



Projekt

Vývoj akční videohry zasazené do prostředí FM

Studijní program:

B0613A140005 – Informační technologie

Autor práce:

Radek Mocek

Vedoucí práce:

Ing. Jan Hybš

Liberec 2023

Tento list nahradte
originálem zadání.

Prohlášení

Prohlašuji, že svůj projekt jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mého projektu a konzultantem.

Jsem si vědom toho, že na můj projekt se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mého projektu pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li projekt nebo poskytnu-li licenci k jeho využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Beru na vědomí, že můj projekt bude zveřejněn Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

6. 5. 2023

Radek Mocek

Vývoj akční videohry zasazené do prostředí FM

Abstrakt

Český abstrakt

Klíčová slova: klíčová slova česky

Development of an action video game set in FM

Abstract

English abstract

Keywords: keywords in English

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 7 |
| 2 | Herní engine Unity | 8 |
| 2.1 | Editor a základní stavební prvky | 8 |
| 2.2 | 2D tvorba | 9 |
| 2.2.1 | Sprite | 9 |
| 2.2.2 | Animace | 10 |
| 2.2.3 | Tilemap | 10 |
| 2.2.4 | Universal Render Pipeline | 11 |
| 3 | Návrh videohry | 12 |
| 3.1 | Příběh | 12 |
| 3.2 | Žánr | 12 |
| 3.3 | Vizuální stránka | 12 |
| 3.4 | Herní mechaniky | 13 |
| 3.4.1 | Pohyb | 13 |
| 3.4.2 | Boj | 13 |
| 3.4.3 | Grafické uživatelské rozhraní | 14 |
| 3.5 | Umělá inteligence | 14 |
| 4 | Implementace | 15 |
| 5 | Kritické zhodnocení | 16 |
| 6 | Závěr | 17 |
| | Seznam použité literatury | 18 |

Seznam obrázků

| | | |
|-----|---------------------------------|----|
| 2.1 | Unity editor | 9 |
| 2.2 | 9-slicing | 10 |
| 2.3 | Rule Tile | 11 |
| 3.1 | Druhy obrysů u spritu | 13 |

1 Úvod

2 Herní engine Unity

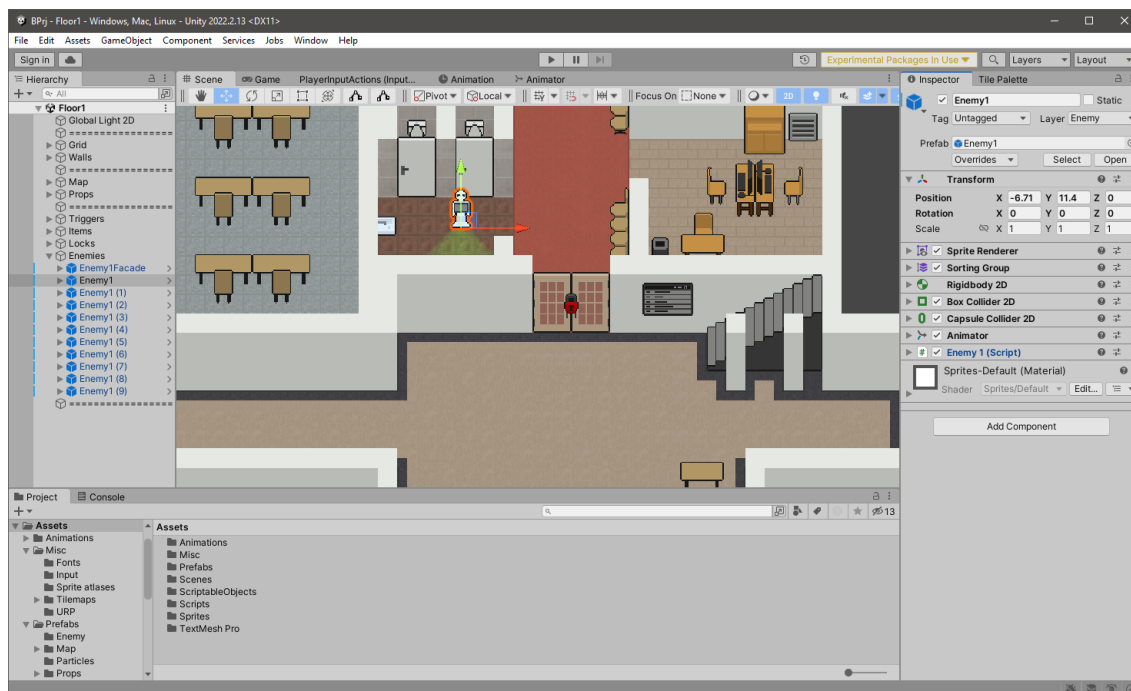
Unity je multiplatformní herní engine s uzavřeným kódem vyvíjený společností Unity Technologies. První veřejná verze vyšla v roce 2005 a od té doby si Unity vybudovalo početnou komunitu a stalo se populární především mezi nezávislými vývojáři. Unity poskytuje nástroje pro vývoj 2D a 3D her, využívá se ale také ve filmovém nebo automobilovém průmyslu. Základní verze Unity je zdarma.

