

**Obchodní akademie, Vyšší odborná škola a Jazyková škola  
s právem státní jazykové zkoušky Uherské Hradiště**



**MATURITNÍ PROJEKT**

**JUMP ADVENTURE**



Obchodní akademie, Vyšší odborná škola  
a Jazyková škola s právem státní jazykové  
zkoušky Uherské Hradiště

### ZADÁNÍ MATURITNÍ PRÁCE INFORMAČNÍ TECHNOLOGIE

Jméno žáka:	Michal Zapletal
Téma maturitní práce:	Hra v Unity - plošinovka
Vedoucí práce:	Mgr. Karel Jindra
OpONENT práce:	Mgr. Jaroslav Hodl
Způsob zpracování:	<p>Žák v enginu Unity a vytvoří 2D hru:</p> <p>Bude se jednat o plošinovku (platformer) s minimálně pěti levely zpracovanými grafickým stylem pixel art.</p> <p>Postup hrou bude primárně založen na správně promyšleném pohybu postavou úrovněmi a používáním různých předmětů/bonusů, vybrané pasáže úrovně budou připomínat hádanky, které je třeba k postupu vyřešit (například sebrat a použít klíč).</p> <p>Bude implementován dostačující počet těchto předmětů a bonusů (minimálně sedm). Obtížnost hry se bude s jednotlivými úrovněmi zvyšovat.</p> <p>Žák zároveň hře vytvoří jednoduchý příběh, který vysvětlí motivaci hlavní postavy.</p> <p>Hru otestují minimálně dva testéři a jejich poznatky budou zmíněny v dokumentaci.</p> <p>Projekt bude řádně veden na GitHubu.</p>
Pokyny k odevzdání:	<p><b>Žák odevzdá práci v tištěné a elektronické podobě</b></p> <ul style="list-style-type: none"><li>• Tištěná podoba práce obsahuje uživatelskou a technickou dokumentaci.</li></ul> <p><b>Tištěnou podobu</b> (v kroužkové vazbě) žák odevzdá v jedné verzi na studijní oddělení školy, místnost 311.</p> <p><b>Elektronická podoba práce obsahuje</b></p> <ul style="list-style-type: none"><li>• Dokumentaci ve formátu PDF/A</li><li>• Resumé ve formátu PDF/A</li><li>• Výsledný projekt, zdrojové soubory a potřebné knihovny pro spuštění projektu</li><li>• Prezentaci projektu</li></ul> <p><b>Elektronická podoba</b> práce se nahrává do IS školy dle pokynů vedoucího práce nebo vedení školy.</p> <p>V případě, že se jedná o projekt, na kterém pracovalo více žáků, je povinnou součástí dokumentace podrobné rozdělení činností při práci na projektu.</p>



Obchodní akademie, Vyšší odborná škola  
a Jazyková škola s právem státní jazykové  
zkoušky Uherské Hradiště

Kritéria hodnocení:	Hodnocení se skládá z celkové kvality zpracování práce, dokumentace, z kvality prezentace při obhajobě práce, diskuse a z průběžného hodnocení žáka v rámci kontrolních dnů.
Obhajoba projektu	Obhajoba projektu se skládá ze dvou částí - prezentace projektu (včetně podpůrné elektronické prezentace) a diskuse nad řešením. Celková délka obhajoby je 20 minut, délka prezentace projektu by neměla překročit 10 minut.

..... 17. 10. 2022 .....

Datum

Obchodní akademie, Vyšší odborná škola  
a Jazyková škola s právem státní  
jazykové zkoušky Uherské Hradiště  
Nádražní 22, 686 01 Uherské Hradiště  
IČO: 60371731, tel.: 572 433 011

  
.....  
Podpis ředitele školy

**Prohlášení:**

Souhlasím s tím, že s výsledky mé práce může být naloženo podle uvážení vedoucího maturitní práce a ředitele školy. V případě publikace budu uveden jako spoluautor.

Prohlašuji, že jsem na celé maturitní práci pracoval samostatně a veškeré použité zdroje jsem citoval.

V Uherském Hradišti, dne 31.3.2023

.....  
podpis absolventa

## RESUMÉ

*Aplikace JUMP ADVENTURE je hra vytvořena k tomu, aby uživatele zabavila na krátkou chvíli. Hra je jednoduchá na pochopení a dělaná tak aby ovládání bylo pro uživatele intuitivní. Na samotné ovládání hry nám postačí jen levé tlačítko myši a šipky nebo klávesy w, a, s, d. Celkové prostředí hry je zpracované tak, aby bylo jednoduché ji používat a uživatel v ní nebyl zmatený a grafická stránka aplikace je dělana do pixel art stylu. Cílem hráče je zvládnout projít různými překážkami a dostat se přes všechny levely.*

# OBSAH

<b>ÚVOD.....</b>	<b>7</b>
<b>1 UŽIVATELSKÁ DOKUMENTACE .....</b>	<b>8</b>
1.1 Příběh.....	8
1.2 Návod ke stažení hry .....	8
1.3 Prostedí hry .....	8
<b>2 PROGRAMÁTORSKÁ DOKUMENTACE .....</b>	<b>13</b>
2.1 Grafická část.....	13
2.1.1 Postup: .....	13
2.2 Wireframe.....	14
2.3 Scripty použité pro tento projekt: .....	18
2.4 Scény a v nich přidané GameObjects použité pro tento projekt:.....	19
2.5 GameObjects ve scénách Level 1, 2, 3, 4, 5, 6, 7 .....	20
2.6 Ukázka skriptů.....	34
<b>TESTOVÁNÍ.....</b>	<b>46</b>
<b>ZÁVĚR.....</b>	<b>49</b>
<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>50</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>51</b>

## ÚVOD

V dnešní době vychází spousta her. Řekl bych, že je to už neodmyslitelnou součástí spousta lidských životů. V takových hrách se lze totiž jednoduše odizolovat od světa a všeho stresu, který se kolem nás nachází. Také moderní doba už nabízí možnost si hraním her i vydělávat. Hlavně kvůli zahlcenosti trhu s hrami a také kvůli tomu, že se už od malička hraní her věnuji, jsem si řekl, že by nebylo vůbec na škodu se takové hry naučit programovat. Motivací je pro mě to, že bych se díky tomu mohl v budoucnosti věnovat něčemu, co mě baví a dále se v tom oboru posouvat. Proto jsem si vybral učení programování her ve vývojovém prostředí Unity.

Cílem mého projektu bude uživatelům nabídnout chvilkovou hru na pouhé odreagování a pobavení. Uživatelem může být kdokoli, kdo ovládá práci s počítačem. Ovládání zde totiž bude velmi jednoduché, ale bude potřeba se do rychlosti hry dostat. Také chci, aby samotné prostředí bylo pro uživatele snadné na pochopení. Určitě bude nutné navrhnout mechaniku hry tak, aby se lišila od ostatních her. Chci tedy, aby se hra nedala dohrát jen tím způsobem, že hráč dojde z bodu A do bodu B. Proto ji obzvláštním tak, aby v každém levelu bylo nějaké „ale“. Hráč se tedy musí dostat do cíle, ale musí při tom splnit nějakou podmínku. Nutné textury, které nejsou obsažené v balíčku z Asset Store budou vytvořeny mnou. Stavebním kamenem pro tuhle hru bude můj projekt ze třetího ročníku, který bude značně rozšířen o více předmětů, levelů a kritérií.

# 1 UŽIVATELSKÁ DOKUMENTACE

## 1.1 Příběh

Znáte ten pocit, když máte noční můru a probudíte se zpocení, vyděšení a plní hrůzy? Ano? To je dobře, protože náš hrdina takové štěstí nemá. Je zachycen v této noční můře, ze které se nemůže probudit a hledá cestu ven.

## 1.2 Návod ke stažení hry

Hra JUMP ADVENTURE je vytvořena pro operační systém Windows. Abyste ji mohli spustit musíte stáhnout z GitHubu: <https://github.com/GDGDJGH/JUMP-ADVENTURE.git> složku Hra, ve které se nachází následující soubory:

- JUMP ADVENTURE\_BurstDebugInformation\_DoNotShip
- MonoBleedingEdge
- JUMP ADVENTURE\_Data
- JUMP ADVENTURE.exe
- UnityCrashHandler64.exe
- UnityPlayer.dll

Samotná hra se spouští přes soubor JUMP ADVENTURE.exe. Po spuštění se vám jako první otevře hlavní menu, odkud se následně budete dostávat dále.

## 1.3 Prostředí hry

Většina tlačítek, která se ve hře objevují mají tu funkci, že když na ně uživatel najede myší změní se jejich barva na šedou. Tato funkce může posloužit k tomu, aby uživatel mohl rozpoznat, jestli aplikace nepřestala odpovídat a také je to uživatelsky příjemnější na použití hlavně kvůli tomu, že následně přesně ví, na co se dá kliknout a na co ne.



Obrázek 1 Tlačítko start





Obrázek 2 Tlačítko start s najetým kurzorem

Na hlavním menu můžete najít:

- Nápis „JUMP ADVENTURE“
- Tlačítko START
- Tlačítko CONTROLS
- Tlačítko LEVELS
- Tlačítko EXIT
- Jméno Autora
- Obrázek hlavní postavy zasazené do motivu hry



Obrázek 3 Hlavní menu hry

Ve chvíli, kdy chce uživatel opustit hru, klikne na tlačítko “EXIT“ a hra se automaticky vypne.

Po kliknutí na tlačítko “Controls“ bude uživatel přesměrován do scény, kde je zobrazeno, jak se má hra ovládat. Pro vrácení zpět do hlavního menu je zde tlačítko “Back“. V této scéně vidíme, že pro šplhání po žebříku má hráč využít klávesy w, s nebo šipky nahoru a dolů. Po tom, co hráč sebere luk může střílet levým tlačítkem myši. Pro pohyb doleva a doprava použije hráč klávesy a, d, nebo

šipky doleva a doprava. Pro stopnutí hry je klávesa ESC. Na závěr může vidět, že pro skok je zde mezník.



