STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
MLADÁ BOLESLAV

**ROČNÍKOVÁ PRÁCE**

Michal Adámek

Mladá Boleslav 2024

STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
MLADÁ BOLESLAV

**ROČNÍKOVÁ PRÁCE**

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

Mladá Boleslav 2024

# Obsah

[1 Úvod 7](#_Toc164182454)

[2 Obsah práce 8](#_Toc164182455)

[2.1 Použitá technologie 8](#_Toc164182456)

[2.1.1 Unity 8](#_Toc164182457)

[2.1.2 Microsoft Visual Studio 2022 8](#_Toc164182458)

[2.1.3 Aseprite 9](#_Toc164182459)

[2.2 Kamera 9](#_Toc164182460)

[2.2.1 Pohyb 9](#_Toc164182461)

[2.3 Jednotky (Units) 10](#_Toc164182462)

[2.3.1 Pohyb 10](#_Toc164182463)

[2.3.2 Hledání nepřátel (Boj) 12](#_Toc164182464)

[2.3.3 Animace 12](#_Toc164182465)

[2.4 Pevnosti (základna hráče a nepřítele) 13](#_Toc164182466)

[2.4.1 Vytváření jednotek 13](#_Toc164182467)

[2.4.2 Boj 14](#_Toc164182468)

[2.4.3 Evoluce 14](#_Toc164182469)

[2.5 UI 15](#_Toc164182470)

[2.5.1 Výroba jednotek 15](#_Toc164182471)

[2.5.2 Ukazatel životů základen 16](#_Toc164182472)

[2.5.3 Ukazatel hodnot (Peníze, Zkušenosti) 17](#_Toc164182473)

[2.6 Menu 17](#_Toc164182474)

[2.6.1 Hrát (Play) 17](#_Toc164182475)

[2.6.2 Nastavení (Settings) 17](#_Toc164182476)

[2.6.3 Odejít (Exit) 18](#_Toc164182477)

[3 Závěr 19](#_Toc164182478)

[4 Přílohy 20](#_Toc164182479)

[4.1 Seznam obrázků 20](#_Toc164182480)

[4.2 Zdroje 21](#_Toc164182481)

**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) s podobností žánru hry. Hra byla vytvořena ve vývojovém prostředí Unity engine. Tento software jsem si vybral z důvodu toho, že je známý svou tvorbou mnoha různých her jak na weby, tak i samostatně na počítačích a také pro jiné operační systémy. Před začátkem ročníkové práce jsem si už zkoušel vytvořit hry různého žánru. Při tvoření těch her jsem se dozvěděl hodně věcí a při tvoření ročníkové práce jsem tyto zkušenosti využil, a ještě se naučil nové a lepší. Hra měla původně být hratelná ve dvou možnostech, tím prvním je že by hráč hrál sám proti počítači a druhá, že by bojoval s člověk přes LAN nebo online, ale od plánu jsem ustoupil z důvodu dosavadní náročnosti hry. Hra také měla obsahovat věže, které by bránily základny před nepřátelskou jednotkou. Věže by se kupovaly za peníze a následně se stavěly na základny. Kromě toho jsem také hodně času strávil testováním hry z důvodu toho, aby bylo možné dohrát hru a aby si uživatel užil hru. Název hry vzniknul z odvození inspirované hry Age of War a zbytek vzniknul z důvodu toho, že v poslední době každou hru, kterou vytvořím tak obsahuje slovo Wish a díky tomu vzniknul název Wish of War. K vytvoření hry jsem použil obrázky, které jsem si nakreslil, a ještě z animoval. Jediné, co jsem nevytvořil já byla hudba, kterou je Glorious Morning od autora Waterflame[[2]](#footnote-2). Tato hudba je volně použitelná, ale s podmínkou tou, že hra nebude z peněžně na a bude se ve hře vyskytovat odkaz na autora hudby.