2.1 Editor a základní stavební prvky

Základem každého Unity projektu jsou tzv. *assets*. Asset je jakýkoliv prvek, který je v projektu využit k vytvoření finální hry nebo aplikace. Asset je reprezentován ve formě souboru a často se jedná o prvek tvořící audiovizuální stránku hry (např. 3D model, textura, sprite, zvukový soubor). Dalšími důležitými *assets* jsou scény, které slouží jako herní prostředí a uchovávají v sobě ostatní *assets* v podobě herních objektů, a skripty psané v jazyce C#, které definují chování daných herních objektů. [1]

Každý prvek použitý ve scéně je herním objektem. Herní objekt nemá sám o sobě žádnou funkcionalitu, jedná se pouze o kontejner pro komponenty, které pak určují jeho vzhled a chování [2]. Základní komponenta umístěná na téměř každém herním objektu je *Transform*, která určuje jeho umístění, rotaci a měřítko v prostoru scény. Do některých komponent lze jako argument dosadit asset, např. dosazení spritu do komponenty *Sprite Renderer*, která zajistí jeho vykreslení. Pro opakované použití stejného herního objektu je vhodné z něj vytvořit tzv. *prefab*. Prefab je asset reprezentující herní objekt a změna hodnot v jeho komponentách se projeví ve všech jeho instancích.

Unity editor lze ve svém výchozím rozložení rozdělit na čtyři základní části. Dole se nachází průzkumník projektu, ten umožňuje procházení, třídění a importování *assetů* do projektu. Uprostřed je editor scény, ve kterém si lze scénu prohlížet a vybírat v ní obsažené herní objekty pro následné úpravy. Nalevo od editoru scény je průzkumník její hierarchie, v něm jsou vidět všechny herní objekty (včetně instancí *prefabů*), které jsou obsaženy v právě prohlížené scéně. V neposlední řadě se na pravé straně nachází inspektor, který zobrazuje detailní informace o právě vybraném herním objektu nebo *assetu* a umožňuje jeho modifikaci. Pro herní objekt zobrazuje všechny jeho komponenty, zatímco u *assetů* se zobrazení informací liší podle typu zvoleného *assetu*.