Obrázek 4 Controls

Po kliknutí na tlačítko “PLAY“ se již uživateli spustí samotná hra, kde má za úkol dostat se na druhou stranu mapy, kde jsou pro něj připraveny dveře, které ho přesunou do dalšího levelu. Při hraní je také možnost sbírání mincí, které hráči přičítají skóre. Hráč poté může chtít zkoušet nasbírat co největší skóre a více procházet mapu. Když hráč sebere minci přičte se +10 skóre a když sebere velkou minci přičte se +20 skóre. Také se při sebrání mince přehraje zvuk. V levém horním rohu můžeme vidět životy. Hráč má na začátku hry 3 životy. Tyto životy se odčítají ve chvíli, kdy hráč spadne na hroty, nebo když se dotkne nepřítele, lávy, nebo padajícího ostnu. Pod životy je zobrazen počet šípů, které hráč má. Šípy může získat tím, že se bere toulce, které se ve hře vyskytují. Na pravé straně je napsáno skóre. Pod skórem můžeme vidět čas. Čas se nachází pouze v některých levelích. Hráč musí v takovém případě dokončit level dřív, než se dostane odpočet na nulu.



Obrázek 5 7. level

Když hráč zmáčkne klávesu ESC zobrazí se mu pause menu, které stopne hru a následně může přejít do hlavního menu, do výběru levelů, nebo pokračovat ve hraní.



Obrázek 6 Pause Menu

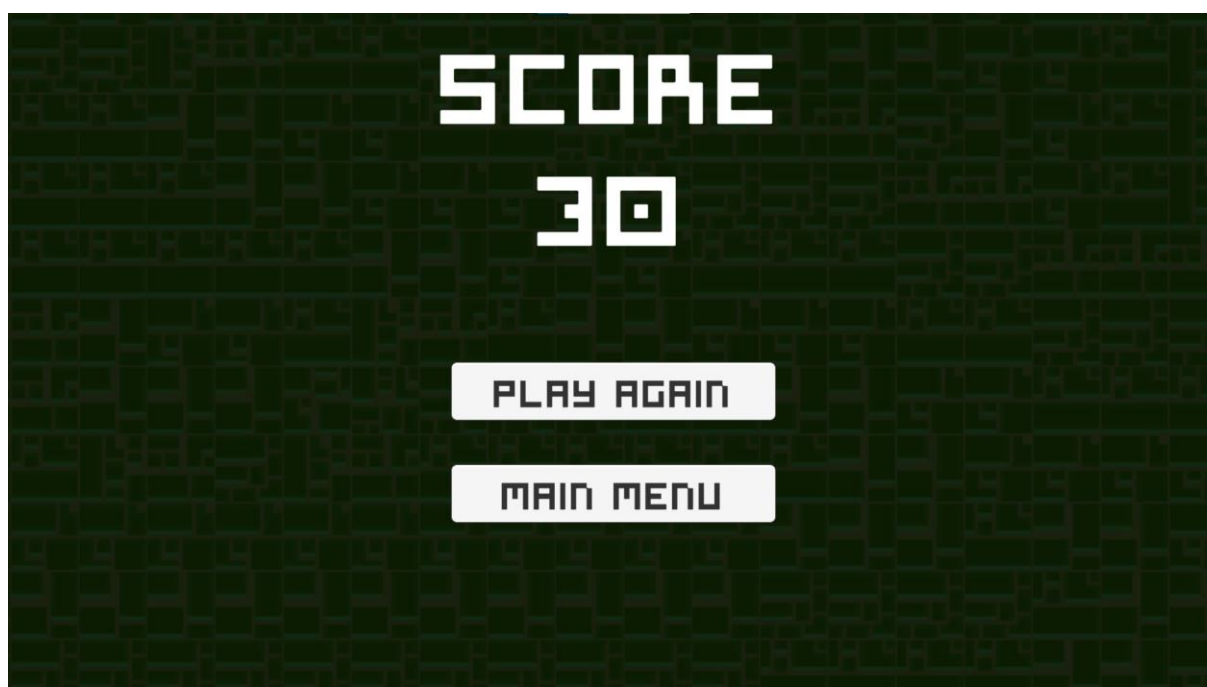
Hráč může přejít do scény LEVELS buď z hlavního menu, nebo z Pause Menu. Zde uvidí levely 1-7 a může si zapnout kterýkoliv z nich. Kliknutím na tlačítko s levellem se level spustí. V pravém dolním rohu je tlačítko BACK, kterým se uživatel dostane do hlavního

menu.



Obrázek 7 Levels

Jakmile hráč přijde o všechny životy, tak se objeví scéna, kde je napsané hráčovo skóre a dvě tlačítka. Tato scéna se také objeví ve chvíli, kdy se hráč dostane na konec sedmého levelu, kde úspěšně dokončí hru. Nachází se zde tlačítko “Play again“ po jehož kliknutí se hráč objeví znovu v první levelu a může si zkusit zapnout hru znovu a tlačítko “Main menu“ které hráče přesune do hlavního menu, ve kterém se také objeví, když zapne hru. Pokud hráč klikne na jakékoli z těchto tlačítek a bude chtít hrát znovu, skóre bude již resetované.



Obrázek 8 Endgame menu

## 2 PROGRAMÁTORSKÁ DOKUMENTACE

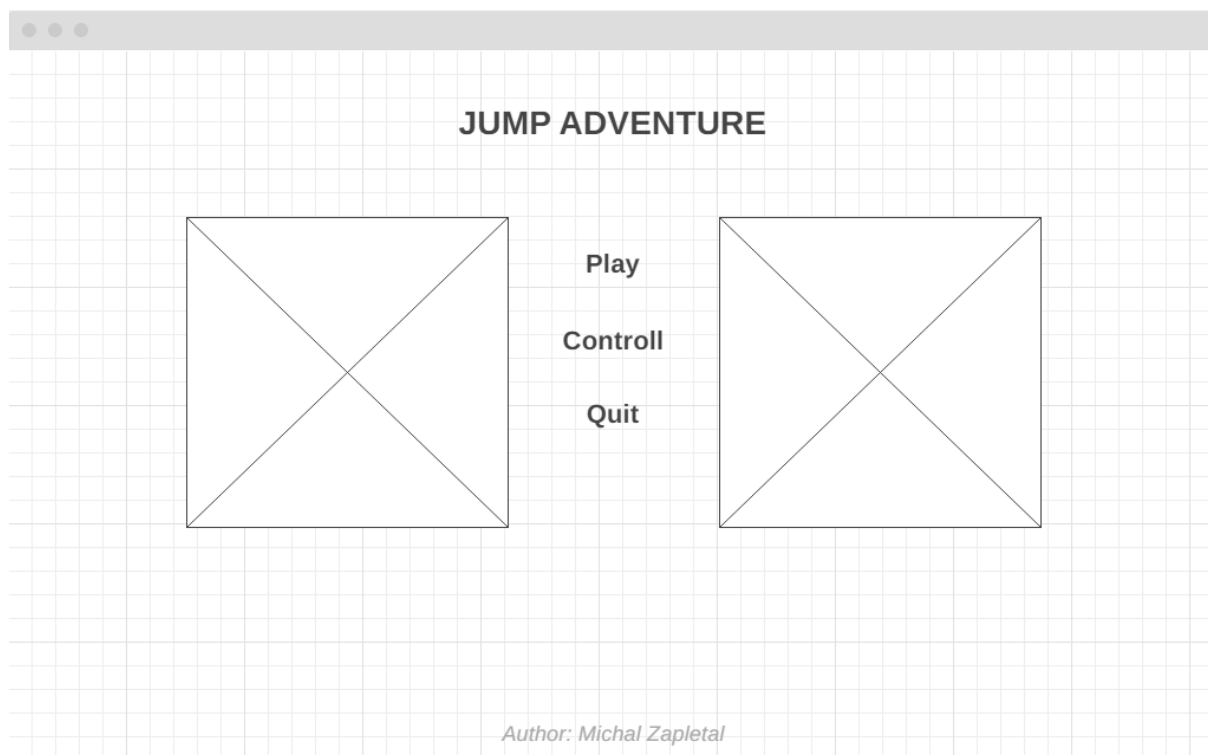
### 2.1 Grafická část

V projektu bylo nutné udělat několik textur, abych zde neporušoval autorská práva (obrázek 16, 17, 18, 19, 20, 21, 23, 24, 25). Pro tvorbu pixel artu jsem zvolil aplikaci Photoshop. Ostatní textury byly použity z (1). (Dříve zde byly textury zadarmo, nyní jsou na tomto odkazu tyto textury velmi pozměněné)

#### 2.1.1 Postup:

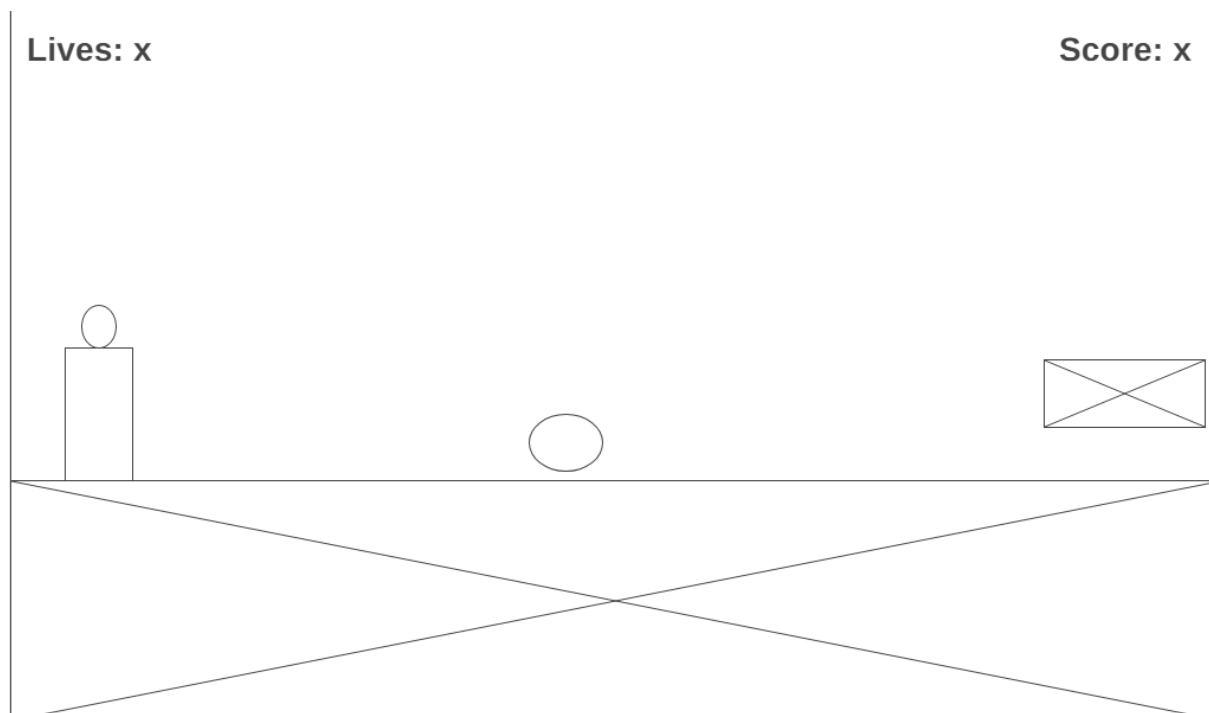
1. Založil jsem si nové plátno o velikosti 16x16 px.
2. Nastavil jsem si mřížky, které určují jednotlivé pixely
3. Nastavil jsem štětec na jeden pixel
4. Našel jsem si několik obrázků dané textury, ze kterých jsem se inspiroval
5. Texturu jsem nakreslil
6. Obrázek je poté nutné texturu zvětšit do větších rozměrů, jelikož je velmi malý a při použití by byl rozmazaný
7. Následně stačí exportovat do PNG a použít ho v Unity

