

Dokumentace klauzurní práce

Rhythm Revolutions

[Odkaz](#)

**STŘEDNÍ ŠKOLA FILMOVÁ, MULTIMEDIÁLNÍ
A POČÍTAČOVÝCH TECHNOLOGIÍ, s.r.o.**

Vypracoval: Hanuš Valenta 2.C 2025

1. Úvod	3
Co je v dokumentu uvedeno	3
2. Popis použitých technologií.....	4
3.1 Unity	4
3.2 InstallForge	4
3.3 C#.....	4
3. Použitý editor	6
4.1 VSCodium	6
4.2 Výhody VSCodium	6
4. Popis adresářů	7
5.1 Assets/Animations.....	7
5.2 Assets/BossFightPatterns	7
5.3 Assets/Fonts	7
5.4 Assets/MaterialsAndShaders	7
5.5 Assets/SongScripts	7
5.6 Assets/Prefabs	7
5.7 Assets/Scenes.....	7
5.8 Assets/Scripts	7
5.9 Assets/Songs.....	7
5.10 Assets/SoundEffects	7
5.11 Assets/Sprites.....	7
5. Popis klíčových scriptů	9
6.1 Player.cs	9
6.2 PlayerHeartControler.cs.....	9
6.3 GameManager.cs.....	10
6.4 BossFightSpawnPattern.cs a SpawnPattern.cs	10
6.5 Fade.cs	10
6.6 BossFight.cs	11
6.7 MenuManager.cs	11
6.8 .gitignore	11
7. Zdroje.....	12

1. Úvod

Hra inspirovaná filmem Scott Pilgrim vs. The World využívá stylizovanou 2D grafiku, akční boss fight systém a několik logických hádanek které věrně navazují na filmovou předlohu i komiksový styl. Projekt vznikl jako školní klauzurní práce s cílem vytvořit zábavnou hru s zajímavým příběhem a obtížnými boss fighty.

Pro vývoj jsem zvolil Unity Game Engine (dále referencován jako Unity), protože nabízí hotové nástroje pro 2D hry, podporuje C# a umožňuje rychlé prototypování. Oproti jiným enginům nebo vlastnímu řešení je jednodušší na použití a lépe se hodí pro školní projekt.

Co je v dokumentu uvedeno

Dokumentace popisuje vývoj hry v prostředí Unity, použití technologií, organizaci projektu a logiku jednotlivých částí hry. Zaměřuje se i na prezentaci herních mechanismů, jako je pohyb, útok, sběr předmětů a souboje s bossy. V závěru jsou uvedeny zdroje použité při vývoji.

2. Popis použitých technologií

3.1 Unity

Unity je výkonný herní engine určený pro vývoj 2D i 3D her. Umožňuje vizuální návrh herního prostředí a skriptování herní logiky pomocí jazyka C#. Nabízí širokou podporu pro různé platformy (Windows, Android, WebGL aj.) a obsahuje nástroje pro animace, fyziku, ovládání postav, práci se scénami i zvukem. Unity je oblíbené pro svou přehlednost, flexibilitu a rozsáhlou komunitu. [1]

3.2 InstallForge

InstallForge je program určený pro jednoduché a efektivní vytváření instalačních balíčků pro software. Umožňuje snadno sestavit instalační soubor, který uživateli umožní rychlou a bezproblémovou instalaci aplikace na jeho počítači. InstallForge nabízí přehledné uživatelské rozhraní, podporuje různé verze Windows a umožňuje přizpůsobit instalační proces podle potřeb projektu.

Pro svůj projekt jsem zvolil právě InstallForge díky jeho jednoduchosti, spolehlivosti a širokým možnostem konfigurace. Program umožňuje rychle vytvořit profesionální instalační balíček bez nutnosti psát komplikované skripty nebo investovat čas do složitého nastavování. Díky tomu mohu zajistit, že finální produkt bude pro uživatele snadno dostupný a instalace proběhne bez problémů.

