

**Gymnázium Christiana Dopplera, Zborovská 45, Praha  
5**

**Ročníková práce**

**2D Unity hra ve stylu Plants vs.  
Zombies**

Vypracovala: Tereza Píšková

Třída: 4.C

Školní rok: 2023/2024

Seminář: Programování

Prohlašuji, že jsem svou ročníkovou práci napsala samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 1.1.2024

Tereza Písková

Ráda bych poděkovala profesoru Bc. Petru Vincenovi za odborné vedení práce, mým rodičům za místné a expertní poznámky, mladší sestře za grafický návrh pro ořech a Honzovi Libicherovi za zasvěcení do tajů hudby TempleOS. Díky samozřejmě patří i obyvatelům stackoverflow.com a forum.unity.com, jejichž síla přátelství dokáže překonat jakékoli errory na neexistujících řádcích.

# **Obsah**

<b>1</b>	<b>Úvod</b>	<b>6</b>
<b>2</b>	<b>Použité programy</b>	<b>8</b>
2.1	Unity . . . . .	8
2.1.1	Scény a objekty . . . . .	8
2.1.2	Prefabs . . . . .	8
2.1.3	Programování . . . . .	9
2.1.4	User Interface . . . . .	10
2.2	Visual Studio 2022 . . . . .	10
2.3	Procreate . . . . .	12
<b>3</b>	<b>Hra</b>	<b>13</b>
3.1	Scény . . . . .	13
3.1.1	Hlavní menu . . . . .	13
3.1.2	Zahrada . . . . .	14
3.2	Herní objekty . . . . .	14
3.2.1	Rostliny a munice . . . . .	16
3.2.2	Duchové . . . . .	17
3.2.3	Slunce . . . . .	18
3.3	Kupování rostlin . . . . .	19
3.4	Práce s daty . . . . .	20
3.4.1	Level data . . . . .	20
3.4.2	PlayerPrefs . . . . .	21
3.4.3	Sloty . . . . .	24
3.5	Game Over . . . . .	24

3.6	Testy . . . . .	25
3.7	Hudba . . . . .	27
3.8	Animace . . . . .	28
3.9	Problémy . . . . .	29
3.9.1	Velikost spritů . . . . .	29
3.9.2	Nekonzistentní pořadí skriptů . . . . .	29
3.9.3	Grafika . . . . .	30
3.10	Rozdíly v původní hře . . . . .	30
3.10.1	Kvantitativní část . . . . .	31
3.10.2	Grafická část . . . . .	32
3.10.3	Verze zahrady . . . . .	32
3.10.4	Minihry a Obchod . . . . .	33
<b>4</b>	<b>Závěr</b>	<b>34</b>

# 1 Úvod

V mojí seminární práci bych ráda představila svoji 2D hru, kterou jsem vytvořila v prostředí Unity. Inspirovala jsem se velmi populární hrou Plants vs. Zombies, dále PvZ. Jedná se o strategickou 2D hru pro jednoho hráče vyvíjenou PopCap Games a poprvé zveřejněna 5.5.2009. Úkolem hráče je ubránit dům před zombies tím, že sází do herního pole, velikosti 5x10, rostliny, které na ně útočí. Květiny si kupuje za sluníčka, která padají z nebe, nebo z některých rostlin, jako například slunečnice.

V každém levelu hráč najde jednu kartu nové rostliny, která se přidá do jeho itineráře, a na začátku každého kola si může vybrat až 7 rostlin. Zkušení hráči si vybírají rostliny podle toho, jací zombies se na zahradu chystají. Pokud jede zombie v rolbě na led, je nejstrategičtější si zvolit takovou rostlinu, která prořízne pneumatiky. [8]

Hra se mi zdála dostatečně jednoduchá, aby její vytvoření bylo jednoduché i pro někoho, kdo se s Unity dosud nesetkal. V zápětí jsem zjistila, že toto nebyl správný předpoklad.

Při vývoji hry jsem využívala Unity jako hlavní herní engine. Bylo to pro mě zcela nové prostředí, proto jsem velkou část přípravy této práce věnovala výukovým programům na oficiálních stránkách, skrz které jsem se s Unity seznamovala. Dále jsem využila Procreate k vytváření veškeré grafiky, tedy pozadí, menu, i sprity. Veškerá práce v Procreate mi zabrala okolo 30 hodin. Procreate pro mě také bylo nové, ale díky předchozí zkušenosti s grafickými editory byla práce vcelku intuitivní.

V seminární práci představuji všechny programy, které jsem využívala, a jakým způsobem jsem vytvářela hru. Věnuji se i práci s externími soubory. Popisují problémy, na které jsem během vyvíjení narazila, a kam by práce

mohla potenciálně směřovat.

Všechn zmiňovaný kód je dostupný zde:

<https://github.com/terpis/Plants-vs-Zombies>

A hra samotná je ke stažení zde:

[https://drive.google.com/file/d/1fxXKd19RpHYUzj7S9T8yM2W5S\\_aqcEym/view?usp=drive\\_link](https://drive.google.com/file/d/1fxXKd19RpHYUzj7S9T8yM2W5S_aqcEym/view?usp=drive_link)

## 2 Použité programy

K vývoji jsem využila Unity, Microsoft Visual Studio 2022 a Procreate.

### 2.1 Unity

Unity je multiplatformní herní engine vyvíjený společností Unity Technologies. Je široce využíván pro tvorbu 2D i 3D her, simulací a interaktivních zážitků. Unity podporuje různé platformy, včetně stolních počítačů, mobilních zařízení, konzolí a zařízení pro virtuální realitu (VR). [7]

#### 2.1.1 Scény a objekty

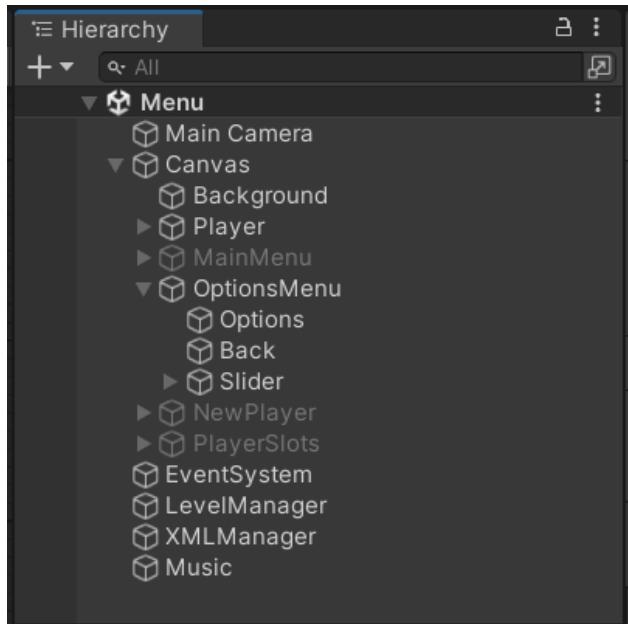
Hra vytvořená v Unity se skládá ze scén, ve kterých jsou herní objekty. Každá scéna má vlastní hierarchii (obr. č.1), kam se objekty vkládají. Objekty se bud' vybírají z předpřipravených, nebo se vytvoří prázdné. V inspectoru se ke každému objektu můžou přidat komponenty, jejichž výběr je omezen minimálně. Právě komponenty udělají z prázdného objektu herní figurku, tlačítko, či manager hudby.

Například v inspectoru slunečnice (obr. č.2) má objekt Transform, který si drží souřadnice, rotaci a velikost objektu, Sprite Renderer, který obstarává obrázek objektu, Box Collider a Rigidbody, které mají na starost fyziku a kolize, některé skripty a Animator s Animation, které zařizují animaci.