# 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.[[3]](#footnote-3)

### Microsoft Visual Studio 2022

Microsoft Visual Studio 2022 je integrované vývojové prostředí (IDE) vyvinuté společností Microsoft. Toto IDE poskytuje vývojářům prostředí pro vytváření, ladění a správu softwarových projektů napříč různými platformami, včetně Windows, macOS, Linux, iOS, Android a dalších. Visual Studio 2022 podporuje různé programovací jazyky, jako je C#, C++, Visual Basic, F#, JavaScript, TypeScript a mnoho dalších. Mezi jeho klíčové funkce patří integrované nástroje pro ladění kódu, správu verzí, tvorbu uživatelských rozhraní a podporu pro různé typy projektů, včetně desktopových aplikací, webových aplikací, mobilních aplikací a cloudových služeb.[[4]](#footnote-4)

### Aseprite

Aseprite je bitmapový grafický editor a animační nástroj zaměřený především na tvorbu pixel artu. Tento software je oblíbený mezi umělci, herními designéry a animátory pro svou jednoduchost, efektivitu a specifické funkce pro práci s pixelovou grafikou. Tento Software poskytuje širokou škálu nástrojů pro kreslení, editaci, animaci a export pixelové grafiky a umožňuje uživatelům pracovat s různými vrstvami, animačními sekvencemi a paletami barev.[[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 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 obrazovky. Dle pozice myši lze zjistit, jaké části obrazovky (pole) se dotýká myš 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 jsou nastaveny jako pevné objekty, které mají kolizi a jsou staticky umístěny (viz. Zdrojový kód č. 1 nebo 4.2.1).

**if** (Input.mousePosition.x < Screen.width / touchField && Input.mousePosition.y < (Screen.height \* 3) / 4 && Input.mousePosition.y > Screen.height/ touchField)

{

activeX = -1;

}

**else** **if** (Input.mousePosition.x > Screen.width - Screen.width / touchField && Input.mousePosition.y < (Screen.height \* 3) / 4 && Input.mousePosition.y > Screen.height / touchField)

{

activeX = 1;

}

**else**

{

activeX = Input.GetAxis("Horizontal");

}

Zdrojový kód č. 1

## Jednotky (Units)

V této části se bude nacházet logika použita pro vytvoření bojových jednotek ve hře a jejich funkce. Jednotky se dělí do tří tříd. První třída je „soldier“, která je charakteristická tím, že vlastní pouze zbraň na blízko. Druhou třídou je „ranger“, která se liší tím, že může ubírat životy na dálku a vlastní pouze zbraň na dálku. Poslední třídou je „tank“, tato třída je charakteristická tím, že má hodně životů, ale zase neubírá hodně životů a vlastní štít, který je použit jako zbraň na blízko.

### Pohyb

Pohyb jednotek je hodně podobný pohybu kamery, ale s tím, že se jednotka pohybuje konstantně do jednoho směru. 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 „collision 2D“.

Obsah obrázku text, snímek obrazovky, software, Multimediální software

Popis byl vytvořen automaticky

Obrázek 1: Funkce Rigidbody 2D (vlastní zdroj)

Obsah obrázku text, snímek obrazovky, software, Multimediální software

Popis byl vytvořen automaticky

Obrázek 2: 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 daného objektu kruh a jakmile do daného kruhu se dostane cílový objekt tak se vrátí hodnota „true“.

Obsah obrázku snímek obrazovky, umění

Popis byl vytvořen automaticky

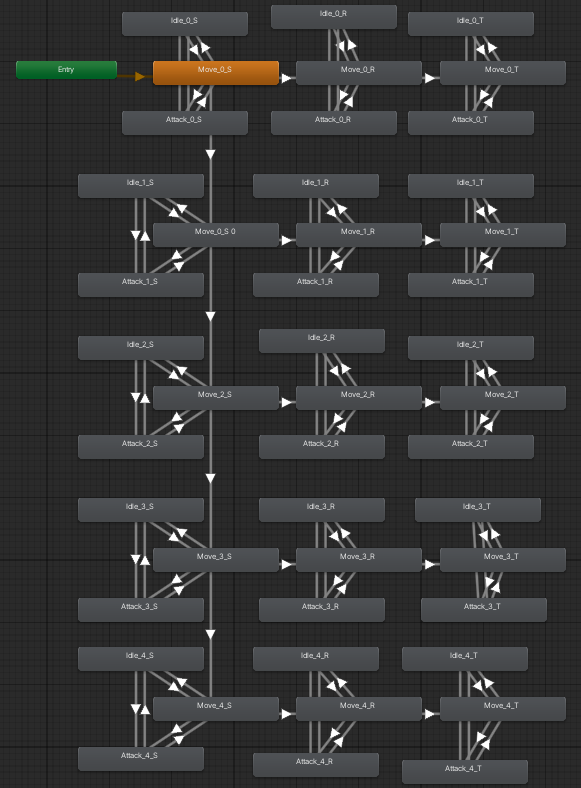
Obrázek 3: Kolize jednotek (vlastní zdroj)