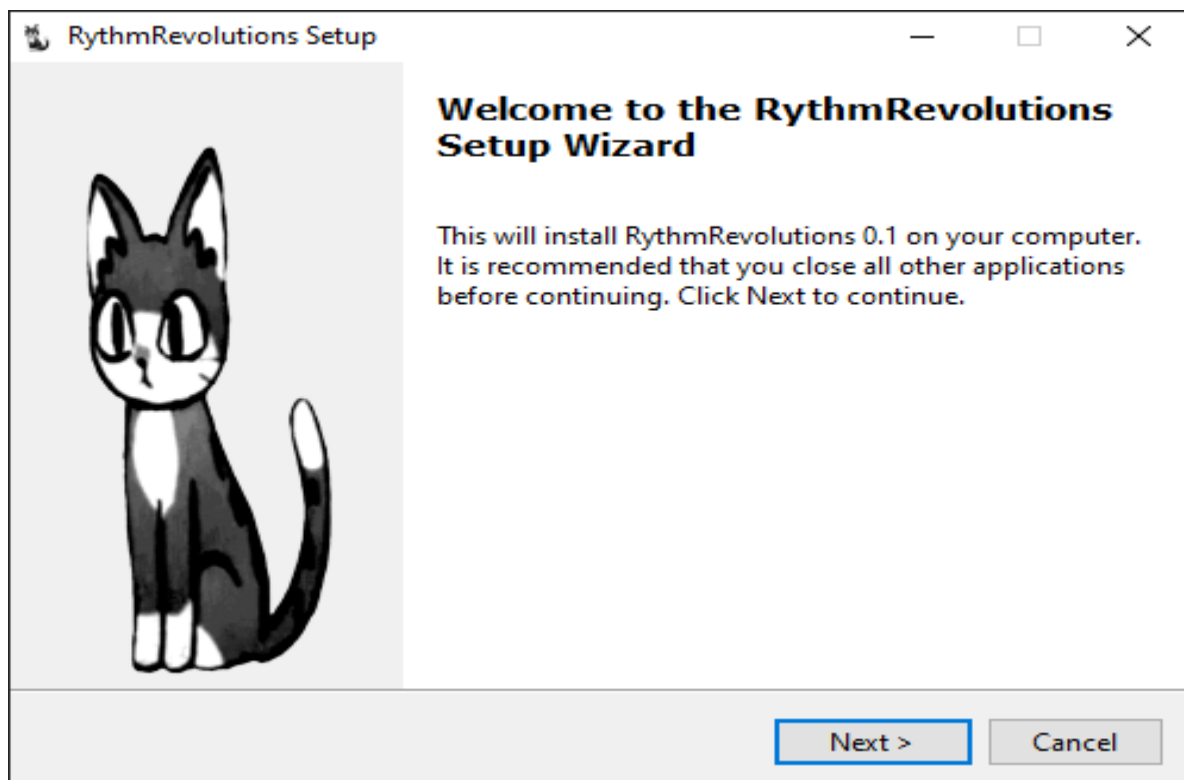
Použití InstallForge významně zjednodušuje distribuci mé hry a pomáhá minimalizovat možné technické potíže při instalaci, což přispívá k lepší uživatelské zkušenosti a profesionálnějšímu dojmu z celého projektu. [2]

3.3 C#

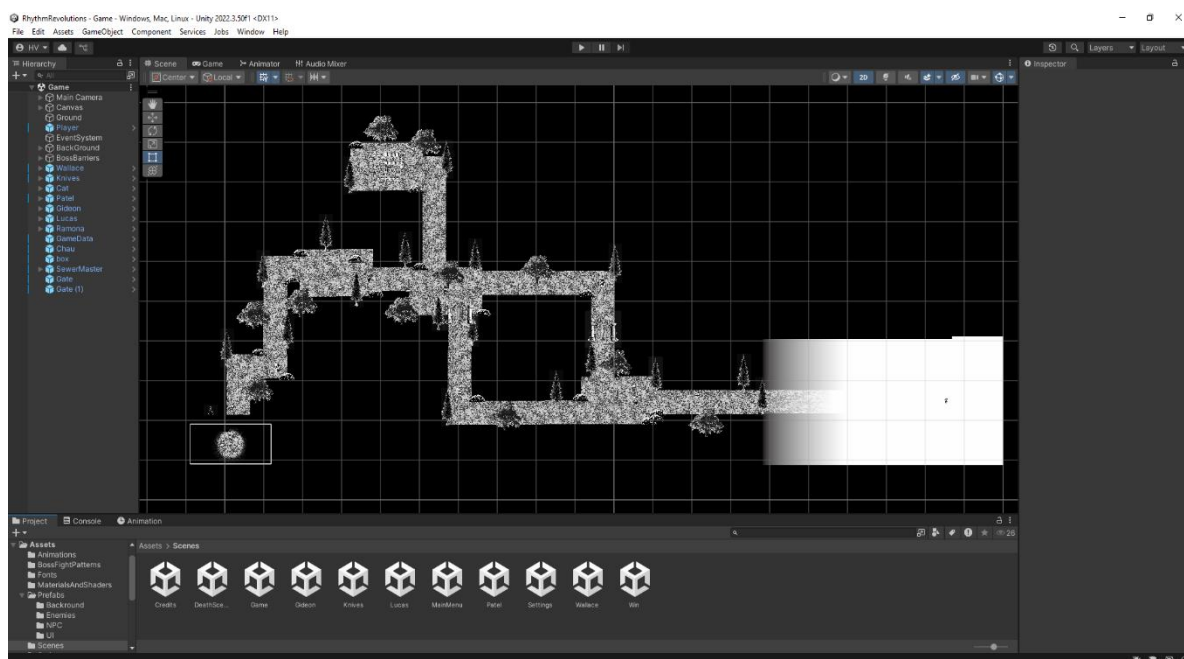
C# je moderní, objektově orientovaný programovací jazyk vyvinutý společností Microsoft v roce 2000 jako součást platformy .NET a využíván v Unity.