Každý objekt pak má nastavitelný tag a vrstvu, které slouží k lepší orientaci a přehlednosti při strukturování skriptů.

#### 2.1.2 Prefabs

Prefabs, neboli prefabrikáty, jsou herní objekty, které slouží k opakovanému vytvoření duplikátů. V prefabu si můžeme ukládat téměř cokoliv, až na od-



Obrázek 1: Hierarchie

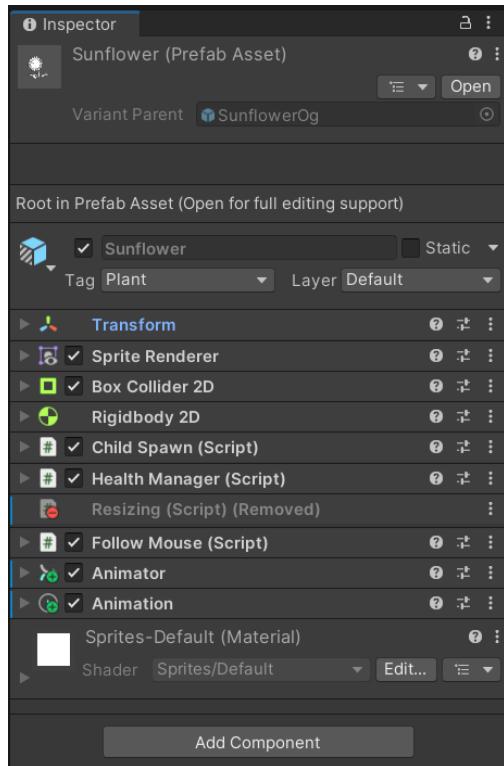
kazy na objekty z nějaké scény. Není problém odkazovat na jiné prefabs, jelikož k nim je neomezený přístup na rozdíl od objektů vázané na scény.

Tento systém využívám například při střílení tobolky durmanem. Po umístění durmanu se v intervalech vytvoří tobolka v místě rostliny, která má skript pro pohyb a sprite, není tedy potřeba při každém vytvoření instance tobolky říkat, jak má vypadat a chovat se.

### 2.1.3 Programování

V Unity se využívá ke psaní skriptů C#. Je k dispozici mnoho unity knihoven a skripty defaultně dědí od MonoBehaviour [5]. MonoBehaviour má vyštavěný je cyklus funkcí. Herní čas se dělí na snímky, ve kterých se v určitém pořadí udávají některé funkce, které můžeme vidět na obrázku č.3.

MonoBehaviour také nabízí funkce, které se spouští při konkrétní události, čímž usnadňují stavění kódu. Například OnTriggerEnter() se spustí, pokud spolu dva objekty kolidují. Pokud by takováto funkce neexistovala, musela by se kolize, spolu se spousty dalšími těmito funkcemi, zjišťovat ve funkci



Obrázek 2: Inspector

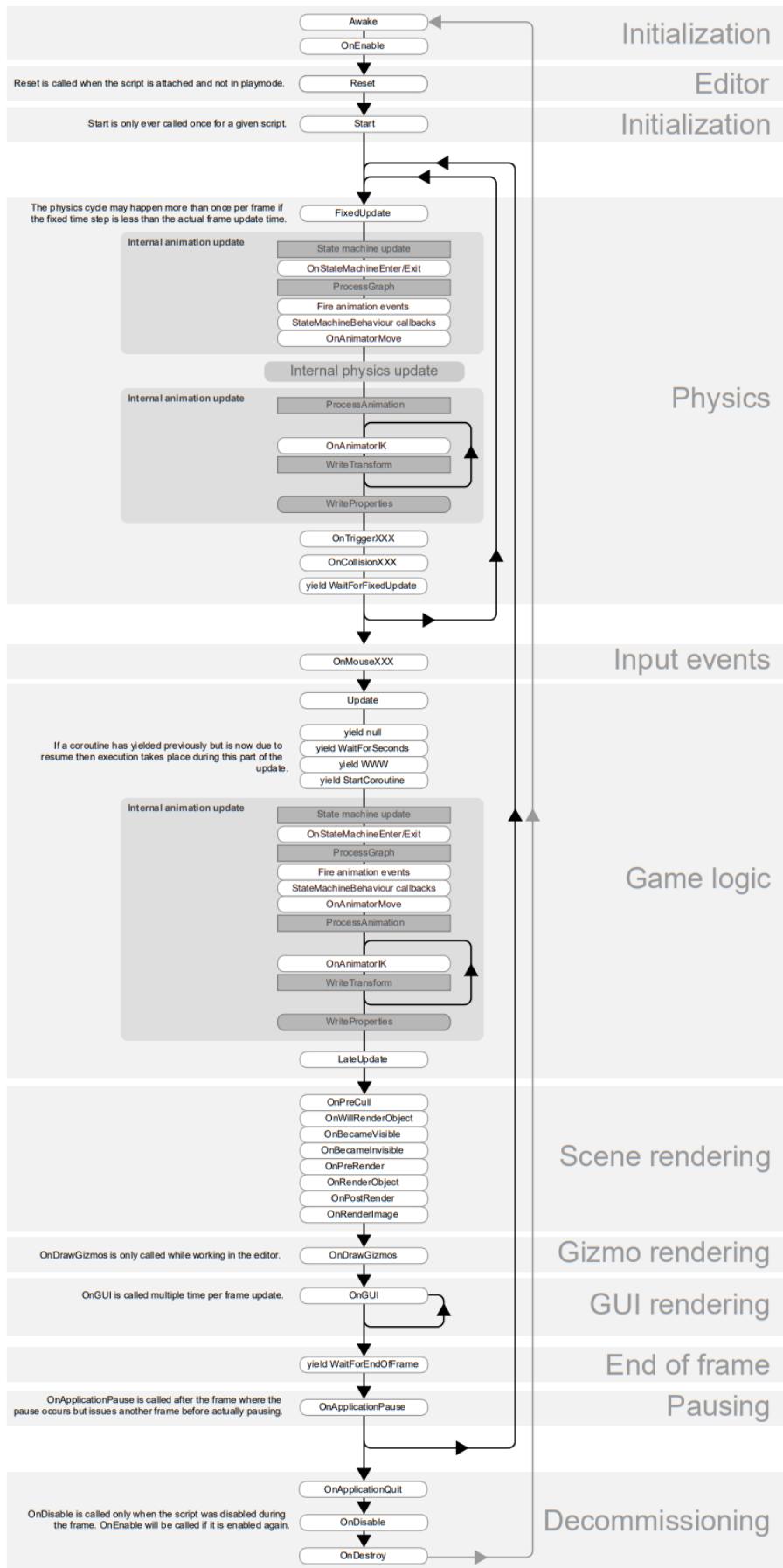
Update(), která se spouští každý snímek, čímž by se stal kód nepřehlednějším a delším.

#### 2.1.4 User Interface

User Interface neboli UI je oddělená část hry, která se vztahuje na obrazovku a ne na scénu, tedy se nehýbe společně s postavou, a je ukotvená. Využívá se například na menu, itinerář, ukazatel životů apod.

## 2.2 Visual Studio 2022

Visual studio jsem používala ke psaní skriptů a debuggování. Jedná se o vývojové prostředí od Microsoftu, využívané k vytváření webových stránek a aplikací. Studio má široké použití, jelikož podporuje 36 jazyků. Nabízí funkci Debug, kdy se kód prochází řádek po řádku s nahlédnutím do vnitřních událostí v konkrétním bodě. Používá IntelliSense, kdy nabízí doplnění kódu,



Obrázek 3: Cyklus skriptů

často až překvapivě přesně[1].