Obrázek 2.1: Unity editor

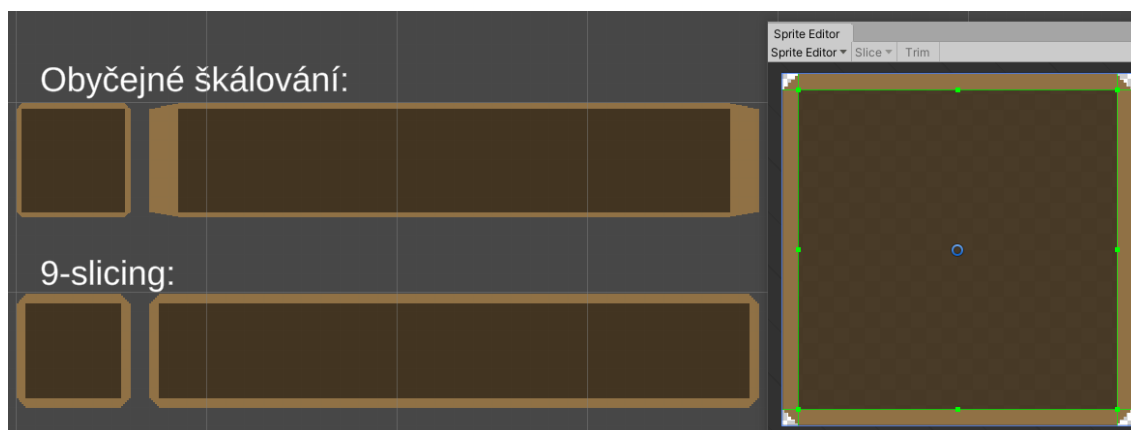
2.2 2D tvorba

Unity vzniklo původně jako engine pro vývoj 3D her a nástroje pro 2D tvorbu byly přidány až v pozdějších verzích. Volba mezi 2D a 3D je prvním krokem při zakládání nového projektu, rozdíl mezi nimi ale není velký a oba přístupy lze během vývoje kombinovat. Hlavním viditelným rozdílem pro 2D projekty je editor scény, který je přepnut do dvourozměrného režimu. V tomto módu je změněn význam osy Z, která místo popisu třetího rozměru značí hloubku (podobně jako např. vlastnost z-index v CSS). [3]

2.2.1 Sprite

Sprite je jeden z nejběžnějších assetů ve 2D videohře. V Unity vzniká přetažením bitmapového obrázku do průzkumníka projektu (obvykle do složky *Sprites*) a má několik vlastností, které lze změnit v inspektoru. Za zmínku stojí hodnota *Pixels Per Unit*, která určuje, kolik pixelů ve spritu odpovídá jednotce vzdálenosti v herním světě (ve scéně). Druhý důležitý parametr je *Pivot*, ten značí bod, kolem kterého sprite rotuje, a také může ovlivnit pořadí vykreslení daného spritu.

Každý sprite nemusí být reprezentován samostatným bitmapovým souborem, místo toho lze naimportovat tzv. spritesheet. Jedná se o větší obrázek obsahující více spritů uspořádaných do mřížky, ty jsou pak rozděleny na jednotlivé assety v nástroji editor spritů. Tento nástroj také umožňuje připravit obrázek pro specifický typ škálování zvaný 9-slicing (obrázek 2.2).



Obrázek 2.2: 9-slicing

2.2.2 Animace

Unity umožňuje tvorbu animací po jednotlivých snímcích, kdy je každý snímek animace reprezentován samostatným spritem. Jednotlivé sprity jsou pokládány na časovou osu v nástroji *Animation*. Nevýhoda této metody se projeví v situaci, kdy je potřeba např. nějaké postavě změnit barvu vlasů. V tomto případě je nutné všechny sprity všech animací postavy upravit v (externím) grafickém editoru. Problém by šel také vyřešit tvorbou speciálního shaderu nebo editací jednotlivých pixelů ze skriptu, nejedná se ale o standardní postupy. Alternativně lze použít metodu trojúhelníkové sítě a kostry. Jedná se o běžný způsob tvorby 3D animací, Unity jej ale umožňuje aplikovat i na 2D sprity.

