



## Projekt

# Vývoj akční videohry zasazené do prostředí FM

*Studijní program:*

*Autor práce:*

*Vedoucí práce:*

B0613A140005 – Informační technologie

**Radek Mocek**

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.

28. 4. 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

<b>1</b>	<b>Úvod</b>	<b>7</b>
<b>2</b>	<b>Herní engine Unity</b>	<b>8</b>
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
<b>3</b>	<b>Návrh videohry</b>	<b>12</b>
<b>4</b>	<b>Návrh umělé inteligence</b>	<b>13</b>
<b>5</b>	<b>Implementace</b>	<b>14</b>
<b>6</b>	<b>Kritické zhodnocení</b>	<b>15</b>
<b>7</b>	<b>Závěr</b>	<b>16</b>
	<b>Seznam použité literatury</b>	<b>17</b>

## Seznam obrázků

2.1	Unity editor . . . . .	9
2.2	9-slicing . . . . .	10
2.3	Rule Tile . . . . .	11

# 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.

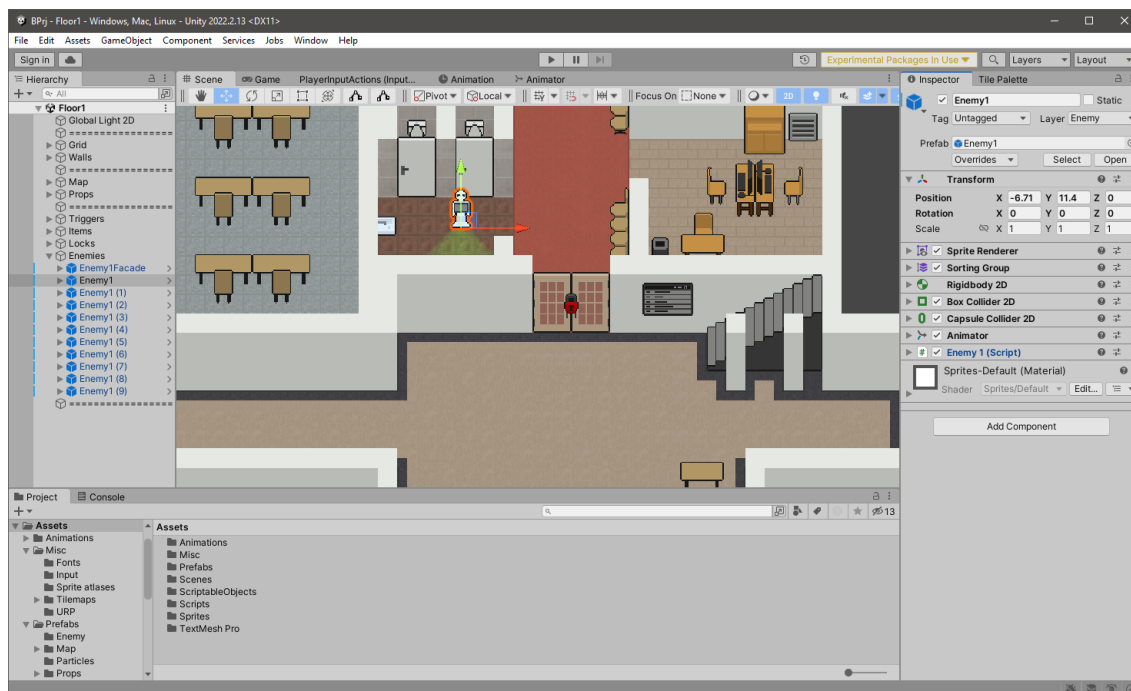
### 2.1 Editor a základní stavební prvky

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 *sprite* 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