## 2.3 Procreate

V Procreate jsem vytvořila všechnu grafiku ve hře. Procreate je rastrový grafický editor vyvíjený pro iPad. Ke kreslení jsem využívala Apple Pencil. Procreate nabízí jednoduché prostředí vhodné spíše pro vytváření vlastních designů. K dispozici je možnost vytváření vrstev, vybírání si, a stahování, všemožných štětců, a jednoduché úpravy jako rozostření, barevné úpravy a další. Pro složitější fotomanipulaci bych jednoznačně využila spíše GIMP nebo Photoshop, jelikož Procreate je pro tyto účely téměř nepoužitelný.

## 3 Hra

### 3.1 Scény

Ve hře jsou dvě Unity scény: Menu, neboli hlavní menu, které se ukáže hned při spuštění aplikace, a Level, což je zahrada, kde se odehrává lítý boj.

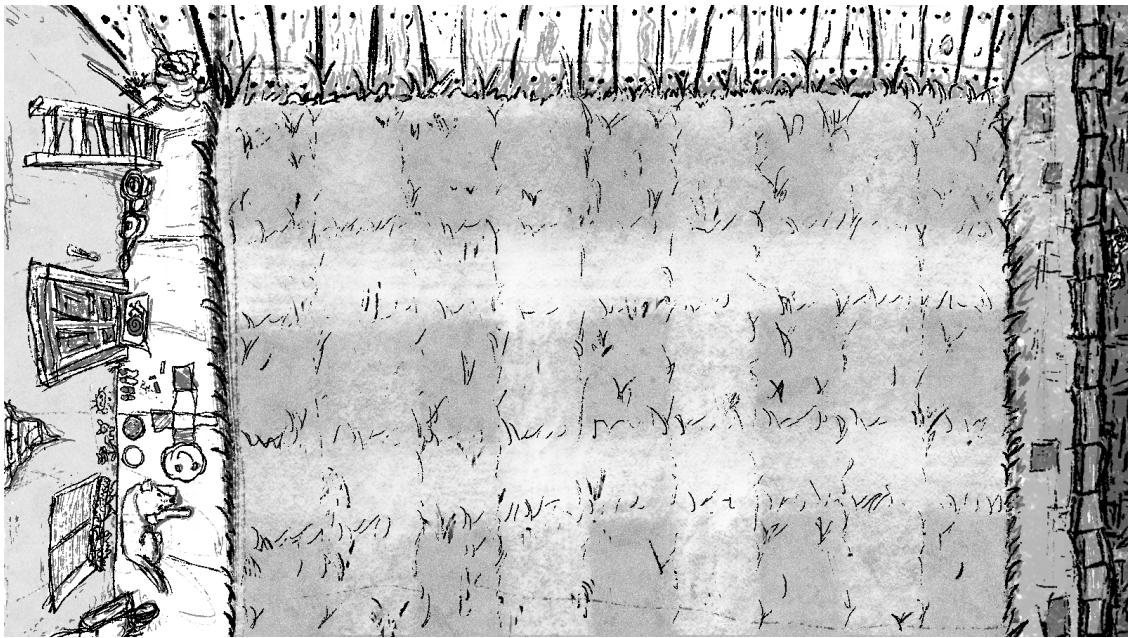
#### 3.1.1 Hlavní menu

Hlavní menu (obr. č.6) jsem celé dělala pomocí UI. Při zapnutí hry jsou k dispozici tři tlačítka na hrobu: PLAY, OPTION a EXIT. Vše je ovládané ze skriptu MainMenu.cs. EXIT pouze vypne aplikaci. OPTION (obr. č.5)(Jelikož v nastavení je k dispozici pouze jedna možnost, přišlo mi vhodné přiznat singulár) přepne z MainMenu na OptionsMenu. Zde se nachází slider, který ovládá hlasitost hudby.

Hráč má možnost kliknout na ceduli vlevo nahoře, pokud si chce změnit aktivního hráče. Při kliknutí se objeví menu, kde si může vybrat z několika slotů. Pokud klikne na prázdný, vyskočí pole, kam zadá jméno. Pokud klikne na již zabraný slot, nastaví se jako aktivní hráč. O ukládání jména se stará skript PlayerName.cs

Při stisknutí tlačítka PLAY se zkontroluje, zda má hráč vybrané jméno, a v Levelmanager.cs spustí funkci LoadGame() s levelem, kde hráč minule skončil, nebo s levelem 1, pokud je nový. Pokud neexistuje aktivní hráč, vyskočí opět pole na zadávání jména a zapíše se automaticky do slotu 1.

Aktivní hráč přetrvává i po vypnutí aplikace. Po opětovném otevření hraje tedy pořád stejný hráč.



Obrázek 4: Zahrada

### 3.1.2 Zahradá

V zahradě (obr. č.4) se celá hra odehrává. Trávník, neboli herní pole, je rozdělen na 50 políček, kam je možno pokládat rostliny. Generování políček má na starost GridMaker.cs. Každé políčko se skládá ze tří součástí. TileFloor, který má collider pouze na dolní hranici, aby se o ni mohla zastavovat slunce. FullTile má collider přes celé políčko a slouží k pokládání rostlin. Tile je prázdný objekt a slouží pouze k přehlednosti, jelikož ve hře je rodič zbylých dvou Tilů a položené rostlinky.

Vlevo je menu pro nakupování rostlin a vpravo od něho je aktuální finanční status hráče, jehož fungování rozebírám v sekci Slunce. V pravo nahoře je tlačítko MENU. Po kliknutí se herní čas nastaví na 0, čímž se zastaví, a ukáže se menu s možnostmi RESUME, RESTART, OPTION a MAIN MENU.

## 3.2 Herní objekty

Ve hře se nacházejí rostlinky, duchové, munice a slunce. Rostlinky si hráč kupuje a ukládá na herní pole. Jsou to jediné věci, které hráč může ovládat. Přidávají



Obrázek 5: Nastavení



Obrázek 6: Hlavní menu

mu sluníčka, neboli herní měnu, a útočí na duchy. Duchové jsou nepřátelé, kteří chodí z opačné strany zahrady. Úkolem hráče je všechny zlikvidovat, aniž by prošli do domu. Některé rostliny střílejí munici, jiné vytvářejí slunečnice.

### 3.2.1 Rostliny a munice

Rostliny mají přiložené skripty ChildSpawn.cs, HealthManager.cs a FollowMouse.cs. ChildSpawn.cs je jednoduchý skript, v intervalech vytvoří potomka rostliny, což je buď munice, nebo slunce.

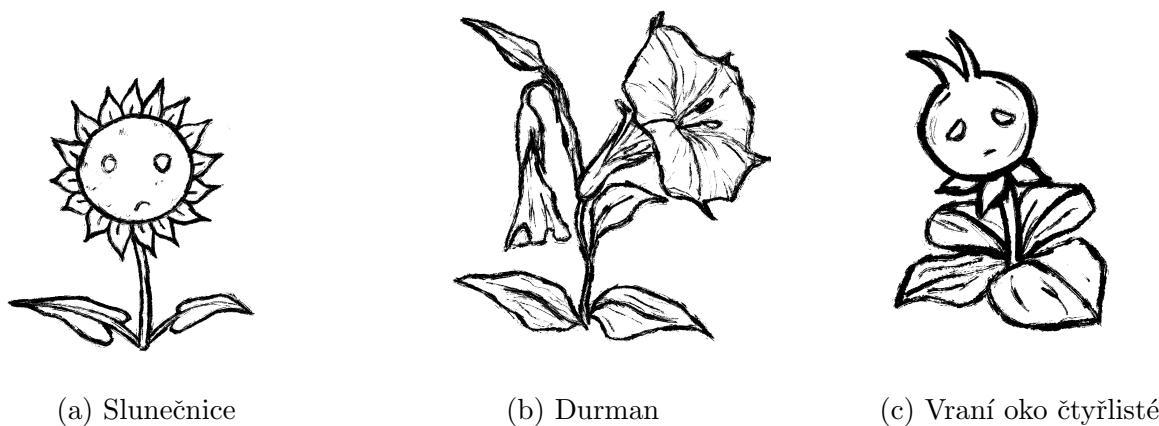
HealthManager.cs udává, kolik životů objekt má a kolik ubírá při kolizi. Tento skript používají i duchové a munice. Pokud je rostlina v kolizi s duchem, ubírají se jí v intervalech životy a pokud klesnou pod nulu, rostlina mizí.