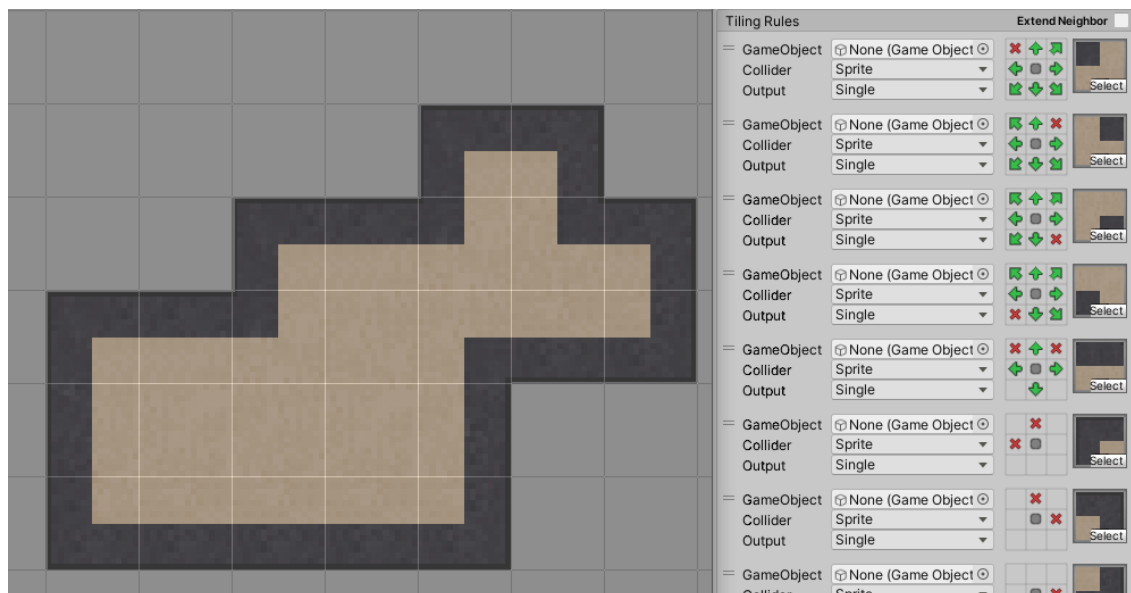
Každý herní objekt schopný animace musí mít přiřazený asset typu *Animator Controller* (dále jen AC). Ten je vytvořen a upravován přímo v Unity editoru a jeho diagram připomíná schéma stavového automatu. Všechny animace, ve kterých se herní objekt může nacházet, jsou v diagramu reprezentovány obdélníkem. Obdélníky lze spojovat šipkami, které značí podmíněný přechod mezi jednotlivými animacemi. AC umožňuje definovat si v něm interní proměnné (*parameters*), jejichž hodnoty lze měnit ze skriptu a využívají se pro sestavení podmínek přechodu. AC diagramy složitějších herních objektů mohou značně nabýt na složitosti, je možné se jim ale vyhnout. Volání funkce *Animator.CrossFade* dovoluje změnit animaci kdykoliv bez ohledu na stav AC diagramu.

2.2.3 Tilemap

Tilemap je nástroj pro tvorbu herních světů převážně ve 2D. Základem je pravidelná mřížka, do které jsou vkládány dlaždice (*tiles*). Dlaždice je malý sprite nakreslený tak, aby více dlaždic položených vedle sebe vypadalo přirozeně a dala se z nich takto poskládat celá herní mapa. Využitím tilemap lze tvořit velké herní mapy bez dopadu na výkon a zároveň je kdykoliv snadno editovat. Unity nabízí obdélníkovou tilemap (nejběžnější, vhodné pro plošinovky a top-down videohry), ale také šestiúhelníkovou nebo izometrickou. V jedné scéně se může nacházet více tilemap a každá může být

v jiné vrstvě. To např. umožňuje, aby se některé dlaždice vykreslovaly před a některé za postavou hráče.

Rule Tile je speciální druh dlaždice, která mění svůj vzhled na základě jednoduchých nastavitelných pravidel. Sprite této dlaždice se mění podle toho, zdali se v jejím okolí nachází dlaždice stejného druhu. Funkcionalita je ale omezená, nelze např. nastavit, aby dlaždice reagovala na své sousedy z jiných vrstev nebo na jiné dlaždice ve stejné vrstvě.