Je silně inspirován jazykem Java, a proto sdílí podobnou syntaxi a strukturu, což usnadňuje přechod mezi těmito jazyky. C# je navržen pro jednoduchost, bezpečnost a vysoký výkon při vývoji aplikací, a to zejména desktopových, webových a herních.

Díky své silné integraci s Unity je C# velmi populární jazyk pro vývoj 2D a 3D her.



1



2

¹ Ukázka instalačního okna z instalačního balíčku vytvořeného pomocí InstallForge.

² Ukázka otevřeného projektu v Unity.

3. Použitý editor

4.1 VSCodium

VSCodium je open-source distribuce Visual Studio Code, která je poskytována bez integrovaných služeb Microsoftu, jako je například Telemetry. To znamená, že VSCodium je úplně stejný jako Visual Studio Code, ale neobsahuje žádný kód, který by sledoval nebo sbíral informace o uživatelském chování. [7]

4.2 Výhody VSCodium

VSCodium nabízí širokou škálu funkcí pro vývojáře, včetně podpory pro mnoho programovacích jazyků, integrovaných nástrojů pro ladění a správu verzí a rozsáhlého ekosystému rozšíření. VSCodium je ideální volbou pro ty, kteří hledají výkonný a flexibilní textový editor, který respektuje jejich soukromí a bezpečnost.

4. Popis adresářů

5.1 Assets/Animations

Zde se nachází předvytvořené animace vytvořené pomocí keyframes v unity.

5.2 Assets/BossFightPatterns

V několika souborech se zde nachází instrukce kde a jaké „nepřátele“ umístit do hry při druhé fázi boje z nepřátely.

5.3 Assets/Fonts

Všechny typy písma použité v projektu.

5.4 Assets/MaterialsAndShaders

Zde najdete kolekci shaderů pro materiály aplikované na několik objektů ve hře.

5.5 Assets/SongScripts

V několika souborech se zde nachází instrukce jaké noty umístit do hry při první fázi boje z nepřátely.

5.6 Assets/Prefabs

Zde se nachází soubory s předpřipravenými objekty které se nachází v hlavní „game“ scéně nebo jsou umístěny až později dedikovanými scripty.

5.7 Assets/Scenes

Zde se nachází Unity scény tvořící hru.

5.8 Assets/Scripts

Zde se několika adresářích nachází scripty aplikované na objekty ve hře zaručující jejich funkcionalitu a interaktivitu.