## 2.2 Wireframe



**Obrázek 9 Wireframe hlavního menu**

Tento obrázek je náhled, jak bude vypadat hlavní menu. Je zde přechod do hry (tlačítko „Play“), přechod do ovládání (tlačítko „Controll“) a vypnutí hry (tlačítko „Quit“). Nad těmito tlačítky se nachází název hry. Po stranách budou obrázky a dole uprostřed se nachází jméno autora.



Obrázek 10 Wireframe hry

Tento obrázek je náhled, jak bude vypadat hra. Máme zde plošinu, na které se nachází hráč vlevo. Další tu je nepřítel, který se nachází uprostřed a vpravo je portál, kterým se hráč přenesení do dalšího levelu. Nad hráčem můžeme vidět životy, a nad portálem se nachází skóre.

## Funkce

1. **Pohyb hráče** (chození doleva a doprava, šplhání po žebříku, skok do výšky tří textur)
  - Hráč ovládá pohyb pomocí šipek nahoru, doleva a doprava, nebo za pomoci kláves w, a, s, d. Klávesu w/s, nebo šipku nahoru/dolů hráč využije pouze v případě, že šplhá po žebříku, nebo plave ve vodě. Pro skok slouží klávesa mezerník při jejímž stisku postavička vyskočí tři textury vysoko.
2. **Pohyb nepřítel** (chození doleva a doprava na konec platformy)
  - Nepřítel se pohybuje krátkými skoky v daném směru (tento pohyb bude automatický, tudíž ho nijak neovlivní přítomnost hráče), dokud nenarazí na konec platformy nebo do stěny. V takovém případě se otočí a skáče na druhou stranu
3. **Konec hry** (smrt: dotknutí nepřítel, spadnutí na hrot, hrot spadne na hráče, dotknutí se lávy, pokud dojde hráči čas; úspěšný konec: projití všech levelů)



- Smrt: ve chvíli, kdy hráč umře, odečte se mu jeden ze život a obnoví se na začátku levelu, ve kterém umřel. Ve chvíli, kdy hráč vyplývá všechny životy, musí začít od prvního levelu.
- Úspěšný konec: ten dosáhne hráč tím, že se dostane na konec mapy, kde se musí dotknout portálu, který ho přenesení dál.

#### 4. **Střelba z luku a zabití nepřítele** (počet šípů omezený)

- Hráč může sebrat luk až ve třetím levelu, kde se s ním může úspěšně bránit. Hráč vystřelí šíp levým tlačítkem myši. Ve chvíli, kdy zmáčkne toto tlačítko, tak se šíp vystřelí a poletí celou trasu rovně a ve chvíli, kdy narazí do nepřítele, nebo stěny, tak šíp zmizí. Nepřítele lze porazit pouze na jeden výstřel. Hráč může vystřelit jen tolikrát, kolik má nasbíraných šípů.

#### 5. **Hlavní menu** (vstup do hry, ovládání, vypnutí hry, levely)

- V hlavním menu jsou čtyři tlačítka nad kterými je velký nápis: „JUMP ADVENTURE“. Díky těmto tlačítkům se můžeme dostat do hracího pole, ovládání, ve kterém se hráč dozví, jak hru ovládat, do výběru levelů, nebo zde může hru vypnout.

#### 6. **Skóre**

- Po mapě jsou rozmístěny mince, které při sebrání hráči přičtou skóre + 10, nebo + 20 podle toho, jak velkou minci hráč sebere.

#### 7. **Koncové menu**

- Ve chvíli, kdy hráč vyčerpá všechny 3 životy, tak se zobrazí koncové menu, ve kterém je napsané jeho dosažené skóre a má možnost hru restartovat, nebo opustit.

#### 8. **Rule tile**

- Aby bylo vytváření mapy mnohem pohodlnější a jednodušší využijí zde možnost „Rule tile“, která fungují na principu sousedních dlaždit. Je zde tedy nastaveno, jak má každá textura, která není interaktivní, vypadat podle toho, kde se zrovna vyskytuje.

#### 9. **Interaktivní textury** (houba, žebřík, voda, láva)

- Houba: když hráč na tuto texturu stoupne, vyhodí ho to do vzduchu. Výška odražení se odráží od toho, kolikrát již hráč na tuto texturu dopadnul.



- Žebřík: hráč po žebříku může šplhat pomocí klávesy w/s, nebo šipky nahoru/dolů. Při šplhání je hráč zpomalen.
- Voda: hráč, který se nachází ve vodě je zpomalen. Může se ve vodě pohybovat všemi směry
- Láva: pokud se hráč dotkne lávy odebere se mu jeden život. Pokud sebere elixír, láva pak funguje jako voda

#### 10. **Kamera**

- Kamera je tvořena tak, aby následovala hráče a byla vidět jen malá část mapy. Když se hráč nebude nějakou dobu pohybovat kamera se sama více přiblíží k hráči, ale když bude chodit kamera se oddálí.

#### 11. **Animace**

- Animace jsou zde pro případy, když hráč stojí na místě, běží nebo šplhá po žebříku. Další z animací je zde pro mince. Tato animace ukazuje, jak se mince točí.

#### 12. **Boty**

- Pokud hráč sebere boty, zvýší se mu rychlost do konce levelu, ve kterém boty získal. Pokud v tomto levelu umře, musí boty vzít znovu.

#### 13. **Toulec**

- Po mapě jsou rozmístěny toulce se šípy, které když hráč vezme tak se mu šípy přičtou a následně je může vystřelit.

#### 14. **Srdce**

- Po mapě jsou rozmístěné srdce, které když hráč vezme tak se mu přičte jeden život. Život jde v jednom levelu vzít pouze jednou, a to i když hráč umře.

#### 15. **Hodiny**

- Některé levely jsou limitované časem. Za konkrétní čas se bude muset hráč dostat na konec levelu. Proto jsou v takových levelech umístěny přesýpací hodiny, které když hráč sebere, tak mu přičtou nějaký čas.

#### 16. Elixír

- Pokud hráč sebere elixír, bude moci plavat v lávě

#### 17. Klíč a terč

- Pokud jeden z těchto předmětů v levelu existují, tak budou dveře pro přechod do dalšího levelu uzavřeny.
- Klíč: je nutné ho sebrat
- Terč: je nutné ho zasáhnout šípem

#### 18. Pause Menu

- pokud hráč zmáčkne klávesu ESC spustí se Pause Menu. V tomto menu může ve hře zase pokračovat, zvolit si level, nebo přejít do Hlavního Menu.

### 2.3 Scripty použité pro tento projekt:

- ArrowSO
- ArrowPickUp
- Arrow
- Boots
- BowPickup
- Clock
- CoinPickup
- CoinPickupManager
- EnemyMovement
- FallingHazzard
- GameSession
- Heart
- Key
- LevelExit
- PauseMenu

- PlayerMovement
- QuiverPickUp
- SceneControll
- ScenePersist
- ScoreManager
- Target
- Timer

## **2.4 Scény a v nich přidané GameObjects použité pro tento projekt:**

- Controls
  - Canvas (Tlačítka a texty)
  - Tilemap Grid
  - SceneControll
- EndGameMenu
  - Tilemap Grid
  - Canvas (Tlačítka a texty)
  - CoinPickUp
  - SceneManager
- MainMenu
  - Canvas (Tlačítka a texty)
  - Tilemap Grid
  - SceneControll
- Level 1, 2, 3, 4, 5, 6, 7
  - Cameras
  - Exit
  - Game Session
  - Player
  - Tilemap Grid
  - Enemies
  - CoinPickUp
  - ArrowPickUp
  - BowManager
  - Quivers

- Scene Persist (mince, srdce)
- CoinPickUp
- Bow (Pouze ve 3. levelu)

## 2.5 GameObjects ve scénách Level 1, 2, 3, 4, 5, 6, 7

### Main Camera

Komponenty:

- CinemachineBrain
  - Tento komponent je určený k tomu, aby kamera následovala hráče. Je zde možnost přidat GameObject, který obsahuje kamery, které se uzpůsobují ději hráče. Pro tuto možnost je zde přiřazen GameObject **State Driven Camera**, který obsahuje právě tyto ději přizpůsobující se kamery.
- State Driven Camera
  - Jak jsem již zmínil v předchozím bodu, tak se zde nachází kamery zakládající se na ději. Pro bližší představu je zde nastaveno, aby se kamera přibližovala nebo oddalovala podle toho, co zrovna hráč dělá. Také je zde nastaveno, jak dlouho se má oddalovat, nebo přibližovat a také je možné zde nastavit, jak má kamera dlouho čekat, než se začne přibližovat nebo oddalovat, pokud již hráč změnil svůj děj. Pro tuto možnost zde byl přidán komponent CinemachineStateDrivenCamera, kde jdou všechny tyto možnosti nastavit.