### Hledání nepřátel (Boj)

Boj jednotek je funkce, bez které by se tato hra nedokázala nazývat hrou. Tato část byla nejvíce náročná a vznikalo tu hodně problematik. Po několika pokusech a strávených hodinách se vytvořil funkce, která dokáže najít každého nepřítele. Funkce funguje na základě „tagů“[[6]](#footnote-6) a „layerů“[[7]](#footnote-7). Každá jednotka, jakmile se k ní přiblíží tag nepřátelského značení, spustí funkci, která najde všechny objekty typu nepřítele a následně se všechny hodnoty uloží do „arraye“[[8]](#footnote-8). Daný „array“ se vyfiltruju, tak aby se poznalo, co je spojenec a co je nepřítel. Potom se vytvoří další „array“, který bude obsahovat pouze nepřátelské objekty a vybere si první objekt ke kterému přidá script. Když se k objektu tedy přiblíží nepřátelský objekt tak na základě uloženého scriptu se nepříteli uberou životy. Stejný smysl je proveden pro ubírání životů základnám, ale s tím, že se tam staticky nastavil objekt a script.

### Animace

Veškeré obrázky, které tvoří animace jednotek byly vytvořeny v programu Aseprite s tím, že to bylo vytvořeno vlastnoručně. Animace fungují s pomocí vrstev. Ve scriptu se nachází příkaz, který vytvoří okolo objektu pomyslný kruh, který reaguje na vrstvy. Následně když v kruhu je konkrétní vrstva (nepřátelská či spojenecká), tak výstup bude „true“, pokud je „true“ tak se objekt zastaví a bude se přehrávat animace typu „idle“, to stejné je použité proto když tam bude vrstva nepřítele, ale s tím, že se přehraje animace „attack“, jestli by před objektem nic nebylo tak se bude pohybovat kupředu a bude přehrána animace „move“.



Obrázek 4: Animace jednotek (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 svoje životy, jakmile jedna ze základen bude mít méně nebo rovno 0 tak je hra u konce. Základny dokážou vytvářet jednotky a vylepšit svou úroveň.

### 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 také „tag“. 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. Systém výběru na výrobu jednotek je náhodný. Při vybrání jednotky bude čekat, po naplnění času se cyklus opakuje. Nepřítel má vyrábění jednotek o trošičku delší z toho důvodu, aby se nestalo, že by byly vyrobeny jednotky stejného typu opakovaně za sebou, které jsou nejtěžší a nejdražší.

### Boj

Základna nemá žádné obranné prvky, jediné obranné prvky jsou jednotky. Když 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.

Životy základny se 100%

Obrázek 5: Životy základny (vlastní zdroj)

### Evoluce

Základna má pět úrovní neboli evolucí. Když jednotky hráče porazí jednotky nepřítele, tak získají zkušenosti celkově započítané pro hráče, jakmile hráč dosáhne určitého počtu zkušeností, získá novou evoluci. Hráč vidí zkušenosti ve formátu procent. Evoluce se projeví na zvýšení životů s tím, že se vypočítá v procentech, kolik životů měla v předchozí úrovni a následně se tyto procenta vloží do nových maximálních životů. Životy procentuálně zůstanou, ale hodnotně budou životy větší. Vylepšení základny, také upravuje vzhled základny a úroveň jednotek. Úroveň jednotek se projevuje změnou vzhledem a zvýšením životů a poškozením, které bude daná jednotka udělovat.

