. 1

### Nastavení (Settings)

Tlačítko s názvem „Nastavení“ je o trošičku jiný od toho prvního. Pokud by bylo aktivováno, tak se scéna nezmění, ale pouze se prohodí, tím je myšleno, že se vypne hlavní část nabídky a zapne se část s nastavením, kde se uživatel může podívat, jak se konkrétně ovládá kamera ve hře. Instrukce jsou znázorněny obrázkem, který obsahuje šipky „doprava“ a „doleva“ a písmena „A“ a „D“, čímž značí, jakou klávesou se ovládá. Další věcí je zvuk, který si uživatel může nastavit za pomocí posuvníku. V nastavení se konkrétně nacházejí dva, tím prvním je posuvník na hudbu, který se stará o hlasitost hudby a tím druhým je posuvník na zvukové efekty, čímž jsou například zvuky, které vydávají jednotky při útoku na nepřítele. Další a zároveň poslední věcí je nastavení pohybu kamerou. Uživatel si může vybrat, zda bude pohybovat s kamerou za pomoci klávesnice nebo myši a pokud mu vyhovují obě věci, tak si je může nechat obě zapnuté. Zapnutý typ pohybu pozná tak, že není průhledný a zase naopak pohyb, který je vypnutý pozná tak, že je lehce průhledný. Dále v pohybovém nastavení platí pravidlo, že musí mít alespoň klávesnici nebo myš zapnutou. Pokud by měl oba vstupy pro kameru vypnuté, nemohl by se pohybovat kamerou.



Obrázek : Vzhled nastavení (vlastní zdroj)

### Odejít (Exit)

Poslední tlačítko se stará o odchod ze hry. Funkce tlačítka je vytvořena za pomoci před vytvořené funkce, která bezpečně ukončí aplikaci, aniž by jí uživatel vypínal násilně.

**public** **void** QuitGame()

{

Application.Quit();

}

Zdrojový kód č. 2

# Závěr

Při vytváření hry se naskytlo hodně problémů. Některé chyby nebyly tak vážné a byly opraveny během pár chvilek, zatímco jiné chyby dokázaly zabrat až týdny na opravení. Původní myšlenky některých věcí, které měly velké chyby, byly změněny na něco jiného nebo odstraněny ze hry. S výsledkem práce jsem spokojený tak na sedmdesát procent, protože tam není vše, co jsem původně zamýšlel. Na maturitní práci jsem pracoval od úplného počátku až ke konci s tím, že jsem se snažil pracovat pravidelně, což se mi dařilo, ale stále si myslím, že jsem měl do práce vložit větší úsilí. Kdybych měl vytknout část, která se mi nejméně dařila, tak by to bylo kreslení a vytváření vlastní hudby. Tyto dvě činnosti mi zabraly spoustu času a stále si myslím, že jsem je neudělal dostatečně kvalitně. Na druhou stranu jsem se toho hodně naučil ohledně vývojového prostředí Unity a také v oblasti kreslení pixelových obrázků a animování v softwaru Aseprite. Díky této práci jsem se také naučil pár příkazů v jazyce C# a hlavně jsem si vyzkoušel vytvářet vlastní hudbu na webové stránce BeepBox. S těmito zkušenostmi, které jsem nasbíral při vytváření této práce, bych rád vytvořil další hru, ale s tím, že by hra byla větší a rozsáhlejší. Na budoucí hře bych chtěl pracovat s kamarády, protože by to bylo zábavnější a určitě by to bylo kvalitnější.

# Přílohy

## Seznam obrázků

[Obrázek 1: Soldier 3.úroveň (vlastní zdroj) 11](#_Toc193908553)

[Obrázek 2: Ranger 2.úroveň (vlastní zdroj) 11](#_Toc193908554)

[Obrázek 3: Tank 1.úroveň (vlastní zdroj) 11](#_Toc193908555)

[Obrázek 4: Funkce Rigidbody 2D (vlastní zdroj) 12](#_Toc193908556)

[Obrázek 5: Funkce Box Collider 2D (vlastní zdroj) 12](#_Toc193908557)

[Obrázek 6: Kolize jednotek (vlastní zdroj) 13](#_Toc193908558)

[Obrázek 7: Animace jednotek (vlastní zdroj) 14](#_Toc193908559)