From	To	Style	Time
Run Camera	Idle Camera	Ease In Out	s 5
Idle Camera	Run Camera	Ease In Out	s 1
**ANY CAMERA**	Ladder Camera	Ease In Out	s 0.5
Ladder Camera	**ANY CAMERA**	Ease In Out	s 3

Obrázek 11 Přechod kamer

Zde můžeme vidět natavení pro čas, který proběhne, než se kamera dostane do konečné vzdálenosti, ve které se má na konci ocitnout. Můžeme zde vidět v prvním případě, že pokud hráč běží a poté se zastaví, tak se kamera bude přibližovat po dobu pěti sekund. Například v případě třetím je zde nastaveno, že pokud bude hráč dělat

cokoli jiného, než šplhat po žebříku a poté začne šplhat, tak se kamera bude měnit po dobu půl sekundy.

State	Camera	Wait	Min
Running	Run Camera	0	0
Idling	Idle Camera	1	0
Climbing	Ladder Camera	0	0

**Obrázek 12 Čas před přechodem kamer**

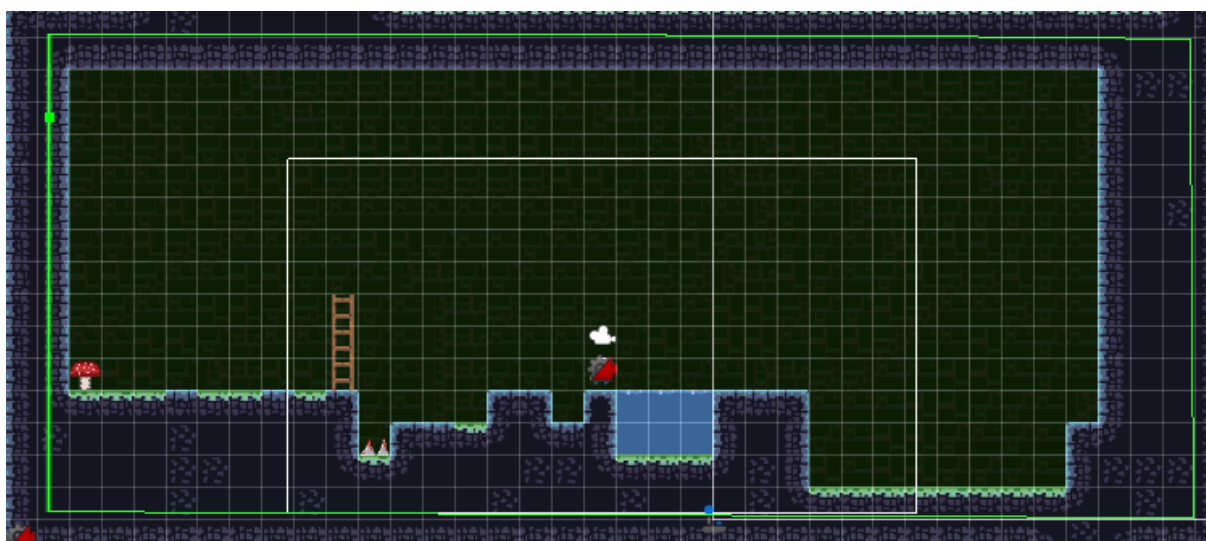
Zde můžeme vidět nastavení pro čas, který v určitých momentech kamera počká, než přijde oddalování nebo přibližování. Tato možnost je zde využita pouze v případě, že hráč stojí a něco udělá, nebo když něco dělá a zastaví se.



**Obrázek 13 Kamery k dějům**

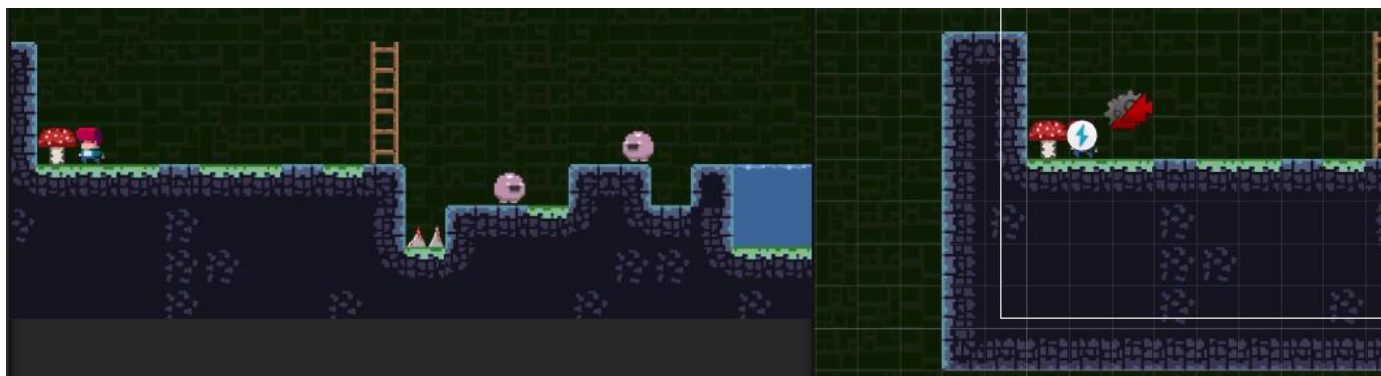
Abychom mohli kamery vybírat, tak zde je i nutnost umístit do tohoto komponentu kamery ve kterých už jsou podstatné vědci nastaveny, a tak je možné s nimi dále pracovat, jak jsme mohli vidět v předchozích dvou obrázcích.

- Run Camera, Idle Camera, Ladder Camera
  - Tohle jsou už specifické kamery pro konkrétní děj, se kterými se pracuje v předchozím komponentu. Kromě dané vzdálenosti, které mají mít kamery od hráče v konkrétních dějích je zde i nastaveno pro každý level to, aby kamera nepřesahovala ohraničení mapy.



**Obrázek 14 Ohraničení mapy pro kameru**

Zelená čára zde ukazuje ohraničení mapy, přes které kamera nemůže, a tudíž uživatel neuvidí, co se za tímto ohraničením nachází. Teď uvidíte obrázek, jak to vypadá pro hráče s ohraničenou kamerou, a jak mapa vypadá pro programátora.

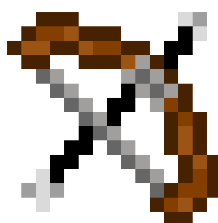


Obrázek 15 Uživatel vs programátor

## **Bow**

Komponenty:

- Sprite Renderer
  - Vykreslí tvar pro 2D grafiku. Je zde uložen obrázek luku.



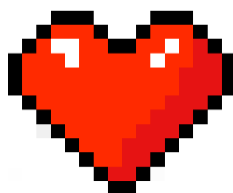
Obrázek 16 Luk

- Box Collider 2D
  - Neviditelný obdélník, který zajišťuje zjištění doteku s tímto objektem.
- Script: BowPickup
  - Zde je naprogramováno chování luku.

## **Heart**

Komponenty:

- Sprite Renderer
  - Vykreslí tvar pro 2D grafiku. Je zde uložen obrázek srdce



Obrázek 17 Textura srdce

- Capsule Collider 2D
  - Neviditelný kapslový tvar, který zajišťuje zjištění doteku s tímto objektem.
- Script: Heart
  - Zde je naprogramováno chování srdce.

### Elixír

Komponenty:

- Sprite Renderer
  - Vykreslí tvar pro 2D grafiku. Je zde uložen obrázek elixíru



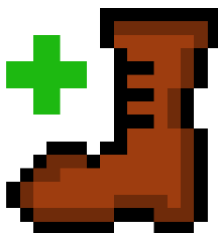
Obrázek 18 Textura elixíru

- Capsule Collider 2D
  - Neviditelný kapslový tvar, který zajišťuje zjištění doteku s tímto objektem.
- Script: Elixir
  - Zde je naprogramováno chování elixíru.

### Boots

Komponenty:

- Sprite Renderer
  - Vykreslí tvar pro 2D grafiku. Je zde uložen obrázek boty



Obrázek 19 Textura boty

- Capsule Collider 2D
  - Neviditelný kapslový tvar, který zajišťuje zjištění doteku s tímto objektem.
- Script: Boots
  - Zde je naprogramováno chování boty.

## Clock

Komponenty:

- Sprite Renderer
  - Vykreslí tvar pro 2D grafiku. Je zde uložen obrázek přesýpacích hodin



Obrázek 20 Textura přesýpacích hodin

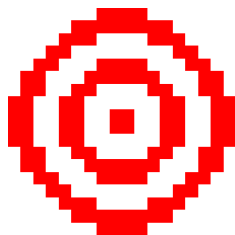
- Capsule Collider 2D
  - Neviditelný kapslový tvar, který zajišťuje zjištění doteku s tímto objektem.
- Script: Clock
  - Zde je naprogramováno chování přesýpacích hodin.

## Target

Komponenty:

- Sprite Renderer
  - Vykreslí tvar pro 2D grafiku. Je zde uložen obrázek terče





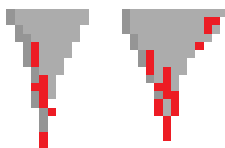
Obrázek 21 Textura terče

- Capsule Collider 2D
  - Neviditelný kapslový tvar, který zajišťuje zjištění doteku s tímto objektem.
- Script: Target
  - Zde je naprogramováno chování terče

### **Falling Hazzard**

Komponenty:

- Sprite Renderer
  - Vykreslí tvar pro 2D grafiku. Je zde uložen obrázek padajících ostnů



Obrázek 22 Textura padajících ostnů

- Capsule Collider 2D
  - Neviditelný kapslový tvar, který zajišťuje zjištění doteku s tímto objektem.
- Box Collider 2D
  - Neviditelný obdélník, který zajišťuje zjištění doteku s tímto objektem.
- Rigidbody 2D
  - Slouží k řízení polohy objektu pomocí fyzikální simulace.
- Script: FallingHazzard
  - Zde je naprogramováno chování padajících ostnů

### **Key**

Komponenty:

- Sprite Renderer

- Vykreslí tvar pro 2D grafiku. Je zde uložen obrázek klíče.



**Obrázek 23 Textura klíče**

- Box Collider 2D
  - Neviditelný obdélník, který zajišťuje zjištění doteku s tímto objektem.
- Script: Key
  - Zde je naprogramováno chování klíče.