Základna nepřítele má stejný počet úrovní, ale nevylepšuje se na základě zkušeností nasbíraných za ničení 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 je 15 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 dovoluje pracovat se samotnou hrou přes čudlíky.

### Výroba jednotek

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



Obrázek 6: Čudlíky pro výrobu jednotek (vlastní zdroj)

Při výrobě více jednotek je vidět kolik se plánovaně bude vyrábět jednotek a také kolik máme 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, pokud by neměl a chtěl by vyrobit jednotku, tak se chyba ukáže s tím, že nemá dostatek finančních prostředků.

Fronta pro budoucí výrobu jednotek.

Obrázek 7: Fronta pro budoucí výrobu jednotek (vlastní zdroj)

Pro vidění aktuální výroby jednotek ve smyslu, kolik času ještě se bude čekat k vyrobení 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.

Grafický ukazatel výroby jednotek.

Obrázek 8: Grafický ukazatel výroby jednotek (vlastní zdroj)

### Ukazatel životů základen

Pro vidění aktuálních životů základen nám slouží grafický ukazatel životů nebo numerický ukazatelé. Grafické ukazatelé 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 bude červený.

Životy základny s 80%

Obrázek 9: Životy základny s 80% (vlastní zdroj)

Životy základny s 20%

Obrázek 10: 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 umře. Konkrétně před tím, než jednotka umře, tak 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 proměnou je označeno obrázkem, který definuje název proměnné.

Ukázka zobrazení hodnot zkušeností a peněz.

Obrázek 11: Ukazatel hodnot zkušeností a peněz (vlastní zdroj)

## Menu

Tato část bude obsahovat logiku použitou pro vytvoření menu, které obsahuje tři čudlíky. Každý čudlík obsahuje jinou funkci a logiku. Pozadí menu je doplněno videem, které se přehrává ve smyčce. Ve videu lze vidět všechny druhy postaviček, které jsou seřazeny od nejnovější doby po nejstarší.

### Hrát (Play)

První čudlík je hrát. Tento čudlík zaručuje, že jakmile ho hráč stiskne tak ho to pustí do hry a hra bude připravena ke hraní. Funkce je vytvořena za pomoci před vytvořeného příkazu pro načtení scény ve, které se nachází samotná hra.

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

{

SceneManager.LoadScene("GameScene");

LogScript.isGameOver = **false**;

}

Zdrojový kód č. 2

### Nastavení (Settings)

Čudlík s názvem nastavení je trošičku jiný od toho prvního. Pokud by byl aktivován, tak se scéna nezmění, ale upraví. V upravené scéně se bude nacházet jiný nadpis a bude obsahovat název autora a hudby i odkaz na autorův profil. V nastavení je také znázorněno, jak funguje pohyb kamerou ve hře obrázkem. Uživatel si také může upravit hlasitost hudby a herních efektů a také vyzkoušet hlasitosti.

Obsah obrázku text, snímek obrazovky, Písmo, design

Popis byl vytvořen automaticky

Obrázek 12: Vzhled nastavení (vlastní zdroj)

### Odejít (Exit)

Poslední čudlík se stará o odchod ze hry. Funkce čudlíku 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. 3

# Závěr

Při vytváření hry se naskytlo hodně problému. Některé chyby nebyly tak vážné a byly opraveny během pár chvilek, zatím což některé chyby dokázaly zabrat až týdny na opravení. Původní myšlenka některých věcí, které měly velké chyby byly změněny v něco jiného nebo odstraněny ze hry. S výsledkem práce jsem spokojený na půl, protože tam není vše, co jsem původně zamýšlel. Na ročníkové práci jsem pracoval od úplného počátku až ke konci s tím, že jsem se snažil pracovat pravidelně, což se mi ze začátku hodně dařilo i dokonce jsem pracoval více, než jsem očekával. Když bych měl vytknout část, která se mi nejméně dařila tak za mě by to bylo kreslení, které mi zabralo spoustu času. Na druhou stranu jsem se toho hodně naučil ohledně vývojového prostředí Unity a také ohledně kreslení obrázků a animování v softwaru Aseprite. 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 možná i kvalitnější.