[Obrázek 8: Animátor těla (vlastní zdroj) 17](#_Toc193908560)

[Obrázek 9: Animátor zbraně (vlastní zdroj) 17](#_Toc193908561)

[Obrázek 10: Životy základny (vlastní zdroj) 19](#_Toc193908562)

[Obrázek 11: Den, slunce (vlastní zdroj) 21](#_Toc193908563)

[Obrázek 12: Noc, měsíc (vlastní zdroj) 21](#_Toc193908564)

[Obrázek 13: Meteorit (vlastní zdroj) 22](#_Toc193908565)

[Obrázek 14: Ohňostroj (vlastní zdroj) 24](#_Toc193908566)

[Obrázek 15: Tlačítka pro výrobu jednotek (vlastní zdroj) 24](#_Toc193908567)

[Obrázek 16: Fronta pro budoucí výrobu jednotek (vlastní zdroj) 25](#_Toc193908568)

[Obrázek 17: Grafický ukazatel výroby jednotek (vlastní zdroj) 25](#_Toc193908569)

[Obrázek 18: Symbol pro koupení věže (vlastní zdroj) 26](#_Toc193908570)

[Obrázek 19: Symbol pro prodání věže (vlastní zdroj) 26](#_Toc193908571)

[Obrázek 20: Životy základny s 80% (vlastní zdroj) 27](#_Toc193908572)

[Obrázek 21: Životy základny s 20% (vlastní zdroj) 27](#_Toc193908573)

[Obrázek 22: Ukazatel hodnot zkušeností a peněz (vlastní zdroj) 28](#_Toc193908574)

[Obrázek 23: Vzhled nastavení (vlastní zdroj) 29](#_Toc193908575)

## Zdroje

OpenAI. ChatGPT 3.5. [online]. [cit. 2025-03-14]. Dostupné z <https://openai.com/blog/chatgpt>.

Age of War [online]. Age of War Wiki. [online]. [cit. 2025-03-14]. Dostupné z <https://aowg.fandom.com/wiki/Age_of_War_Wiki>.

*Unity (herní engine)*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-. Dostupné z: <https://cs.wikipedia.org/wiki/Unity_(hern%C3%AD_engine)>. [cit. 2025-03-14].

*Microsoft Visual Studio*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-. Dostupné z: <https://cs.wikipedia.org/wiki/Microsoft_Visual_Studio>. [cit. 2025-03-14].

*Aseprite*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-. Dostupné z: <https://cs.wikipedia.org/wiki/Aseprite>. [cit. 2025-03-15].

*BeepBox*. Online. 2012. Dostupné z: <https://www.beepbox.co/>. [cit. 2025-03-14].

1. Age of War [online]. Age of War Wiki. [online]. [cit. 2025-03-14]. Dostupné z <https://aowg.fandom.com/wiki/Age_of_War_Wiki> [↑](#footnote-ref-1)
2. *Unity (herní engine)*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-. Dostupné z: <https://cs.wikipedia.org/wiki/Unity_(hern%C3%AD_engine)>. [cit. 2025-03-14]. [↑](#footnote-ref-2)
3. *Microsoft Visual Studio*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-. Dostupné z: <https://cs.wikipedia.org/wiki/Microsoft_Visual_Studio>. [cit. 2025-03-14]. [↑](#footnote-ref-3)
4. *Aseprite*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-. Dostupné z: <https://cs.wikipedia.org/wiki/Aseprite>. [cit. 2025-03-15]. [↑](#footnote-ref-4)
5. *BeepBox*. Online. 2012. Dostupné z: <https://www.beepbox.co/>. [cit. 2025-03-14]. [↑](#footnote-ref-5)
6. Layer v překladu znamená vrstva. Ve vývojovém prostředí se tímto rozdělují objekty do určitých skupin nebo kategorií. Následujícím skupinám můžeme upravovat chování nebo je používat v určitých funkcích. [↑](#footnote-ref-6)
7. Tag v překladu znamená štítek. Ve vývojovém prostředí nám to slouží jako identifikátor, který lze přiřadit objektu. Tagy můžeme používat ve skriptech k rozpoznávání objektů. [↑](#footnote-ref-7)
8. Queue je datový prvek, který si ukládá hodnoty v pořadí první nakonec a první ven neboli FIFO (First in, First out) [↑](#footnote-ref-8)