FollowMouse.cs se vztahuje ke kupování a pokládání rostlin, což je rozviniuto níže.

Ve hře se vyskytují tyto rostliny:

- Slunečnice (obr č.7a) - slunečnice je v prozatímní verzi jediná rostlina schopná vytvářet sluníčka, je tedy nepostradatelnou součástí každé funkční zahrady. Stojí 50 sluníček.
- Datura (obr č.7b) - neboli durman, je první útočná rostlina, kterou hráč odemkne. Stojí 100 sluníček a nahrazuje funkcí hrášek v původní PvZ. Při větším nároku na originalitu bych se snažila implementovat, že pokud je duch zasažen tobolkou, vyfantazíruje si buď další rostliny nebo další duchy, což by bylo rozhodnuto náhodně, a přiblížilo by to tak hráči nevypočitatelnost deliriantů.
- Ořech - stejně jako v původní hře slouží ořech ke zdržení nepřátel. Má více životů než ostatní rostliny, zato nic neprodukuje. Používá se jako spotřební rostlina a málokdy vydrží do konce hry. Stojí 50 sluníček.

- Vraní oko čtyřlisté (obr. č.7c) - vraní oko je druhá, a v této verzi i poslední, útočná rostlina. Při koupení má na sobě bobuli, kterou při prvním kontaktu s duchem ztrácí, a ducha na místě zabije. Poté stále žije. Jeho hlavní úkol je zdržet duchy a jelikož je jedovatý i stonek, ubírá duchovi trochu životů. Než umře, sebere stejně životů jako pět tobolek, tedy přibližně polovinu životů ducha. Stojí 150 sluníček.

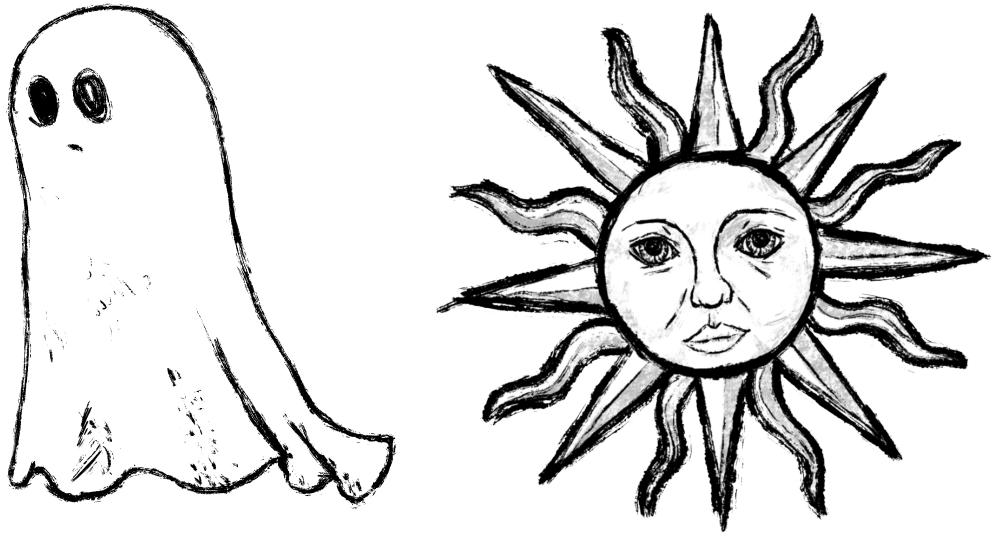


Obrázek 7: Rostliny

### 3.2.2 Duchové

Duchové (obr. č.8a) jsou mojí verzí zombie. Přicházejí z pravé strany zahrady a pojídají rostliny, na které narazí. Pohybují se v jedné z lajn. Mají skripty Move.cs, HealthManager.cs, CollisionDetector.cs. Move.cs zaštiťuje pohyby munice a duchů a kontroluje jejich pozici. Pokud má tag "Ammo", pohybuje se doprava, pokud "Ghost", pak doleva. Pokud je objekt mimo obrazovku, zničí se, a pokud duch vyjde z herního pole, objeví se Game Over scéna.

Zobrazování duchů je spravované SpawnManager.cs. Nejdříve duchové chodili v pevně daných intervalech, což ale vedlo k tomu, že v první části levelu hráč neměl dostatečné prostředky na to se ubránit, naopak ke konci hry, kdy měl hráč již nakoupeno dostatečné množství durmanů, je duchů málo a hra přestává být zajímavá.



(a) Duch

(b) Slunce

Obrázek 8: Další herní objekty

Po úporném zkoumání PvZ jsem dospěla k tomu, že čas je pevně daný u prvních 4 zombií, kdy přijdou po 20s, 30s, 20s, 15s. Poté chodí častěji.

V mé verzi jsem pro prvního ducha nastavila zpoždění 20s, poté chodili po  $(10 + 20/\text{index ducha})\text{s}$ . Poslední třetina duchů chodí v intervalu  $\langle 0; 0.5 \rangle$  za sebou a zbytek v intervalu  $\langle 3; 5 \rangle$ .

### 3.2.3 Slunce

Slunce (obr. č.8b) mají pouze dva skripty, SunMovement.cs a SunCollecting.cs. Slunce padají buď z rostlin, nebo z nebe. V prvním případě jsem při vytvoření instance použila funkci AddForce(), která objekt "vystřelí" v náhodném směru.

V druhém případě se ke slunci nepřistupuje jako k objektu s hmotností a působením gravitace, aby při pádu nezrychlovalo. V původní hře se slunce zastaví náhodně na herním poli. Toto jsem vyřešila náhodně vygenerovaným celým číslem od 1 do 5, které se zmenší pokaždé, co potká collider, resp. projde TileFloor. Pokud je číslo 0, slunce se zastaví.