# Přílohy

## Seznam obrázků

[Obrázek 1: Funkce Rigidbody 2D (vlastní zdroj) 11](#_Toc164182499)

[Obrázek 2: Funkce Box Collider 2D (vlastní zdroj) 11](#_Toc164182500)

[Obrázek 3: Kolize jednotek (vlastní zdroj) 11](#_Toc164182501)

[Obrázek 4: Animace jednotek (vlastní zdroj) 13](#_Toc164182502)

[Obrázek 5: Životy základny (vlastní zdroj) 14](#_Toc164182503)

[Obrázek 6: Čudlíky pro výrobu jednotek (vlastní zdroj) 15](#_Toc164182504)

[Obrázek 7: Fronta pro budoucí výrobu jednotek (vlastní zdroj) 15](#_Toc164182505)

[Obrázek 8: Grafický ukazatel výroby jednotek (vlastní zdroj) 16](#_Toc164182506)

[Obrázek 9: Životy základny s 80% (vlastní zdroj) 16](#_Toc164182507)

[Obrázek 10: Životy základny s 20% (vlastní zdroj) 16](#_Toc164182508)

[Obrázek 11: Ukazatel hodnot zkušeností a peněz (vlastní zdroj) 17](#_Toc164182509)

[Obrázek 12: Vzhled nastavení (vlastní zdroj) 18](#_Toc164182510)

## Zdroje

OpenAI. ChatGPT 3.5. [online]. [cit. 2024-04-13]. Dostupné z <https://openai.com/blog/chatgpt>

Fandom. Age of War Wiki. [online]. [cit. 2024-04-14]. Dostupné z <https://aowg.fandom.com/wiki/Age_of_War_Wiki>

Waterflame. Waterflame. [online]. [cit. 2024-04-15]. Dostupné z <https://www.waterflame.com/>

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. 2024-04-4].

Microsoft Corporation. (2022). Visual Studio IDE. [online] Dostupné z: <https://visualstudio.microsoft.com/>.

Smith, J. (2021). Mastering Visual Studio 2022. Redmond, WA: Microsoft Press.

Vytvořeno pomocí ChatGPT, OpenAI, 2024-04-03

Aseprite. (2022). Aseprite - Animated sprite editor & pixel art tool. [online] Dostupné z: <https://www.aseprite.org/>.

Johnson, C. (2019). Aseprite for Beginners: Animated Pixel Art in 2D. New York: O'Reilly Media.

Vytvořeno pomocí ChatGPT, OPenAI, 2024-04-03

1. Fandom. Age of War Wiki. [online]. [cit. 2024-04-14]. Dostupné z <https://aowg.fandom.com/wiki/Age_of_War_Wiki> [↑](#footnote-ref-1)
2. Waterflame. Waterflame. [online]. [cit. 2024-04-15]. Dostupné z <https://www.waterflame.com/> [↑](#footnote-ref-2)
3. 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. 2024-04-4]. [↑](#footnote-ref-3)
4. Microsoft Corporation. (2022). Visual Studio IDE. [online] Dostupné z: <https://visualstudio.microsoft.com/>.

   Smith, J. (2021). Mastering Visual Studio 2022. Redmond, WA: Microsoft Press.

   Vytvořeno pomocí ChatGPT, OpenAI, 2024-04-03 [↑](#footnote-ref-4)
5. Aseprite. (2022). Aseprite - Animated sprite editor & pixel art tool. [online] Dostupné z: <https://www.aseprite.org/>.

   Johnson, C. (2019). Aseprite for Beginners: Animated Pixel Art in 2D. New York: O'Reilly Media.

   Vytvořeno pomocí ChatGPT, OPenAI, 2024-04-03 [↑](#footnote-ref-5)
6. Tag v překladu znamená štítek. Ve vývojovém prostředí se tímto označují objekty [↑](#footnote-ref-6)
7. Layer v překladu znamená vrstva. Ve vývojovém prostředí se tímto určuje, zda to blíže ke kameře nebo dál. [↑](#footnote-ref-7)
8. Array v programování je proměnná, která je schopna ukládat více proměnných do jedné. [↑](#footnote-ref-8)