5.9 Assets/Songs

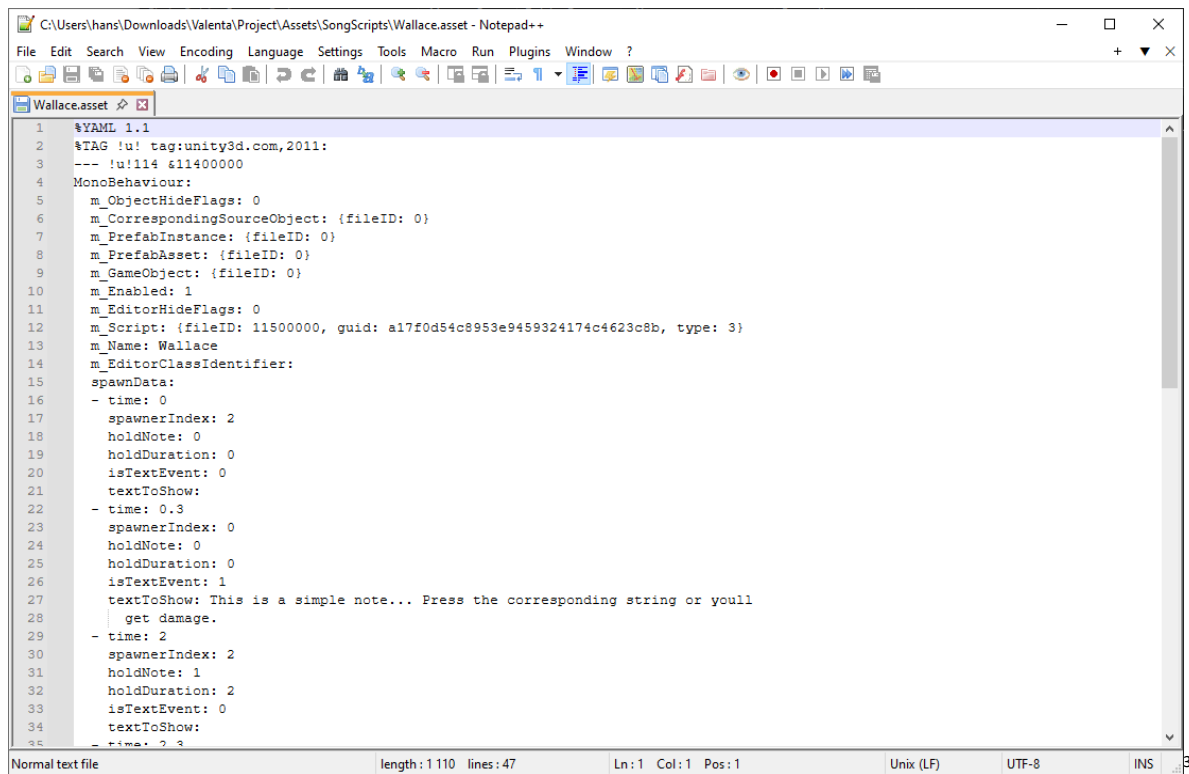
Zde se nachází skladby využité jako hudební podklad pro hru a její scény.

5.10 Assets/SoundEffects

Zde se nachází zvukové efekty využité pro zvukové podbarvení hry a zejména interakcí v ní.