Obrázek 2.3: Rule Tile

2.2.4 Universal Render Pipeline

Universal Render Pipeline (URP) je skriptovatelný vykreslovací řetězec, který je do Unity projektu přidán ve formě balíku. Otázka použití alternativního vykreslovacího řetězce bývá častější u vývoje 3D her, i pro 2D projekty ale URP přináší nové grafické možnosti. Tím nejzásadnějším jsou dvě nové komponenty *Light 2D* a *Shadow Caster 2D*, díky kterým lze do scény přidat dynamická světla a stíny.

3 Návrh videohry

3.1 Příběh

Navrhovaná hra tohoto projektu uvádí hráče do role maturanta, který se chystá na FM podat přihlášku ke studiu. Po vstupu na fakultu se za ním zavřou a uzamknou dveře. Novým cílem je tedy zjistit, co se děje. Posléze je hráč kontaktován osobou, která ho informuje, že celou budovu fakulty má pod kontrolou místní umělá inteligence, jež se vymkla kontrole.

Hlavním cílem hry je zastavit umělou inteligenci a uniknout z fakulty. V budově se nachází nepřátelští roboti, kteří se v tom hráči pokusí zabránit. Zhruba v druhé polovině příběhu se hráč dozvídá, že osoba, jež ho kontaktovala, byla ve skutečnosti samotná umělá inteligence, která ho tímto využívala. Příběh je inspirován hrou System Shock 2 (1999, Irrational Games).

3.2 Žánr

Obecně by se hra dala kategorizovat jako akční a nabízí dva základní styly hraní. Tím prvním je boj na blízko. Hráč má k dispozici zbraň, konkrétně se jedná o hasák, se kterou může útočit na nepřátele. Ti mohou stejným způsobem zase útočit na hráče. Po každém úspěšném zásahu je cíli útoku ubráno jeho zdraví, které je v případě této hry reprezentováno celým číslem, a jakmile se dostane na nulu, tak je cíl zneškodněn.

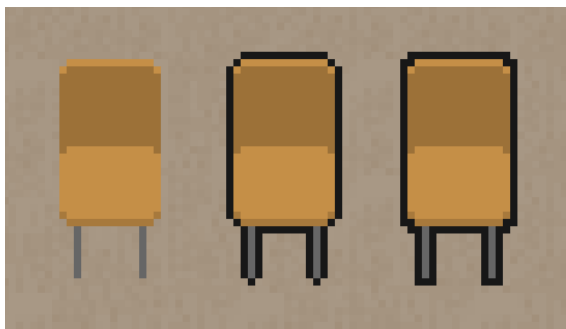
Druhou možností pro hráče je vyhnout se přímé konfrontaci s nepřátele a namísto toho se před nimi skrývat. Žánr, kde se hráč snaží, aby nebyl odhalen nepřátelskými jednotkami, se ve videoherní terminologii nazývá stealth game. Slovem stealth se označuje i hlavní mechanika tohoto žánru – plížení, které hráči pomáhá nebýt spatřen.

3.3 Vizuální stránka

Hra je zobrazována pohledem kamery, která na mapu pohlíží svrchu. Tato perspektiva bývá obecně nazývána jako top-down. V případě tohoto projektu není kamera namířena přímo kolmo na plochu mapy, ale sleduje herní svět pod úhlem přibližně 45 stupňů. Pohled pod určitým úhlem odhaluje hloubku vyobrazovaných objektů a přesněji by se dal nazvat jako axonometrická projekce. [4]

Grafika hry je nakreslena ve stylu pixel art. Jedná se o druh digitálního grafického umění, kde každý pixel ve výsledném obraze hraje svou roli a změna jediného pixelu může značně ovlivnit, jak obraz na pozorovatele působí. Počátky pixel artu sahají do dob 8 a 16bitových konzolí, kde byla omezená paleta barev a rozlišení zobrazovaných spritů. Svou důležitost si ale pixel art zachoval dodnes. Je oblíbený zejména mezi nezávislými vývojáři kvůli jednoduchosti a rychlosti tvorby. [5]

V grafickém stylu této hry není dodrženo pravidlo užití omezené palety barev. Sprite postavy hráče má rozměry 16 na 32 pixelů a od toho jsou přibližně odvozeny velikosti všech ostatních spritů. Kromě dlaždic pro tilemap mají všechny sprity přidán tmavý obrys o šířce jednoho pixelu, aby vynikly a nesplývaly s pozadím.