### **Quiver**

Komponenty:

- Sprite Renderer
  - Vykreslí tvar pro 2D grafiku. Je zde uložen obrázek toulce.



**Obrázek 24 Textura toulce**

- Box Collider 2D
  - Neviditelný obdélník, který zajišťuje zjištění doteku s tímto objektem.
- Script: QuiverPickUp
  - Zde je naprogramováno chování toulce.

### **Exit**

Komponenty:

- Sprite Renderer

- Vykreslí tvar pro 2D grafiku. Je zde uložen obrázek dveří.

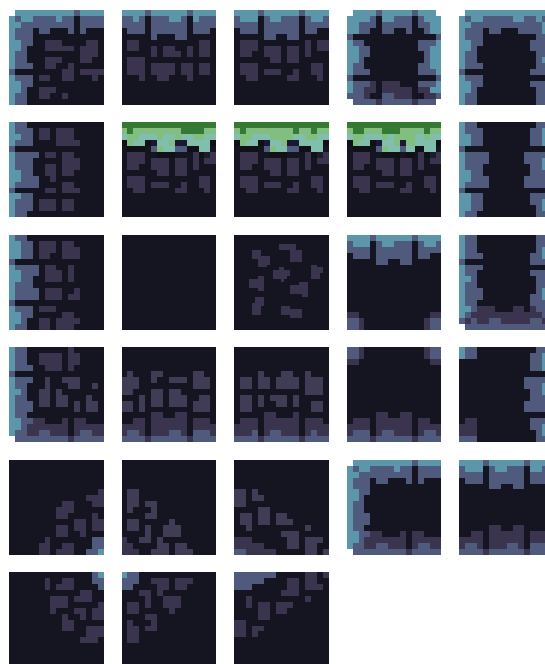


**Obrázek 25 Textura dveří**

- Box Collider 2D
  - Neviditelný obdélník, který zajišťuje zjištění doteku s tímto objektem.
- Script: Level Exit
  - Zde je naprogramováno chování dveří.

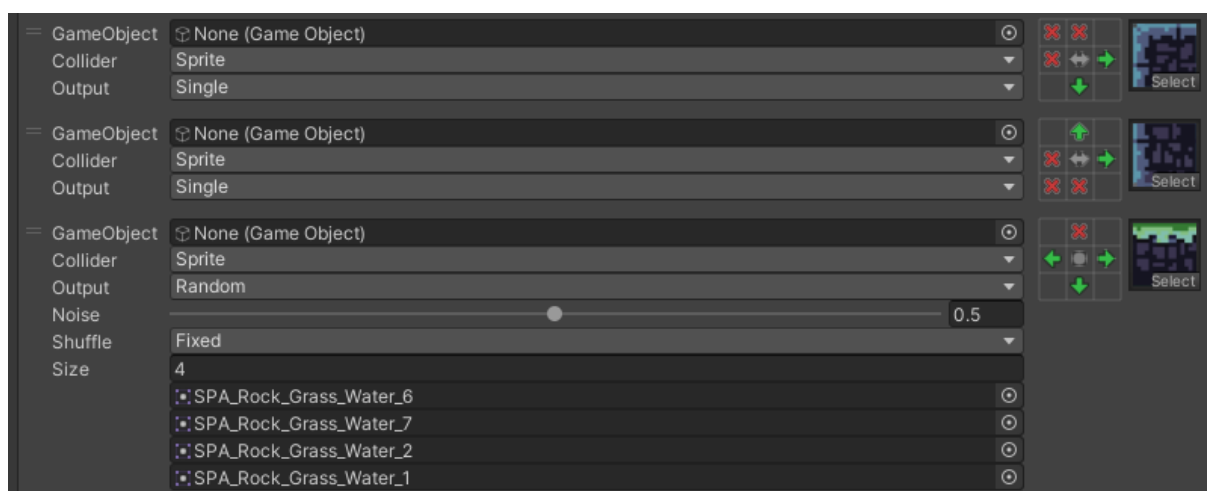
### **Platforms Tilemap, Background Tilemap, Water Tilemap, Climbing Tilemap, Bouncing Tilemap, Hazards Tilemap, Lava Tilemap**

- Tilemap je stažený balíček pro unity, který využívá kreslení textur na principu dlaždic. Díky tomuto balíčku jsme schopni jednoduše kreslit, mazat, nebo upravovat textury. Hlavní výhodou je to, že se tyto textury zachytávají do čtverečků a díky tomu na sebe textury krásně navazují. V tomto projektu je spousta druhů Tilemap, protože každá má jiné chování.
- Platform Tilemap
  - Tyhle dlaždice jsou využity pro platformy, které jsou nejvíce v popředí a nemůže přes ně být žádná z dalších dlaždic. Po těchto dlaždicích se hráč pohybuje. Pro platformy jsou využity tyto textury:



Obrázek 26 Textury platformem

- Abych však nemusel všechny dlaždice ručně vybírat a dávat je tam, kde pasují, využil jsem zde funkci unity jménem „Rule Tile“. Tato funkce slouží tak, že se zde umístí všechny textury a vytvoříme pro mě pravidla, tak abychom nastavili, kde se mají poté umísťovat. Samotný Rule Tile je poté uložen jako jedna textura v Tilemap paletě, kterou můžeme vybrat a následně s ní už jen kreslit mapu, kde se dlaždice podle pravidel upravují. Jelikož je zde i možnost použít více textur dlaždic pro jedno pravidlo, tak je v daném pravidlu uloženo více textur dlaždic, které se náhodně vybírají. Pravidla pro tvorbu poté vypadají následovně:

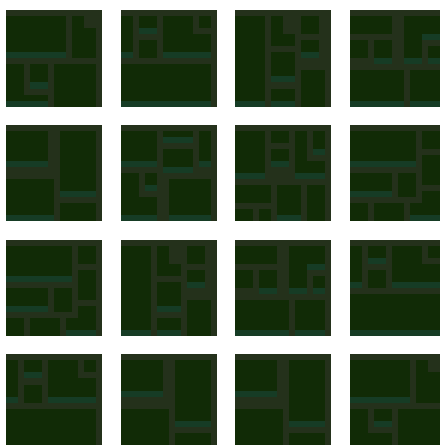


Obrázek 27 Rule Tile

Červené křížky znázorňují, že v této pozici se nesmí nacházet žádná dlaždice. Zelené šipky znázorňují, že se v této pozici musí nacházet dlaždice a pokud zde není ani

křížek ani šipka, tak to znamená, že na tom, jestli tam dlaždice bude, nebo nebude vůbec nezáleží. Ve středu pravidla se nachází také možnosti, ve kterých směrech a rotacích bude pravidlo fungovat. Může se zde nacházet šipka doleva i doprava a to znamená, že se pravidlo bude zrcadlit po ose x. Dále zde je tečka a ta znamená že je pravidlo fixní, a tudíž se nemění. Další z pravidel je zatočená šipka doprava a ta znamená, že toto pravidlo bude fungovat po devadesáti stupních. Nakonec je zde ještě pravidlo, kde je šipka nahoru a dolů a to znamená, že se bude pravidlo zrcadlit po ose y. Ve třetím případě na obrázku můžeme vidět případ, kdy se na dané pravidlo hodí více textur dlaždic a je zde tedy i více textur dlaždic uloženo a nastaveno a by se náhodně generovali.

- Background Tilemap
  - Tyhle dlaždice jsou využity pro pozadí mapy. Dlaždice jsou nejvíce v pozadí, a tudíž jakákoliv textura je může překrýt. Pro pozadí jsou využity tyto textury:



Obrázek 28 Textury pozadí

- Water Tilemap
  - Tyhle dlaždice jsou využity pro vodu. Pokud hráč vkročí do vody, je zpomalen, padá pomalu ke dnu a nemůže z vody vyskočit. Jediná možnost, jak se z vody dostat je při držení šipky nahoru, kdy hráč začne plavat směrem vzhůru. Pro vodu jsou využity tyto textury:



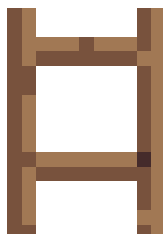
Obrázek 29 Textury vody

- Lava Tilemap
  - Tyhle dlaždice jsou využity pro lávu. Pokud se hráč lávy dotkne umře a je mu sebrán jeden život. Pokud však hráč sebere elixír, tak se láva chová stejně jako voda. Textura lávy je převzata od vody, pouze zbarvená do oranžova.



Obrázek 30 Textura lávy

- Target Tilemap
  - Tyhle dlaždice jsou využity pro falešný terč. Ve hře se nachází level, kde musí hráč najít opravdový cíl, který když zasáhne odemknou se mu dveře. Opravdový terč je však vytvořen jako GameObject. Všechny ostatní jsou tvořeny pomocí Tilemapy a nic nedělají, proto jsou nazvány falešné.
- Climbing Tilemap
  - Tyhle dlaždice jsou využity pro žebřík. Při šplhání po žebříku, nebo pokud na něm hráč pouze stojí, vypne se gravitace a vypadá to, jak kdyby se hráč na žebříku držel po jeho opuštění je gravitace opět vrácena do normálu. Pro šplhání jsou využity šipky nahoru a dolů, přičemž se spustí animace šplhání. Pro žebřík jsou využita tato textura:



Obrázek 31 Textura žebříku

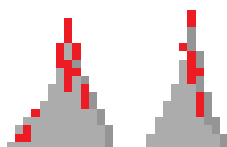
- Bouncing Tilemap

- Tyhle dlaždice jsou využity pro odrazení do vzduchu. Při dotknutí houby se hráč odrazí do vzduchu. Z čím větší výšky na ni hráč spadne tím více se odrazí. Pro odrazení jsou využita tato textura:



Obrázek 32 Textura houby

- Hazards Tilemap
  - Tyhle dlaždice jsou využity jako překážka ve hře. Při dotknutí se hrotu se hráči odebere jeden život. Ve hře je tedy umístněný tak, aby se mu hráč musel vyhnout a kvůli tomu hra nebyla tolik jednoduchá. Pro hroty je využita tato textura:



Obrázek 33 Textura hrotů

## Game Session

- Obsahuje text pro zobrazení životů. Tento text má pixel artový font. Dále obsahuje Pause Menu

Komponenty:

- Script: GameSession

## Timer

- Obsahuje text pro zobrazování času a také texturu přesýpacích hodin

Komponenty:

- Script: Timer

## ArrowPickUpManager

- Obsahuje text pro zobrazování počtu šípů a také texturu šíp

Komponenty:

- Script: ArrowPickUpManager

## **Player**

Komponenty:

- Animator
  - Jsou zde nastavené animace pro jednotlivé děje.
- Sprite Renderer
  - Vykreslí tvar pro 2D grafiku. Je zde uložen obrázek postavy.