Při kliknutí na slunce zmizí a ke score se přičte 25. Toto obstarává skript

### 3.3 Kupování rostlin

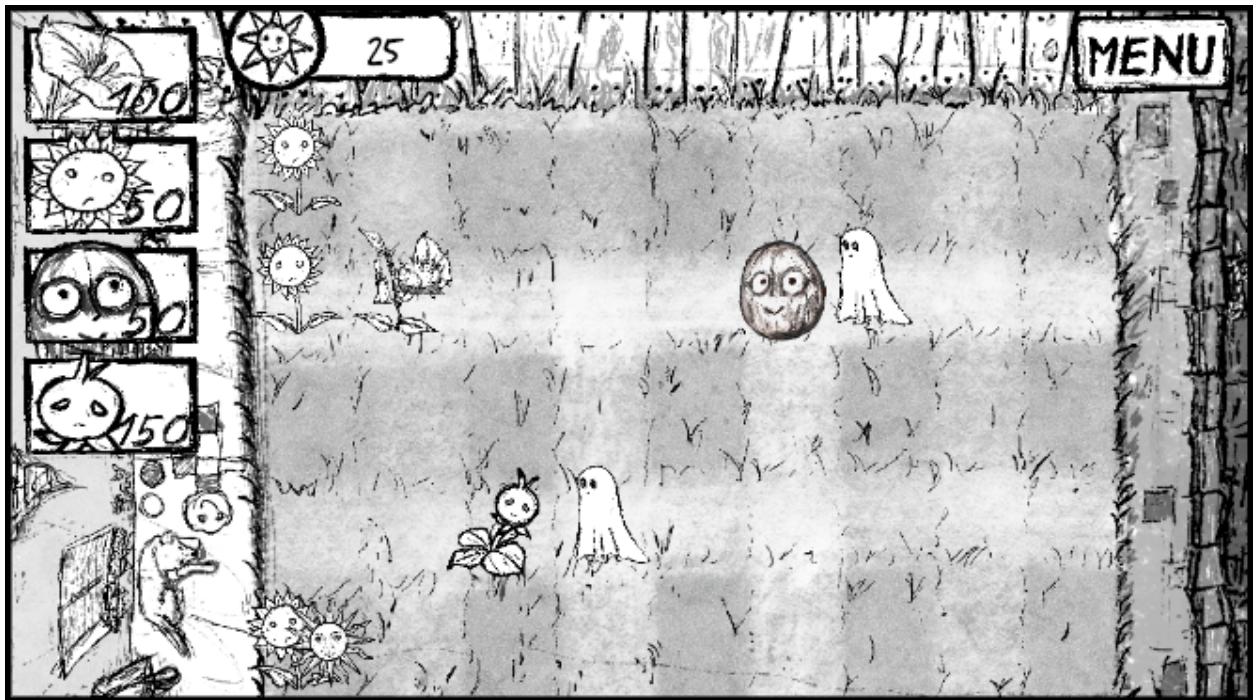
Hráč si za sluníčka, které obdržel z nebe nebo ze slunečnic, kupuje další rostliny k obraně domu. Výběr rostlin závisí na levelu.

V každém levelu jsou k dispozici všechny rostliny z předchozího levelu a k tomu jedna navíc. V původní hře je od určitého levelu k dispozici si rostliny vybírat, to ale už přesahuje rozsah této práce. Každá rostlina má vlastní nákupní button (obr. č.9), který se jmenuje stejně jako rostlina s přidanou koncovkou "UI". V UI prostředí je objekt PlantSelection, který má přiložený skript PlantSelectionManager.cs. Ve funkci Load() se načte list rostlin, které mají být k dispozici k nákupu, a spustí se funkce Display(). Zde se porovnává jméno rostliny se jmény objektů v listu UIs a pak se vytvářejí instance.

Při hře se při kliknutí na konkrétní button se spustí skript CreateInstance.cs, konkrétně funkce OnClick(). Rostlina jde koupit pouze pokud má hráč dostatečné finanční prostředky a již jednu rostlinu nedrží. Pokud jsou podmínky splněny, odečte se cena rostliny od peněz a vytvoří se instance konkrétní rostliny.

V tuto chvíli rostlina nevytváří potomky, nezaznamenává kolize, sorting layer je nastavená na moving objects, aby se pohybovala nad ostatními objekty. Také je spuštěný skript FollowMouse.cs. Ten dává vědět ostatním skriptům, že hráč drží rostlinu, a spouští skript PutDown.cs.

PutDown kontroluje ve funkci Update(), zda hráč klikl na nějaké políčko. Pokud už na něm rostlina není, vypne se skript FollowMouse.cs, nastaví se pozice a rodič, změní se sorting layer na Plants a zapne se skript Child-Spawn.cs.



Obrázek 9: Hra

### 3.4 Práce s daty

Ve hře jsem chtěla ukládat data o jednotlivých hráčích externě, aby i po zavření aplikace bylo možné se vrátit do stejného levelu. Nejdříve jsem se pokusila o postavení save systému podle Youtube tutorialu [2], což selhalo. Nakonec jsem našla jednodušší alternativu nabízenou přímo Unity. Tento systém se mi však líbil a nechěta jsem, aby přišel vždycky, tak jsem ho využila pro zaznamenávání dat levelů.

#### 3.4.1 Level data

Data jsem ukládala do XML souboru. Každý soubor si ukládá číslo levelu, nové odemknuté rostliny, jestli je den a jaké duchové chodí. Je strukturovaný pro co nejjednodušší zápis levelů, takže pro každý level se píše pouze nová rostlina a ostatní se doplní v LevelManager.cs ve funkci CompleteDatabase(). Do XML se dají ukládat pouze jednoduché datové typy, proto byl problém ukládat typ GameObject. Ukládám je podle jména a později je nacházím

v LevelDataProcessor v listu PlantsAndGhosts, kde jsou všichni duchové a rostliny.

Se souborem potom pracuje XMLManager, který data načítá a předává dál. XMLManager je jeden z kódů, které jsou singleton a mají pouze jedinou instanci. Tato praktika se používá převážně v práci s daty, aby nedocházelo k duplikaci požadavků. Tento skrip je stejný, proto jako jeden z mála používá funkci Awake, ve které volá ostatní managery a skripty, aby nedošlo k tomu, že si ostatní skripty volají neexistující XML databázi. Funkce Load() načte či vytvoří soubor level\_data.xml a převede na typ LevelDatabase, což je list XMLSave.

Poté zavolá LevelManager.cs a tím jeho práce končí. XML soubor je velmi izolovaný a má k němu přístup právě jedna třída, čímž se jednoduše zamezí korupci dat.

LevelManager je také singleton a má za práci starat se o načítání levelů, přepínání mezi leveley a práci s daty, která zahrnuje zpracovat XML databázi, doplnit ji o odemčené rostlinky, které v souboru nejsou, a poslat dál do LevelDataProcessor.cs. Celé načítání je popsáné v obrázku č.10