5.11 Assets/Sprites

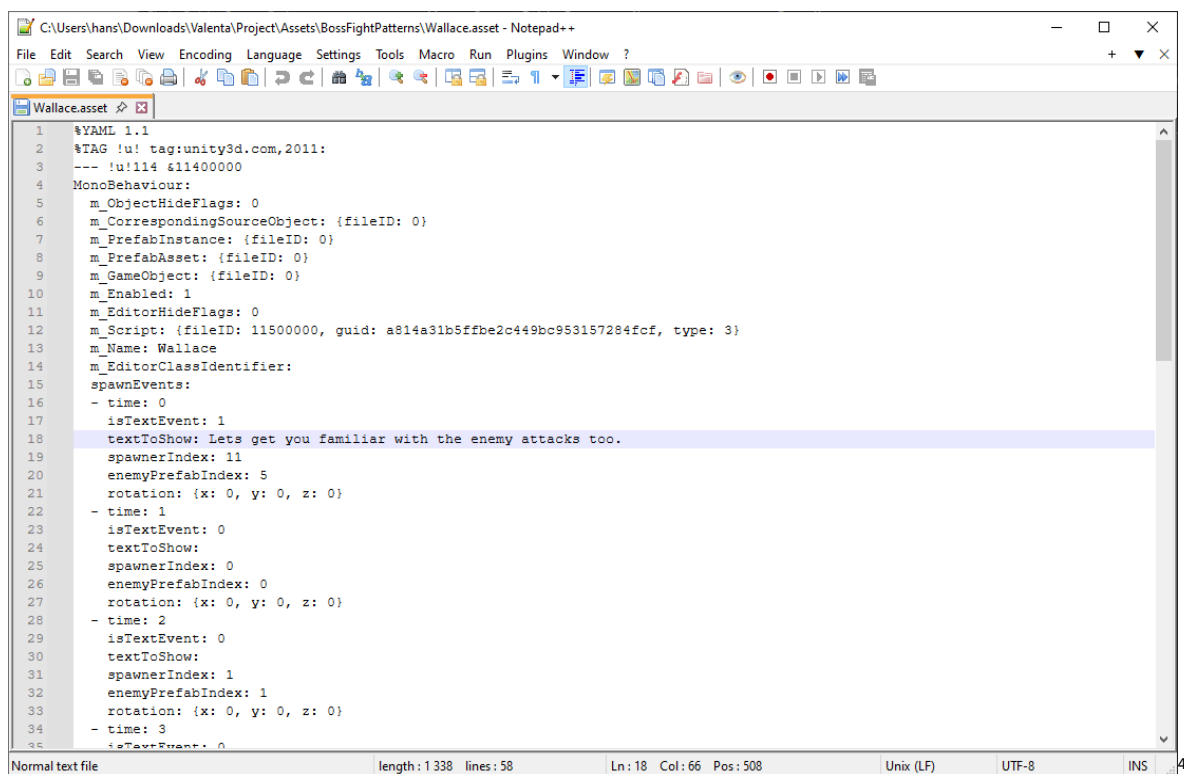
Zde se nachází vizuální podklady pro objekty, uživatelské rozhraní a scény



```
1 %YAML 1.1
2 %TAG !u! tag:unity3d.com,2011:
3 --- !u!114 611400000
4 MonoBehaviour:
5   m_ObjectHideFlags: 0
6   m_CorrespondingSourceObject: {fileID: 0}
7   m_PrefabInstance: {fileID: 0}
8   m_PrefabAsset: {fileID: 0}
9   m_GameObject: {fileID: 0}
10  m_Enabled: 1
11  m_EditorHideFlags: 0
12  m_Script: {fileID: 11500000, guid: a17f0d54c8953e9459324174c4623c8b, type: 3}
13  m_Name: Wallace
14  m_EditorClassIdentifier:
15  spawnData:
16    - time: 0
17      spawnerIndex: 2
18      holdNote: 0
19      holdDuration: 0
20      isTextEvent: 0
21      textToShow:
22    - time: 0.3
23      spawnerIndex: 0
24      holdNote: 0
25      holdDuration: 0
26      isTextEvent: 1
27      textToShow: This is a simple note... Press the corresponding string or you'll
28      get damage.
29    - time: 2
30      spawnerIndex: 2
31      holdNote: 1
32      holdDuration: 2
33      isTextEvent: 0
34      textToShow:
35    - time: 2.3
```

Normal text file length: 1110 lines: 47 Ln: 1 Col: 1 Pos: 1 Unix (LF) UTF-8 INS

3



```
1 %YAML 1.1
2 %TAG !u! tag:unity3d.com,2011:
3 --- !u!114 611400000
4 MonoBehaviour:
5   m_ObjectHideFlags: 0
6   m_CorrespondingSourceObject: {fileID: 0}
7   m_PrefabInstance: {fileID: 0}
8   m_PrefabAsset: {fileID: 0}
9   m_GameObject: {fileID: 0}
10  m_Enabled: 1
11  m_EditorHideFlags: 0
12  m_Script: {fileID: 11500000, guid: a814a31b5ffbe2c449bc953157284fcf, type: 3}
13  m_Name: Wallace
14  m_EditorClassIdentifier:
15  spawnEvents:
16    - time: 0
17      isTextEvent: 1
18      textToShow: Lets get you familiar with the enemy attacks too.
19      spawnerIndex: 11
20      enemyPrefabIndex: 5
21      rotation: {x: 0, y: 0, z: 0}
22    - time: 1
23      isTextEvent: 0
24      textToShow:
25      spawnerIndex: 0
26      enemyPrefabIndex: 0
27      rotation: {x: 0, y: 0, z: 0}
28    - time: 2
29      isTextEvent: 0
30      textToShow:
31      spawnerIndex: 1
32      enemyPrefabIndex: 1
33      rotation: {x: 0, y: 0, z: 0}
34    - time: 3
35      isTextEvent: 0
```

Normal text file length: 1338 lines: 58 Ln: 18 Col: 66 Pos: 508 Unix (LF) UTF-8 INS

4

³ Ukázka souboru udávajícího pořadí a typ not v první fázi boje s nepřáteli vytvořeného pomocí souboru Assets/SongScripts.