Obrázek 3.1: Druhy obrysů u spritu

3.4 Herní mechaniky

3.4.1 Pohyb

Postava hráče se může pohybovat v osmi směrech pomocí kláves W, A, S a D. Stiskem mezerníku se provede rychlý úskok (dash), během kterého je postava nezranitelná. Dash ubírá hráčovu staminu, která se časem regeneruje na svou původní hodnotu, a nelze ho provést, pokud není staminu dostatek.

Aby hráč nenavštěvoval jednotlivé lokace na herní mapě v pořadí, které by porušilo konzistenci příběhu, je jeho pohyb omezen pomocí uzamčených dveří. Aby tyto dveře mohly být otevřeny, musí nejprve hráč nalézt klíč v barvě odpovídající zámku na daných dveřích.

Stiskem klávesy Ctrl se postava hráče skrčí a přepne do režimu plížení. V tomto módu se pohybuje pomaleji, ale může pro přesun využít větracích šachet rozmístěných po mapě. Hráč ve větrací šachtě nemůže být spatřen nepřítelem. To samé platí, pokud je v režimu plížení hráč schovaný za nějakým objektem (např. za stolem).

3.4.2 Boj

Stiskem levého tlačítka myši (LMB) se provede útok na blízko, ten si bere malou část staminu a na chvíli pozastavuje její regeneraci. Útok má také svou vlastní dobíjecí dobu, aby nemohl být opakován jeden za druhým bez jakékoliv pauzy (attack

spamming). Po vypršení dobíjecí doby následuje krátké časové okénko, během kterého stisknutí LMB provede útok s kritickým zásahem, který ubírá více bodů zdraví. Pokud naopak hráč stiskne LMB během dobíjecí doby, již není možné provést útok s kritickým zásahem a dobíjecí doba se o toto časové okénko prodlouží.

Hráč má svou hladinu zdraví, která se, pokud nebyl po určitou dobu zraněn nepřítelem, regeneruje do poloviny své maximální kapacity. Pokud hráčovo zdraví klesne na nulu, neznámá to konec hry. Namísto toho je hráč oživen (respawn) ve vestibulu budovy s tím, že jeho aktuální úkol, sebrané předměty a zneškodnění nepřátelé nejsou resetovány.

3.4.3 Grafické uživatelské rozhraní

3.5 Umělá inteligence

4 Implementace

5 Kritické zhodnocení

6 Závěr

Seznam použité literatury

1. UNITY TECHNOLOGIES. *Unity - Manual: Asset workflow* [online]. 2023-04. [cit. 2023-04-26]. Dostupné z: <https://docs.unity3d.com/2022.2/Documentation/Manual/AssetWorkflow.html>.
2. UNITY TECHNOLOGIES. *Unity - Manual: Important Classes - GameObject* [online]. 2023-04. [cit. 2023-04-26]. Dostupné z: <https://docs.unity3d.com/2022.2/Documentation/Manual/AssetWorkflow.html>.
3. GOLDSTONE, Will. *Unity 4.3: 2D Game Development Overview* [online]. 2013-11. [cit. 2023-04-26]. Dostupné z: <https://blog.unity.com/technology/unity-4-3-2d-game-development-overview>.
4. JAN, Matej. *Game developer's guide to graphical projections (with video game examples)* [online]. 2017-11. [cit. 2023-05-05]. Dostupné z: <https://medium.com/retronator-magazine/game-developers-guide-to-graphical-projections-with-video-game-examples-part-1-introduction-aa3d051c137d>.
5. SAMUELSON, Gustav. *Pixel art - The Medium of Limitation* [online]. 2020. [cit. 2023-05-05]. Dostupné z: <http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1518210&dswid=-4613>.