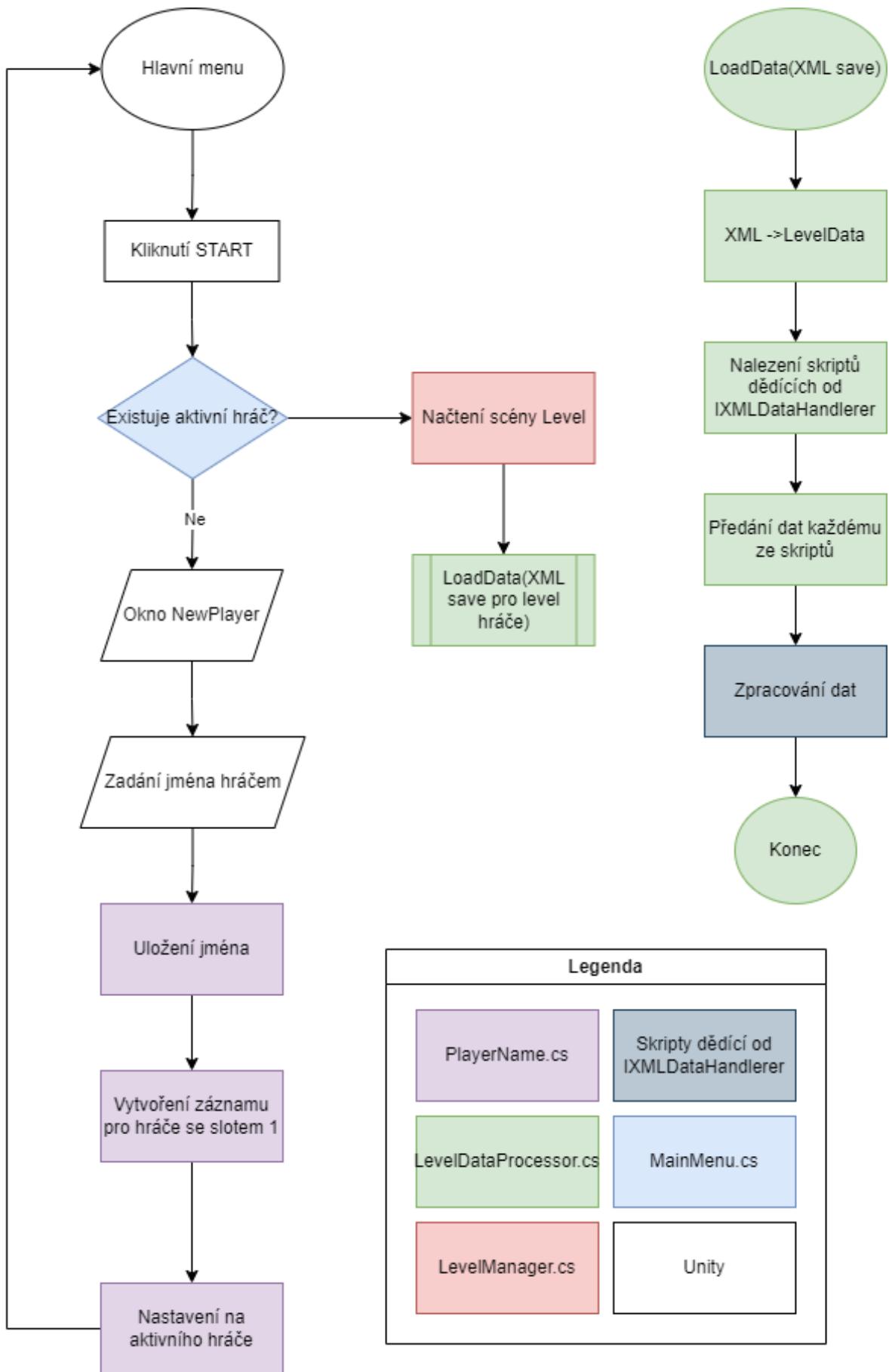
LevelDataProcessor zaštituje jednotlivý level. Ve funkci LoadData() dostane jeden XMLSave, jenž změní na datový typ LevelData. Jediný rozdíl mezi nimi je ten, že LevelData, jelikož je vytvořený pouze k vnitřnímu použití v Unity, dokáže uchovávat objekty typu GameObject. LevelDataProcessor poté předá LevelDatabase každému skriptu, který dědí od IXMLDataHandlerer interfacu, který si data zpracuje.

### 3.4.2 PlayerPrefs

Pro informace o hráči jsem zvolila built-in třídu PlayerPrefs, která umožňuje ukládání jednodušších dat. Původně jsem chtěla postavit podobný systém



Obrázek 10: Načítání dat



Obrázek 11: Načítání levelu



Obrázek 12: Sloty

jako s XML soubory, což by ale bylo nepřiměřeně složité vzhledem k tomu, že pro každého hráče ukládám pouze level, kterého dosáhl, jméno a hlasitost hudby.

Každý hráč má jeden save slot, jehož index se ukládá do PlayerPrefs jako name1 a level1. Index také slouží k identifikaci aktivního hráče.

### 3.4.3 Sloty

Na začátku hry si hráč může zvolit jeden ze tří slotů (obr. č.12), kam zadá jméno. Automaticky se mu nastaví level na 1 a hlasitost na maximum. Při zvolení hráče se jeho jméno zobrazí na ceduli vlevo nahore.

Při zaplnění všech slotů je možnost vymazat všechna data tlačítkem Delete all saves.

## 3.5 Game Over

V momentě, kdy je počítadlo duchů v SpawnManager.cs 0, spustí se koprogram v LevelManager.cs, který kontroluje, zda se ještě nějací duchové nachází



Obrázek 13: Game Over

v hierarchii. Pokud ne, zobrazí se na pár sekund UI scéna s vítězstvím( obr. č.15), uloží se do PlayerPrefs zdolaný level a načte se nový. Načítání levelu trvá déle než jeden snímek, proto jsem vytvořila časovač, který po zavolení funkce SceneManager.LoadScene() počká 2 snímky, než rozdá data ostatním objektům. Asi to není nejlepší řešení, ale funguje.

Pokud duch vejde do domu, resp. projde do levé pětiny obrazovky, ukáže se scéna GameOver( obr. č.13), která nabízí hráči bud' zkusit level znova, nebo odejít do hlavního menu.

V případě, kdy hráč zdolal všechny levele, ukáže se finální scéna( obr. č.14) s možností návratu do hlavního menu. Z tohoto účtu již nejde dále hrát.

Při zobrazení nápisů se vždy nastaví herní čas na 0, tedy se zastaví, zastaví se i hudba v pozadí a zahraje náležitá znělka.

## 3.6 Testy

Nezbytnou součástí vývoje jakékoliv aplikace jsou testy, které odhalí neřádné chování programu. Kvůli malému týmu i projektu jsem zvolila testy manuální.



Obrázek 14: Úplný konec



Obrázek 15: Vítězství

V rámci tohoto testování jsem například objevila chyby:

- Nevyvážené intervaly mezi vytvářením munice, rychlostí duchů a hodnotou rostlin. Docházelo k tomu, že než si hráč stihl koupit durman, už duch vcházel do domu.
- Vytváření duchů moc daleko za hranicí, tudíž hned zmizeli, jak jim bylo řečeno ve skriptu Move.cs, který automaticky maže objekty mimo viditelnou obrazovku, například vystřelené tobolky.
- Nespolehlivé zaznamenávání financí. Při vytvoření slunce si našlo aktuální finanční status hráče a poté při sbírání ho zvýšilo o 25. Problém byl v případě, kdy byla velká prodleva mezi vytvořením a sebráním slunce, jelikož přičítal ke špatné hodnotě. Stávalo se tak, že sebrání slunce snížilo kredit třeba o 100 sluníček.

### 3.7 Hudba

Hdubu má na starost SoundManager s potomky Effects a Music. Tato svatá trojice se nachází v obou scénách. Mým prvním plánem bylo vytvoření pouze jedné trojice, která by zasahovala do obou scén, z čehož nakonec sešlo. Music se stará a hudbu v pozadí a Effects naopak o jednotlivé zvukové efekty.

Hudba v pozadí použita ve hře je vytvořená generátorem Terryho Davise pro operační systém TempleOS. Skrz generátor má promlouvat Bůh, tudíž je hudba přímo do něj. Jelikož již generátor není volně přístupný, hudbu jsem stáhla z [http://terrydavis.mooo.com/stuff/Terry\\_Davis/](http://terrydavis.mooo.com/stuff/Terry_Davis/), kde je rozdělen záznam na jednotlivé skladby díky u/esaym z reddit.com [6].