⁴ Ukázka souboru udávajícího pořadí a typ nepřátel v druhé fázi boje s nepřáteli vytvořeného pomocí souboru Assets/BossFightPatterns.

5. Popis klíčových scriptů

6.1 Player.cs

Tento skript ovládá hráče v game scéně. Hráč se pohybuje pomocí Rigidbody2D s plynulým zrychlením a zpomalením. Ovládání je pomocí tlačítek WASD na klávesnici a pohyb je normalizovaný, aby rychlost byla stejná ve všech směrech. Kamera sleduje pozici hráče plynule pomocí interpolace. Hráč má animaci složenou ze tří snímků pro každý směr pohybu a idle snímek ukazující se pokud se hráč nepohybuje, animace se přepíná podle směru pohybu. Skript kontroluje kolize a trigger zóny, kde hráč může interagovat s objekty jako např. „box“ objekt využívaný při setkání s Chao NPC. Pokud hráč narazí na bariéru, je přesunut na nejbližší respawn point nastavený v editoru. Také zajišťuje aby bubliny interakcí se zobrazovaly, pokud je hráč v zóně interakcí s NPC tyto bubliny využívající. Při stisku klávesy E u specifických objektů jako NPC se zobrazují dialogy s nimi a případně se může spustit přechod do jiné scény, například většinou boje s daným NPC nebo příběhové jako např. u Ramona NPC. Při startu se hráčova pozice nastaví podle uložených dat, aby mohl pokračovat tam, kde skončil nebo u nejbližšího checkpointu pokud umřel na dotknutí s Spike objektem. Skript také přehrává opakující se hudbu s nastavitelnou hlasitostí.

6.2 PlayerHeartControler.cs

Skript umožňuje hráči pohybovat se pomocí klávesnice v omezeném prostoru při druhé fázi bossfightu. Rychlost pohybu je určena proměnnou moveSpeed a pohyb je normalizován. Pozice hráče je omezena na definované hranice ve scéně pomocí funkce Mathf.Clamp.

○ ○ ○

```
void Update()
{
    float horizontalInput = Input.GetAxisRaw("Horizontal");
    float verticalInput = Input.GetAxisRaw("Vertical");

    Vector2 movement = new Vector2(horizontalInput, verticalInput);
    movement = movement.normalized;

    rb.velocity = movement * moveSpeed;

    transform.position = new Vector3(
        Mathf.Clamp(transform.position.x, minX, maxX),
        Mathf.Clamp(transform.position.y, minY, maxY),
        transform.position.z
    );
}
```

5

⁵ Ukázka souboru PlayerHeartControler.cs - Funkce zajišťující pohyb hráče podle vstupu z klávesnice a omezuje jeho pozici na určené hranice „fightSquare“ ve scéně.

6.3 GameManager.cs

Skript GameManager slouží jako centrální správce stavu hry. Využívá tzv. singleton pattern, což znamená, že ve hře existuje jen jedna jeho instance, která se uchová i při přechodu mezi scénami. Uchovává důležité informace o průběhu hry, jako například kolik má hráč životů při startu boss fightu, zda porazil nebo ušetřil jednotlivé NPC postavy, zda interagoval s Cat NPC, nebo například jestli se má přeskočit úvod hry pro zrychlení playtestinu při vývoji. Skriptem se také ukládá pozice hráče při opuštění scény, případně pozice, kam se má hráč znovu umístit při načtení. Dále obsahuje speciální proměnné, které umožňují přeskakovat některé části hry (například souboje nebo úvodní sekvenci) také pro zrychlení playtestingu. Funkce FoughtBoss a SparedBoss automaticky nastavují příslušné hodnoty podle názvu právě aktivní scény. Skript slouží jako centrální bod pro řízení herního průběhu a rozhodnutí hráče, aby tyto rozhodnutí byly využity v konci hry a dali příslušný konec podle nic.

6.4 BossFightSpawnPattern.cs a SpawnPattern.cs

Třídy SpawnData a EnemySpawnEvent uchovávají informace o událostech spawnování ve hře, včetně času, spawneru a dalších parametrů. SpawnData obsahuje údaje o držení noty a textových událostech s možností zobrazit delší text. EnemySpawnEvent navíc specifikuje index prefab nepřítele a jeho rotaci, a také podporuje textové události. Obě třídy slouží k vytváření souborů na spawnování a spouštění událostí během hry.