**Obrázek 34 Textura postavičky**

- Capsule Collider 2D
  - Neviditelný kapslový tvar, který zajišťuje zjištění doteku s tímto objektem.
- Rigidbody 2D
  - Slouží k řízení polohy objektu pomocí fyzikální simulace.
- Player Input
  - Ovládání postavičky.
- Box Collider 2D
  - Neviditelný čtverec, který zajišťuje zjištění doteku s tímto objektem. Ten tady slouží pouze k zabezpečení, aby se hráč nemohl „přilepit“ na stěnu.
- Script: Player Movement
  - Zde je naprogramováno chování postavy.

## **BowPosition**

- Slouží k nastavení pozice, ze které bude vylétat šíp při střelení.

## **Scene Persist**



- Když hráč sebere nějaký objekt, který se zde nachází a umře, tak se tento objekt už znovu neobjeví. Pokud však hráč přijde o všechny životy tak se tyto předměty znovu objeví v daných levelech.

Komponenty:

- Script: ScenePersist

## **Enemy**

Komponenty:

- Sprite Renderer
  - Vykreslí tvar pro 2D grafiku. Je zde uložen obrázek nepřítele.



**Obrázek 35 Textura nepřítele**

- Rigidbody 2D
  - Slouží k řízení polohy objektu pomocí fyzikální simulace.
- Capsule Collider 2D
  - Neviditelný kapslový tvar, který zajišťuje zjištění doteku s tímto objektem.
- Animator
  - Je zde nastavena animace skákání.



**Obrázek 36 Animace nepřítele**

- Box Collider 2D
  - Neviditelný čtverec, který zajišťuje zjištění doteku s tímto objektem.
- Script: EnemyMovement

## **Coin**

Komponenty:

- Sprite Renderer

- Vykreslí tvar pro 2D grafiku. Je zde uložen obrázek mince.



**Obrázek 37 Textura mince**

- Circle Collider 2D
  - Neviditelný kruh, který zajišťuje zjištění doteku s tímto objektem.
- Animator
  - Je zde nastavena animace točení mince.



**Obrázek 38 Animace mince**

- Audio Source
  - Přehraje zvuk, když hráč sebere minci.
- Skript: CoinPickup

### **CoinPickUp**

- Zobrazuje skóre, které má hráč nahrané. Využívá se také ve chvíli, kdy se zobrazuje koncové menu a je zde zobrazeno nahrané skóre hráče.

Komponenty:

- Script: CoinPickUpManager

## **2.6 Ukázka skriptů**

### **BowPickup**

Metody:

- **OnTriggerEnter2D()**

Slouží k tomu, aby hráč nemohl střílet hned od začátku hry, ale až poté co sebere luk. Ve chvíli, kdy se hráč luku dotkne, tak je luk zničen a nastaví se proměnná hasBow ve skriptu PlayerMovement na true.

```
private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.tag == "Player") {
        FindObjectOfType<PlayerMovement>().SetHasBow(true);
        Destroy(gameObject);
    }
}
```

Obrázek 39 BowPickup metoda OnTriggerEnter2D

## Arrow

Metody:

- **Start()**

Na začátku se zjistí, v jaké pozici je hráč postaven a podle toho se nastaví směr letícího šípu. Tento směr je vynásoben rychlostí.

```
xSpeed = player.transform.localScale.x * bulletSpeed;
```

- **OnTriggerEnter2D()**

Pokud šíp zasáhne nepřítele, tak je nepřítel i šíp zničen.

```
void OnTriggerEnter2D(Collider2D other)
{
    if(other.tag == "Enemy")
    {
        Destroy(other.gameObject);
    }
    Destroy(gameObject);
}
```

Obrázek 40 Arrow metoda OnTriggerEnter2D

- **OnCollisionEnter2D()**

Pokud se šíp dotkne stěny, je zničen.

```
void OnCollisionEnter2D(Collision2D other)
{
    Destroy(gameObject);
}
```

Obrázek 41 Arrow metoda OnCollisionEnter2D

- **flipArrow()**

Pokud se hráč otočí a poté vystřelí, otočí se i textura šípu.

```
public void flipArrow(){
    if (!player.GetPlayerRotation())
    {
        ArrowRotation.flipX = true;
    }
}
```

Obrázek 42 Arrow metoda flipArrow

## CoinPickup

Metody:

- **OnTriggerEnter2D()**

Slouží k sebrání mince při dotyku s postavou. Zabezpečuje, aby nemohla být mince v rámci „bugu“ sebrána dvakrát. Zároveň se přehraje zvuk sebrání mince, samotná mince se zničí a přičte se +10 skóre.

```
void OnTriggerEnter2D(Collider2D other)
{
    if (other.tag == "Player" && !wasCollected)
    {
        wasCollected = true;
        score.AddToScore(pointsForCoinPickup);
        AudioSource.PlayClipAtPoint(coinPickupSFX, Camera.main.transform.position);
        gameObject.SetActive(false);
        Destroy(gameObject);
    }
}
```

Obrázek 43 CoinPickup metoda OnTriggerEnter2D

## CoinPickUpManager

- Slouží k zapisování skóre a resetování skóre.

## EnemyMovement

Metody:

- **OnTriggerExit2D()**

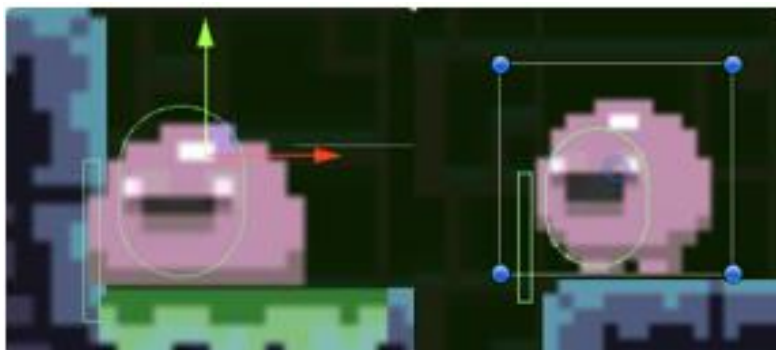
Pohyb nepřítele po platformě. Nepřítel se pohybuje rovně po platformě, dokud jeho Box Collider 2D, který se nachází před ním, neopustí platformu na které stojí. V takový moment se zneguje rychlost chůze a zavolá se metoda FlipEnemyFacing()

```

void OnTriggerExit2D(Collider2D other)
{
    moveSpeed = -moveSpeed;
    FlipEnemyFacing();
}

```

Obrázek 44 EnemyMovement metoda OnTriggerEnter2D



Obrázek 45 Enemy BoxCollider2D

- **FlipEnemyFacing()**

Tato metoda zneguje texturu nepřítele a ta se vzápětí otočí na druhou stranu ve směru chůze

```

void FlipEnemyFacing()
{
    transform.localScale = new Vector2 (-(Mathf.Sign(myRigidbody.velocity.x)), 1f);
}

```

Obrázek 46 EnemyMovement metoda FlipEnemyFacing (2)

## GameSession

- Využitá k tomu, aby zachycovala a vypisovala hráčovi životy.

Metody:

- **ProcessPlayerDeath()**

Pokud má hráč víc než jeden život, tak mu to život sebere, jinak se načte koncové menu a životy se resetují.

```

public void ProcessPlayerDeath()
{
    if (playerLives > 1)
    {
        TakeLife();
    }
    else
    {
        SceneManager.LoadScene(3);
        ResetGameSession();
    }
}

```

Obrázek 47 GameSession metoda Process PlayerDeath

- **TakeLife()**

Ubere hráči život, zjistí a načte momentální scénu a přepíše životy na aktuální životy.

```

void TakeLife()
{
    playerLives--;
    int currentSceneIndex = SceneManager.GetActiveScene().buildIndex;
    SceneManager.LoadScene(currentSceneIndex);
    livesText.text = playerLives.ToString();
}

```

Obrázek 48 GameSession metoda TakeLife

## LevelExit

Metody:

- **LoadNextLevel()**

Pokud se hráč nenachází na posledním levelu, tak se hráč přenesení do levelu následujícího.

Pokud se však nachází na posledním levelu, tak se načte koncové menu. Než se však hráč přenesení do dalšího levelu je zde nastavený menší odpočet, který se zapne a na jeho konci je tedy hráč přenesen do určené scény.

```

IEnumerator LoadNextLevel()
{
    yield return new WaitForSecondsRealtime(levelLoadDelay);
    int currentSceneIndex = SceneManager.GetActiveScene().buildIndex;
    int nextSceneIndex = currentSceneIndex + 1;

    if (nextSceneIndex == SceneManager.sceneCountInBuildSettings)
    {
        FindObjectOfType<GameSession>().ResetGameSession();
        nextSceneIndex = 3;
    }

    FindObjectOfType<ScenePersist>().ResetScenePersist();
    SceneManager.LoadScene(nextSceneIndex);
}

```

Obrázek 49 LevelExit metoda LoadNextLevel (1)

- **OnTriggerEnter2D()**

Metoda zjišťuje, jestli se hráč dotkl dveří. Pokud je na mapě klíč, který je potřeba vzít, nebo terč, který je potřeba trefit tak se kolize nezaznamená. Pokud žádný z těchto objektů ve není a hráč se dveří dotkne, tak ho to přesune do dalšího levelu.

```
void OnTriggerEnter2D(Collider2D other)
{
    if (Key || Target)
    {
        if (FindObjectOfType<PlayerMovement>().HasKey())
        {
            if (other.tag == "Player")
            {
                StartCoroutine(LoadNextLevel());
            }
        }
    }
    else
    {
        if (other.tag == "Player")
        {
            StartCoroutine(LoadNextLevel());
        }
    }
}
```

Obrázek 50 LevelExit metoda OnTriggerEnter2D

## PlayerMovement

Metody:

- **OnFire()**

Je to metoda, která je již přednastavena Unity a spustí se ve chvíli, kdy zmáčkne levé tlačítko myši. Pokud hráč nežije, nevlastní luk, počet šípů je menší než 0, nebo je aktivní pause menu tak se nic nestane. Ale pokud ani jedno z těchto pravidel neplatí tak se vytvoří instance šípu na úrovni, kde je vytvořený GameObject **BowPosition** a z tohoto místa se už dále šíp chová podle skriptu **Arrow**. Počet šípů se aktualizuje.

```
void OnFire(InputValue value)
{
    if (!isAlive || hasBow == false || FindObjectOfType<ArrowsPickUpManager>().GetArrows() <= 0 || FindObjectOfType<PauseMenu>().GetIsPaused() == true) { return; }
    FindObjectOfType<ArrowsPickUpManager>().updateArrows();
    float playerRotation = Mathf.Sign(myRigidbody.velocity.x);
    Instantiate(bullet, bow.position, transform.rotation);
}
```

Obrázek 51 PlayerMovement metoda OnFire (1)

- **GetPlayerRotation()**

V této metodě zjistíme hráčovu rotaci, která je využita ve skriptu **Arrow**.

```

public bool GetPlayerRotation(){
    if(transform.localScale.x == -1)
    {
        isLookingRight = false;
    }
    else
    {
        isLookingRight = true;
    }
    return isLookingRight;
}

```

Obrázek 52 PlayerMovement metoda GetPlayerRotation (1)

- OnJump()

Je to metoda, která je již přednastavena Unity a spustí se ve chvíli, kdy zmáčkne mezerník. Tato metoda je využita k tomu, aby mohl hráč skákat a zabráňuje tomu, aby hráč mohl skočit pouze jednou a poté znovu až se dotkne podlahy.

```

void OnJump(InputValue value)
{
    if (!isAlive) { return; }
    if (!myFeetCollider.IsTouchingLayers(LayerMask.GetMask("Ground"))) { return;}

    if(value.isPressed)
    {
        myRigidbody.velocity += new Vector2 (0f, jumpSpeed);
    }
}

```

Obrázek 53 Playermovement metoda OnJump (1)

- ClimbLadder()

Díky této metodě může hráč šplhat po žebříku. Metoda zjistí, zda se hráč dotýká žebříku a pokud ano, tak se nastaví gravitace na nulu, vypne se možnost skoku. Pokud hráč začne šplhat, zapne se animace šplhání.



```

void ClimbLadder()
{
    if (!myFeetCollider.IsTouchingLayers(LayerMask.GetMask("Climbing")))
    {
        myRigidbody.gravityScale = gravityScaleAtStart;
        myAnimator.SetBool("isClimbing", false);
        return;
    }

    Vector2 climbVelocity = new Vector2 (myRigidbody.velocity.x, moveInput.y * climbSpeed);
    myRigidbody.velocity = climbVelocity;
    myRigidbody.gravityScale = 0f;

    bool playerHasVerticalSpeed = Mathf.Abs(myRigidbody.velocity.y) > Mathf.Epsilon;
    myAnimator.SetBool("isClimbing", playerHasVerticalSpeed);
}

```

Obrázek 54 Playermovement metoda ClimbLadder (1)

## SceneControll

- Slouží pro načítání scén, pokud klikneme na nějaké tlačítko.

## ScenePersist

- Objekty, které se zde nachází se po smrti znovu neobjeví v daném levelu. Pokud však hráč ztratí všechny životy znovu se budou v levelu nacházet. Další možnost je ta, že si hráč level zapne sám. V tom případě se objekty také vyresetují a objeví se znovu. Je to ideální funkce pro ukládání mincí a srdíček, aby hráč nemohl víckrát umřít a vždy si vzít srdce nebo mince.

## ScoreManager

- Tento script je definován jakožto ScriptableObject což znamená, že si hodnotu ukládá i skrz scény. Využil jsem ho k tomu abych do něj mohl nahrát skóre a poté jej zobrazit na konečném menu. Pro přístupování k hodnotám v tomto scriptu slouží script

### CoinPickUpManager

## Boots

- Spravuje dotyk s hráčem, objekt se zničí a nastavuje novou rychlost hráče
- Viz obrázek 43

## Clock

- Spravuje dotyk s hráčem, objekt se zničí a přičítá čas

- Viz obrázek 43

### **FallingHazzard**

- **OnTriggerWithFloor()**  
Zjišťuje, jestli se dotkl osten země, pokud jo, tak se zničí.
- **OnTriggerWithPlayer()**  
Zjišťuje, jestli se osten dotkl hráče, pokud jo tak hráč umře.
- **OnTriggerEnter2D()**  
Pokud se hráč dotkne CapsuleCollideru, tak se nastaví gravitace ostnu a ten začne padat.

### **Heart**

- Spravuje dotyk s hráčem, objekt se zničí a přičte srdce
- Viz obrázek 43

### **Key**

- Spravuje dotyk s hráčem, objekt se zničí, odemkne dveře a změní jejich barvu
- Viz obrázek 43

### **QuiverPickUp**

- Spravuje dotyk s hráčem, objekt se zničí, přičte šípy
- Viz obrázek 43

### **Target**

- Spravuje dotyk s šípem, objekt se zničí, odemkne dveře a změní jejich barvu
- Viz obrázek 43

### **PauseMenu**

- **Update()**

Tato metoda zjišťuje, jestli hráč zmáčkl klávesu ESC. Pokud tuto klávesu zmáčkl zavolá se metoda **Pause()**. Při zmáčknutí klávesy ESC v pause menu se zavolá metoda **Resume()**.

- **Pause()**

Tato metoda aktivuje pause menu a nastaví čas na hodnotu 0, tudíž se hra zastaví

```
public void Pause() {
    SetIsPaused(true);
    pauseMenu.SetActive(true);
    Time.timeScale = 0f;
}
```

Obrázek 55 PauseMenu metoda Pause (11)

- **Resume()**

Tato metoda deaktivuje pause menu a nastaví čas na hodnotu 1, tudíž se hra opět spustí

```
public void Resume() {
    SetIsPaused(false);
    pauseMenu.SetActive(false);
    Time.timeScale = 1f;
}
```

Obrázek 56 PauseMenu metoda Resume (11)

- **MainMenu()**

Tato metoda se spustí pokud hráč klikne v pause menu na tlačítko main menu. Čas se zde nastaví na hodnotu 1 a levely se vyresetují.

```
public void MainMenu() {
    Time.timeScale = 1f;
    pauseMenu.SetActive(false);
    FindObjectOfType<GameSession>().ResetGameSession();
    SceneManager.LoadScene(0);
}
```

Obrázek 57 PauseMenu metoda MainMenu

## Timer

- **UpdateTimer()**

Tato metoda řeší počítáč času. Pokud běží čas tak se od nastavené hodnoty času každou sekundu odečte 1. Pokud hodnota času klesne na nulu tak hráč umře. Pokud se čas neodečítá což znamená že hráč umřel, tak se hodnota znovu nastaví na předtím nastavenou hodnotu a proces odečítání nastavá znovu.

```

void UpdateTimer() {
    timerValue -= Time.deltaTime;
    if (isCounting)
    {
        if (timerValue <= 0)
        {
            isCounting = false;
            FindObjectOfType<PlayerMovement>().OtherDie();
        }
    }
    else {
        if (timerValue <= 0)
        {
            isCounting = true;
            timerValue = timeLimit;
        }
    }
    timerText.text = timerValue.ToString("0");
}

```

Obrázek 58 Timer metoda UpdateTimer (12)

- **SetTimer()**

Slouží k přičítání času při sebrání přesýpacích hodin.

```

public void SetTimer(float time) {
    this.timerValue += time;
    timerText.text = timerValue.ToString("0");
}

```

Obrázek 59 Timer metoda SetTimer (12)

## ArrowSO

- Tento script je definován jakožto ScriptableObject což znamená, že si hodnotu ukládá i skrz scény. Využil jsem ho k tomu abych do něj mohl nahrát počet šípů. Pro přístupování k hodnotám v tomto scriptu slouží script **ArrowPickUpManager**

## ArrowPickUpManager

- Pracuje s hodnoty v **ArrowSO**.

- Resetuje šípy, zjišťuje počet šípů a vypisuje počet šípů

```
void Update()
{
    arrowText.text = arrows.GetArrows().ToString();
}

public void resetArrows()
{
    arrows.SetArrows();
}

public void updateArrows() {
    arrows.ArrowShooted();
}

public int GetArrows() {
    return arrows.GetArrows();
}
```

Obrázek 60 ArrowPickUpManager metody

## TESTOVÁNÍ

Název testovaného projektu: Jump Adventure

Jméno autora projektu: Michal Zapletal

Jméno testera Žákyně 4. ročníku SŠ

Výsledná zpráva:

### Kritické funkce:

1. V menu mezi levely lze přejít pouze do hlavního menu nebo do prvního levelu. Jelikož má hráč možnost v úvodním levelu vybrat který level chce hrát uvítala bych na tomto mezi menu možnost alespoň opakovat level, který právě hráč hrál.



Obrázek 61 Testování 1. obrázek

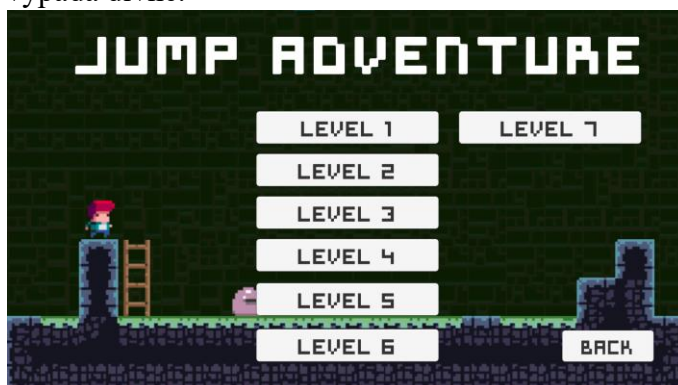
2. V levelu 4 hráč neví, že pro přejítí hnědé země potřebuje lektvar. Je potřeba dát hráči alespoň nápovědu (jak je to např u levelu s terči).
3. Pozdější levely mi přijdou hodně těžké, bylo by vhodné buď přidat u těžších levelů čas, nebo životy.
4. V levelu 5 je dle mého názoru už přehnaně moc terčů, aby hráč hádal, který je ten správný, raději bych jich pár oddělala.
5. V controls bych raději napsala místo „JUMP“ raději „JUMPING“



Obrázek 62 Testování 2. obrázek

**Doporučení:**

1. Tlačítka v levels bych zarovnala do dvou sloupců, aby vypadalo více esteticky. Mít jeden level vedle sloupce vypadá divně.