Zvukové efekty jsou, doufám, k všeobecnému zklamání z prostějšího zdroje <https://pixabay.com/sound-effects/search/coin/>. Stáhla jsem zde efekt



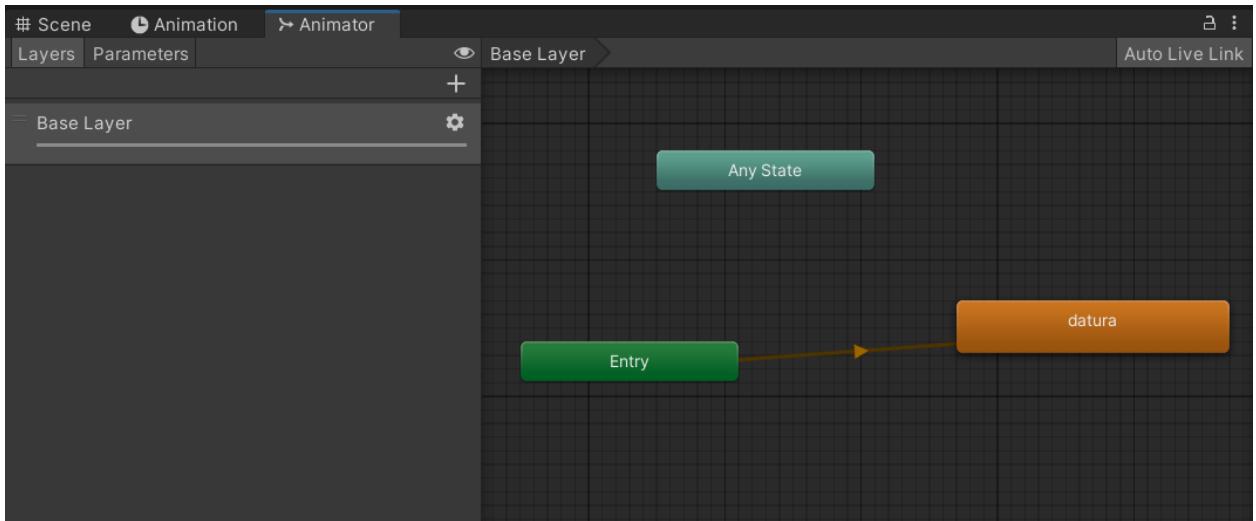
Obrázek 16: Animation

pro kliknutí na tlačítko, sbírání sluncí, kupování a pokládání rostlin a znělky pro konec levelů.

Hlasitost zvuku se ovládá se skrz OPTION tlačítko při vstoupení do libovolného menu. Sliderem se nastaví hlasitost, která se udává jako float v rozmezí mezi 0 a 1. Při změně se uloží preference do PlayerPrefs jako volume + id hráče. Rozdelení audia do tří objektů je základ pro možnost vypnutí zvláště hudby a efektů, což bohužel není naimplementované.

### 3.8 Animace

Všechnu grafiku jsem dělala v Procreate. Duchové a rostlinky mají animace složené z osmi až třinácti snímků. K animaci objektů jsem použila komponenty Animator (obr. č.17). V tomto prostředí se přidávají jednotlivé animace (obr. č.16) pro různé stavy, přechody mezi nimy a podmínky pro přechod. Jelikož duchové i rostlinky mají pouze jednu animaci, jsou Animators pouze dvouprvkové.



Obrázek 17: Animator

## 3.9 Problémy

### 3.9.1 Velikost spritů

Původně měl každý objekt mít přiložený skript Resizing.cs, který by upravil jeho velikost. Problém je v tom, že velikost objektu není jednoduše upravitelná, jelikož se udává jako poměr potomku k rodiči. Ačkoliv jsem s tím počítala, nikdy se mi skript nepovedl odladit. Pokaždé byly objekty bud' nepřiměřeně velké nebo nebyl dodržen původní poměr stran. Na tento skript jsem tedy zanevřela. Upravila jsem prefabs na kýženou velikost vůči pozadí a při každém vytvoření instance jsem ho nastavila jako rodiče, kterého jsem poté změnila, pokud bylo za potřebí.

### 3.9.2 Nekonzistentní pořadí skriptů

Některé skripty drží data, na kterých závisí celá hra, například LevelManager a XMLManager. V původní verzi jsem používala funkce Start() a Awake() u většiny objektů. To vedlo k tomu, že se občas zavolala funkce například ze slunečnice dříve, než se načetla jakákoli data, která ona požadovala.

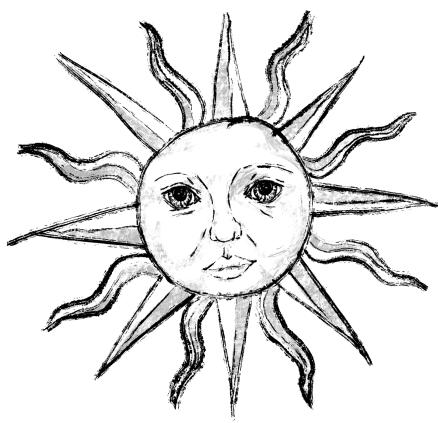
Jako řešení jsem nechala funkce Awake() pouze k tomu, aby se vytvořila

statická instance u některých objektů. Jediný skript využívající tuto funkci je XMLManager.cs. Ostatní skripty se musí explicitně volat, čímž se zajistí exekutivní posloupnost.

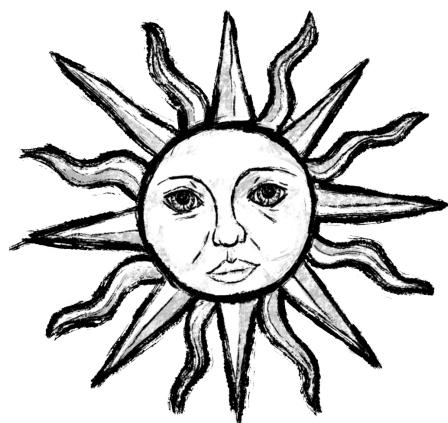
Další možnost je využívat Script Execution Order, kde je možnost skripty přeuspořádat. Toto využívám například při načítání levelu, kdy se mi nejdříve vytvoří šachovnice a až poté ostatní obejky.

### 3.9.3 Grafika

V designu je velmi důležité dbát ne pouze na jednotlivé objekty, ale na celkový dojem ze hry. Toto jsem věděla, ačkoliv podcenila. V původní verzi sice vypadaly všechny objekty z blízka v pořádku, po zmenšení a přidání na herní pole se toto však změnilo a rostliny, duchové i slunce velmi splývaly s okolím. Bylo to způsobené až přílišnou detailností při kreslení a tenkými obrysů. Porovnání staré a nové verze je na obr. č.18



(a) Původní slunce



(b) Aktuální slunce

Obrázek 18: Porovnání grafiky

## 3.10 Rozdíly v původní hře

Moje verze hry je velmi zjednodušená z časových důvodů. Směr, kterým by se měla hra udávat, je vcelku jasný, jelikož celá práce spočívá v napodobování

již existující hry. Cílem bylo dodělat vše, co má původní hra a mojí chybí. Současně však bych ráda i dodělala některé vlastní rostliny se speciálními schopnostmi.

### 3.10.1 Kvantitativní část

V čem je napřed původní PvZ je rozsah hry. Pokud bych chtěla dosáhnout stejné funkcionality, bylo by za vhodno implementovat více druhů duchů s různými vlastnostmi a to stejné provést i u rostlin. Některé rostliny v původní hře nevytváří žádné potomky, ale například vybuchují při dotyku se zombiem, nebo zombii sní celou najednou. Toto doimplementovat by již nebylo složité, jelikož jde jen o jednoduché úpravy. Náročné by to bylo časově, kvůli grafické části.

V PvZ zombie nechodí pravidelně, ale nachází se v průběhu hry několik vln, kdy přijde nadměrné množství zombií. O vlnách hráče informuje ukazatel hry, který se nachází v rohu obrazovky. Zombie také nejsou všechny stejné, některé jsou odolnější, jiné mají speciální schopnosti.