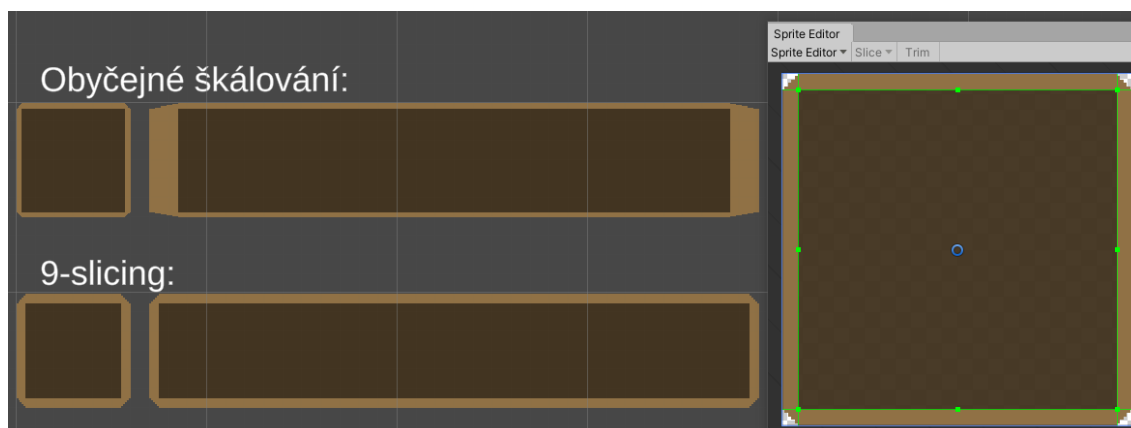
## 2.2 2D tvorba

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.

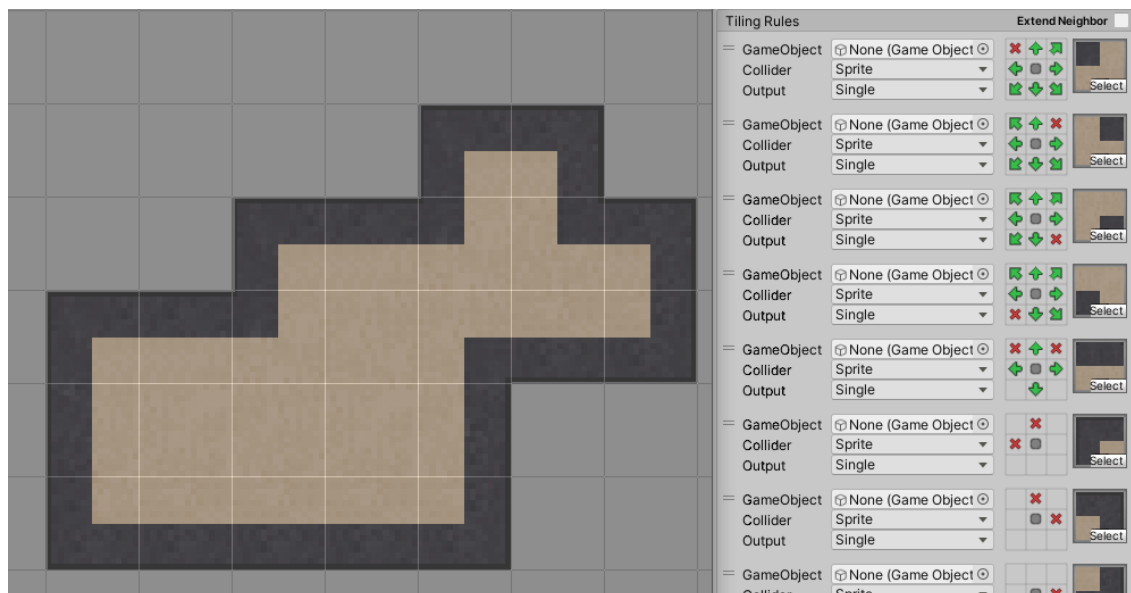
Každý herní objekt schopný animace musí mít přiřazený asset typu *Animator Controller* (dále jen AC). Ten je vytvořen a upravován přímo v Unity editoru a jeho diagram připomíná schéma stavového automatu. Všechny animace, ve kterých se herní objekt může nacházet, jsou v diagramu reprezentovány obdélníkem. 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

## 4 Návrh umělé inteligence

## 5 Implementace

## 6 Kritické zhodnocení

## 7 Závěr



## Seznam použité literatury

1. TECHNOLOGIES, Unity. *Unity - Manual: Asset workflow* [online]. 2023-04. [cit. 2023-04-26]. Dostupné z: <https://docs.unity3d.com/2022.2/Documentation/Manual/AssetWorkflow.html>.
2. TECHNOLOGIES, Unity. *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 / Unity Blog* [online]. 2013-11. [cit. 2023-04-26]. Dostupné z: <https://blog.unity.com/technology/unity-4-3-2d-game-development-overview>.