Obrázek 63 Testování 3. obrázek

2. Přišlo by mi fajn, kdyby dosažené skóre zobrazilo hráči po dokončení každého levelu, a ne pouze po dohrání všech levelů.

**Moje řešení:****Kritické funkce:**

1. Nepřijde mi to nutné. Dá se to však jednoduše přidat.
2. Tento úkol je částí puzzle, které musí hráč vyřešit.
3. Level 6 je dělán tak, aby čas odpovídal co nejmenším chybám, ale přidal jsem 2 sekundy. V levelu 7 jsem přidal dvoje hodiny, které přidávají čas.
4. Tady se opět zastávám myšlenky, že hra má být značně obtížná.
5. Text jsem přepsal

**Doporučení:**

1. Levely jsem rozdělil na dva sloupce.
2. Skóre zde mám, aby hráč viděl, jak dobře hru dokončil.

Název testovaného projektu: Jump Adventure

Jméno autora projektu: Michal Zapletal

Jméno testera Žák 1. ročníku VŠ

Výsledná zpráva:

**Kritické funkce:**

1. Poté co jsem spustil aplikaci a zapnul hru, měl jsem 31 šípů. Když jsem však level restartoval už jsem měl 0 šípů.
2. Když jsem si vybral level, přes tlačítko levels, tak nepřátelé se spawnuli divně ve vzduchu.
3. V levels nefunguje tlačítko back.

Moje řešení:

1. Scriptable object, ve kterém jsou uloženy šípy, jsem nastavil na nulu.
2. Přidal jsem metodu na vyresetování hry.
3. Přidal jsem do něj metodu na vrácení se zpět do menu.



## ZÁVĚR

Prostředí hry JUMP ADVENTURE se mi podařilo vytvořit podle mých představ. Hra obsahuje hlavní menu, kde se je hráč schopen bez problému orientovat a hra obsahuje všechny funkční prvky které jsem si stanovil. Z hlavního menu je schopen se uživatel dostat do ovládání hry, do samotné hry a taky je schopen vybírat si konkrétní level, nebo vypnout hru.

Co se týče hry, podařilo se mi také splnit všechny funkce, které jsem si před vytvořením projektu stanovil. Uživatel tedy může plně ovládat svou postavu, nepřátelé se pohybují z jedné části platformy na druhou část platformy. Dále zde fungují všechny interaktivní textury jako je žebřík, odrazení, hroty. Uživatel si tedy může hru bez větších bugů v klidu zahrát.

Do budoucna by se hra dala vylepšit o následující funkce:

- Přidat nové nepřátele a možnosti, jak umřít.
- Další interaktivní prvky, jako například nějaká krabice, která by se musela posunout, aby mohl hráč dále pokračovat.
- Přidání dalších levelů a kritérií v nich
- Přidání zvukových efektů

## SEZNAM POUŽITÉ LITERATURY

- (1) Complete C# Unity Game Developer 2D Online Course - 2021. *GameDev.tv* [online]. Nottingham: Embrace.IT Limited, 2021 [cit. 2022-10-3]. Dostupné z: <https://www.gamedev.tv/courses/enrolled/1394720>
- (2) Complete C# Unity Game Developer 2D Online Course (Unity 2018). *GameDev.tv* [online]. Nottingham: Embrace.IT Limited, 2021 [cit. 2022-10-27]. Dostupné z: <https://www.gamedev.tv/courses/enrolled/666005>
- (3) Super Platformer Assets. *Unity Asset Store* [online]. Sly Cooper: Sanzaru Games [cit. 2022-09-22]. Dostupné z: <https://assetstore.unity.com/packages/2d/environments/super-platformer-assets-42013> Licence: Standard Unity Asset Store EULA
- (4) 2D Movement in Unity (Tutorial). *YouTube* [online]. Brackeys, 2018 [cit. 2022-10-3]. Dostupné z: <https://www.youtube.com/watch?v=dwcT-Dch0bA>
- (5) 2D Animation in Unity (Tutorial). *YouTube* [online]. Brackeys, 2018 [cit. 2022-10-29]. Dostupné z: <https://www.youtube.com/watch?v=hkaysu1Z-N8>
- (6) Collision.collider. *UNITY Documentation* [online]. 2022 [cit. 2023-02-12]. Dostupné z: <https://docs.unity3d.com/ScriptReference/Collision-collider.html>
- (7) Collider.2D.IsTouching *UNITY Documentation* [online]. 2022 [cit. 2023-03-10]. Dostupné z: <https://docs.unity3d.com/ScriptReference/Collision-collider.html>
- (8) PIXEL ART in Photoshop (Tutorial). *YouTube* [online]. 22.11.2017 [cit. 2022-10-3]. Dostupné z: <https://www.youtube.com/watch?v=rLdA4Amea7Y>
- (9) How To Pixel Art In 10 Minutes | Pixel Art Tutorial. *YouTube* [online]. 1.5.2020 [cit. 2022-10-3]. Dostupné z: <https://www.youtube.com/watch?v=v-ibXM3xBjg>
- (10) Check to see if GameObject exists. *Unity* [online]. 8.11.2015 [cit. 2022-11-10]. Dostupné z: <https://forum.unity.com/threads/check-to-see-if-gameobject-exists.366643/>
- (11) How To Make A PAUSE MENU In 4 Minutes - Easy Unity Tutorial. *YouTube* [online]. 12.3.2021 [cit.2023-01-21]. Dostupné z: [https://www.youtube.com/watch?v=tfzwyNS1LUY&t=3s&ab\\_channel=Hooson](https://www.youtube.com/watch?v=tfzwyNS1LUY&t=3s&ab_channel=Hooson)
- (12) Countdown Timer In Unity - Easy Beginners Tutorial/ Guide. *YouTube* [online]. 12.3.2021 [cit. 2022-12-15]. Dostupné z: [https://www.youtube.com/watch?v=o0j7PdU88a4&t=2s&ab\\_channel=SingleSaplingGames](https://www.youtube.com/watch?v=o0j7PdU88a4&t=2s&ab_channel=SingleSaplingGames)

## SEZNAM OBRÁZKŮ

Obrázek 1 Tlačítko start.....	8
Obrázek 2 Tlačítko start s najetým kurzorem .....	9
Obrázek 3 Hlavní menu hry .....	9
Obrázek 4 Controls .....	10
Obrázek 5 7. level .....	11
Obrázek 6 Pause Menu .....	11
Obrázek 7 Levels .....	12
Obrázek 8 Endgame menu .....	12
Obrázek 9 Wireframe hlavního menu .....	14
Obrázek 10 Wireframe hry .....	15
Obrázek 11 Přechod kamer .....	20
Obrázek 12 Čas před přechodem kamer .....	21
Obrázek 13 Kamery k dějům .....	21
Obrázek 14 Ohraničení mapy pro kameru .....	21
Obrázek 15 Uživatel vs programátor .....	22
Obrázek 16 Luk.....	22
Obrázek 17 Textura srdce .....	23
Obrázek 18 Textura elixíru .....	23
Obrázek 19 Textura boty.....	24
Obrázek 20 Textura přesýpacích hodin .....	24
Obrázek 21 Textura terče .....	25
Obrázek 22 Textura padajících ostnů.....	25
Obrázek 23 Textura klíče .....	26
Obrázek 24 Textura toulce .....	26
Obrázek 25 Textura dveří .....	27
Obrázek 26 Textury platform .....	28
Obrázek 27 Rule Tile .....	28
Obrázek 28 Textury pozadí.....	29
Obrázek 29 Textury vody .....	30
Obrázek 30 Textura lávy.....	30
Obrázek 31 Textura žebříku.....	30

Obrázek 32 Textura houby.....	31
Obrázek 33 Textura hrotů .....	31
Obrázek 34 Textura postavičky .....	32
Obrázek 35 Textura nepřítele.....	33
Obrázek 36 Animace nepřítele.....	33
Obrázek 37 Textura mince .....	34
Obrázek 38 Animace mince .....	34
Obrázek 39 BowPickup metoda OnTriggerEnter2D .....	35
Obrázek 40 Arrow metoda OnTriggerEnter2D .....	35
Obrázek 41 Arrow metoda OnCollisionEnter2D.....	35
Obrázek 42 Arrow metoda flipArrow .....	36
Obrázek 43 CoinPickup metoda OnTriggerEnter2D .....	36
Obrázek 44 EnemyMovement metoda OnTriggerEnter2D .....	37
Obrázek 45 Enemy BoxCollider2D .....	37
Obrázek 46 EnemyMovement metoda FlipEnemyFacing (2) .....	37
Obrázek 47 GameSession metoda Process PlayerDeath.....	38
Obrázek 48 GameSession metoda TakeLife .....	38
Obrázek 49 LevelExit metoda LoadNextLevel (1).....	38
Obrázek 50 LevelExit metoda OnTriggerEnter2D .....	39
Obrázek 51 PlayerMovement metoda OnFire (1).....	39
Obrázek 52 PlayerMovement metoda GetPlayerRotation (1) .....	40
Obrázek 53 Playermovement metoda OnJump (1) .....	40
Obrázek 54 Playermovement metoda ClimbLadder (1) .....	41
Obrázek 55 PauseMenu metoda Pause (11).....	43
Obrázek 56 PauseMenu metoda Resume (11) .....	43
Obrázek 57 PauseMenu metoda MainMenu .....	43
Obrázek 58 Timer metoda UpdateTimer (12) 24.11.2022.....	44
Obrázek 59 Timer metoda SetTimer (12) 24.11.2022 .....	44
Obrázek 60 ArrowPickUpManager metody .....	45
Obrázek 61 Testování 1. obrázek.....	46
Obrázek 62 Testování 2. obrázek.....	46
Obrázek 63 Testování 3. obrázek.....	47