Rozdílné druhy zombií bych implementovala upravením `SpawnManager.cs`. Místo kombinace funkcí `SpawnGhost()` a `InvokeRepeating()` bych volila to, že každý druh duchů by měl přiřazené číslo, jak časté má být jeho zjevování. Čím menší číslo, tím vzácnější duch. Výběr ducha by probíhal náhodně zvoleným číslem a každý druh by měl přiřazený interval. Pokud je číslo v jeho intervalu, objeví se. Interval by se určoval od nejjednoduššího ducha, resp. základní duch bude mít interval  $[0;5]$ , duch s kbelíkem na hlavě  $(5;7)$  atd. s tím, že horní hranice nádohného čísla by se zvyšovala v průběhu hry, aby nepřišel neporazitelný duch hned v první minutě.

Nekonzistentní vytváření zombií by šlo udělat koprogramem, který by po-zrdžoval `SpawnGhost()` o náhodný počet sekund.

### **3.10.2 Grafická část**

Nejvíce výrazný je rozdíl v grafice, která je celá autorská práce, kterou jsem navrhla a nakreslila. Místo zombií na dům útočí duchové, což bylo zvoleno jak z estetických důvodů, tak proto, že animace ducha není ani zdaleka tak náročná, jako animace chůze.

Další věc, co se nachází v původní hře, jsou animace pro případ, kdy zombie jí rostlinu. Také se zombiím a rostlinám mění vzhled podle toho, kolik životu jim zbývá, což nakreslit by vyžadovalo alespoň 3x tolik souborů k animaci.

Po vyhrání levelu z poslední zombie spadne karta nové rostliny. V případě proniknutí zombie do domu je na ni záběr. V mojí verzi se pouze ukážou nápisy informující hráče o úspěšnosti jeho strategie.

Zahrada není stejně rozvržená, na rozdíl od původní hry má méně prostoru nahoře a naopak více vlevo, kde je vidět dům. Toto ovlivnilo umístění menu na kupování rostlin. V původní hře je nahoře, zde vlevo. Umístění počítadla sluncí pak bylo kvůli tomuto krkolumnější. V mobilní verzi PvZ je rozvržení podobné mému, odtud jsem čerpala design pro karty.

### **3.10.3 Verze zahrady**

Některé leveley v PvZ se odehrávají noci, kdy je možné používat navíc houby, které jsou levnější než denní rostliny, což je nejspíše způsobené tím, že nepadají slunce z nebe. Pokud se hráč pokusí použít houby za dne, schovají se a jsou neaktivní.

K tomuto jsem se již nedostala, ale v souboru s daty o levelech se stále nachází možnost pro každý level vyplnit bool, zda je den. Toto by asi byl jeden z programátorský jednoduších doplňků. Funkce `SpawnSlunicko()` by zůstala neaktivní a na obrazovku by se přidal poloprůhledný černý nebo šedý

obdélník. Houby by musely mít navíc funkci zabraňující jim ve fungování, pokud je den.

Další verze zahrady v PvZ obsahuje bazén. Zahrada má v těchto levelech šest pruhů místo pěti, z čehož dva prostřední jsou voda. Aby hráč mohl zasadit květinu na vodní políčko, musí tam nejdříve umístit leknín. K dispozici jsou také vodní rostliny, které se nedají použít na souši.

Při některých levelech jsou na zahradě hroby, na jejichž políčka není možné sázet rostliny. Při každé z vln s nich vylezou další zombie. Je možné se jich zbavit rostlinou Grave Buster, která hrob sní a uvolní místo.

### 3.10.4 Minihry a Obchod

Od určitého levelu jsou v PvZ k dispozici minihry, kde si hráč rostliny ne-kupuje, ale samy přijíždí na pásu. Každá minihra má vlastní ozvláštnění, například chodí zombie s rostlinami místo hlavy, tedy útočí na hráčovo rostliny již z dálky, nebo chodí miniaturní verze zombií [4].

V průběhu hry se odemyká obchod Crazy Dave's Twiddydinkies, kde je možné zakoupit například další místo na kart rostlin, či různá vylepšení na zahradu [3].

O obchod bych se nepokoušela, jelikož jde o novou scénu, tudíž by to zahrnovalo grafickou práci a složitější komunikaci mezi scénami.

## 4 Závěr

Cílem mojí seminární práce bylo vytvořit zjednodušenou verzi PvZ s vlastní grafikou. Jako vývojové prostředí jsem využila Unity a ke grafickému zpracování Procreate. Vytvoření i v podstatě jednoduché hry zabralo o hodně více času, než jsem čekala, což ale je způsobené částečně i tím, že s Unity jsem neměla žádné zkušenosti. Je ještě spoustu místa pro zlepšení hry, at' už rozšířením diverzity, přidáním dalších funkcí, či pouze optimalizováním kódu.

Unity má kolem sebe velkou komunitu, tudíž je práce s enginem velmi vstřícná v oblasti debuggování a podpory. Unity je přesně v tom bodě, kdy je vývoj hry zjednodušen, ale zároveň není omezený, tudíž mi přijde jako vyhovující engine minimálně pro nenáročné vývojáře. Nad rozdíly při vytváření koplexnější či profesionálnější hry v různých enginech si nedovolím spekulovat.

Procreate jsem zvolila z důvodu dostupnosti. Je uživatelsky vcelku jednoduché a při vytváření grafiky záleží spíše na dovednostech umělce, než na programu samotném.

Při vývoji jsem postupovala od jádra hry po kosmetické úpravy. Minimální životaschopný produkt bylo herní pole s výběrem dvou rostlin, slunečnice a jedné útočné. Ve hře mělo být počítadlo sluníček, která padají jak z rostlin, tak z nebe. Dále měli chodit nepřátelé v pravidelných intervalech v určitém počtu.

Ve výsledku je hra komplexnější. Hráč má k dispozici uživatelské účty, které přetrvávají i po zavření aplikace, a možnost si pro každý zvolit hlasitost hudby. V nynější verzi se nacházejí 4 rostliny, každá s jiným způsobem boje. Duchové chodí náhodně, nejdříve zřídka a poté se intenzita zvyšuje. Napodobuje to tedy PvZ a level je takto po celé délce pro hráče stejně náročný.

Výsledkem mojí semiární práce plně funkční hra s originální grafikou a cíl práce byl tedy splněn.

## Reference

- [1] Microsoft Corporation. Visual studio.  
<https://visualstudio.microsoft.com/vs/>, 2023.
- [2] Board To Bits Games. Saving data in unity: Xml files. [https://www.youtube.com/watch?v=6v11IYMpwVQ&ab\\_channel=BoardToBitsGames](https://www.youtube.com/watch?v=6v11IYMpwVQ&ab_channel=BoardToBitsGames), 2016.
- [3] Plants vs. Zombies Wiki. Crazy Dave's Twiddydinkies.  
[https://plantsvszombies.fandom.com/wiki/Crazy\\_Dave%27s\\_Twiddydinkies](https://plantsvszombies.fandom.com/wiki/Crazy_Dave%27s_Twiddydinkies),
- [4] Plants vs. Zombies Wiki. Plants vs. Zombies Mini-games.  
<https://plantsvszombies.fandom.com/wiki/Mini-games>.
- [5] Unity Technologies. Monobehaviour - unity scripting api.  
<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>, 2022.
- [6] u/easym. RIP of God/Worship/Faith. [https://www.reddit.com/r/TempleOS\\_Official/comments/ag20fd/rip\\_of\\_godwordfaith/](https://www.reddit.com/r/TempleOS_Official/comments/ag20fd/rip_of_godwordfaith/), 2019.
- [7] Unity Technologies. Unity Engine.  
<https://unity.com/products/unity-engine>.
- [8] Wikipedia. Plants vs. Zombies (video game). [https://en.wikipedia.org/wiki/Plants\\_vs.\\_Zombies\\_\(video\\_game\)](https://en.wikipedia.org/wiki/Plants_vs._Zombies_(video_game)), 2024.