6.5 Fade.cs

Třída AudioFader poskytuje korutiny pro plynulé zeslabení a zesílení zvuku u AudioSource během zadaného času. Třída Fade také umožňuje přechody mezi scénami s vizuálním efektem blednutí pomocí animátoru a zároveň plynulé zeslabení zvuku během přechodu. Při přechodu si také ukládá pozici hráče a aktuální scénu, aby mohl být hráč případně respawnován na správném místě. Metody umožňují spustit fade in/out zvuku na konkrétním AudioSource s nastavitelnou délkou trvání.

○ ○ ○

```
public static IEnumerator FadeIn(AudioSource audioSource, float targetVolume, float fadeTime)
{
    if (audioSource == null) yield break;
    audioSource.volume = 0;
    if (!audioSource.isPlaying)
    {
        if(audioSource.clip != null) audioSource.Play();
    }
    if (fadeTime <= 0) fadeTime = 0.001f;
    targetVolume = Mathf.Clamp01(targetVolume);
    while (audioSource.volume < targetVolume - 0.001f)
    {
        audioSource.volume += targetVolume * Time.unscaledDeltaTime / fadeTime;
        yield return null;
    }
    audioSource.volume = targetVolume;
}
```

6

⁶ Ukázka funkce z souboru Fade.cs - Postupně zvyšuje hlasitost zvuku na cílovou úroveň v daném čase a spustí přehrávání, pokud ještě nehraje.

6.6 BossFight.cs

Skript řídí průběh boss fightu, včetně spawnování nepřátel a zobrazování textových událostí podle časového plánu. Při začátku boje se nastaví hudba a inicializují se proměnné potřebné pro sledování průběhu. V Update metodě se postupně zpracovávají umístovací události z daného souboru udávající průběh umístování a po skončení boje nebo slitováním nad ním se spustí závěrečná sekvence s efektem otřesu jeho spritu a přechodem zpět do game scény.

○ ○ ○

```
void ProcessSpawnEvent(EnemySpawnEvent spawnEvent)
{
    if (spawnEvent.isTextEvent)
    {
        if (textBox != null)
        {
            textBox.ShowText(spawnEvent.textToShow);
        }
    }
    else
    {
        if (spawnEvent.spawnerIndex >= 0 && spawnEvent.spawnerIndex <
enemySpawners.Count)
        {
            if (enemySpawners[spawnEvent.spawnerIndex] != null)
            {
                enemySpawners[spawnEvent.spawnerIndex].SpawnEnemy(spawnEvent.enemyPrefabIndex, Quaternion.Euler(spawnEvent.rotation));
            }
        }
    }
}
```

7

6.7 MenuManager.cs

Skript spravuje menu ve hře a několika jejích scénách, ovládá zobrazení tlačítek pro boj, slitováním nad bossem a start hry. Zajišťuje přechod do druhé fáze bossfightu s efektem postupného zobrazení „fightSquare“ který definuje hranici kam se hráč může pohybovat při ní a spawnováním „PlayerHeart“ představeného objektu sekundu po plném zobrazení „fightQuare“. Obsahuje také funkce pro načtení hry, ukončení aplikace a otevírání odkazů na Discord a Itch.io využívané v úvodní menu scéně.

6.8 .gitignore

Soubor .gitignore určuje, které soubory a složky Git nebude sledovat nebo přidávat do verzovacího systému. Používá se k ignorování dočasných, generovaných nebo citlivých souborů.[4]

⁷ Ukázka funkce z souboru BossFight.cs - Zpracuje událost a buď zobrazí text, nebo zavolá umístění nepřítele z určeného spawneru s danou rotací.

7. Zdroje

[1] UNITY TECHNOLOGIES. Unity. Online. 2025, 24.05.2025. Dostupné z: <https://unity.com/>. [cit. 2025-05-24].

[2] SOLICUS. InstallForge. Online. 2007, 24.05.2025. Dostupné z: <https://docs.installforge.net/>. [cit. 2025-05-24].

[3] TIM O'BRIEN. VSCodium. Online. 2019, 24.05.2025. Dostupné z: <https://vscodium.com/#why>. [cit. 2025-05-24].

[4] SOFTWARE FREEDOM CONSERVANCY, INC. Git-scm. Online. 2025. Dostupné z: <https://git-scm.com/docs/gitignore>. [cit. 2025-05-24].