

Střední průmyslová škola Třebíč

Maturitní práce

Webová aplikace pro tvorbu 3D map

Profilová část maturitní zkoušky

Studijní obor: Informační technologie

Třída: ITA4

Školní rok: 2024/2025 Viktor Čada

Zadání práce

Cílem této ročníkové práce je vytvořit webovou aplikaci, která umožní procedurálně generovat místnosti pomocí herního enginu Unity a programovacího jazyka C#. Rozsah, složitost, počet místností a pater těchto vytvořených místnosti bude záviset na uživatelem zadaných parametrech a seedů.

Vytvořené mapy bude možné po dokončení prozkoumat přímo v aplikaci nebo uložit do vlastního souborového formátu (.dnd). Tento formát bude využívat technologie podobné formátu XML.

ABSTRAKT

Cílem teto práce je tvorba a popis tvorby webové aplikace pro procedurální

generování map pro TTRPG hry, jako je Dungeons & Dragons, za využití herního

enginu Unity a programovacího jazyka C#. Aplikace umožní uživatelům zadávat

parametry, jako je počet pater, frekvence a typy dekorativních prvků, přičemž

klíčovým faktorem pro generování map bude uživatelem zadaný seed. Procedurálně

vytvořené místnosti bude možné prozkoumat přímo v aplikaci nebo exportovat do

souboru ve vlastním formátu “.dnd”, který bude využívat technologie podobné XML.

Pro agilní plánování projektu bude použita metoda Kanban, přičemž správa verzí a

dokumentace bude řešena prostřednictvím GitHubu.

KLÍČOVÁ SLOVA

Unity, C#, GameObject, Seed, Procedurálně generováno, D&D

ABSTRACT

The aim of this thesis is to design and describe the creation of a web application for procedural map generation for TTRPG games such as Dungeons &amp; Dragons, using the Unity game engine and the C# programming language. The application will allow users to specify parameters such as the number of floors, frequency and types of decorative elements, with the user-specified seed being the key factor for the map generation. Procedurally created rooms will be able to be explored directly in the application or exported to a custom ”.dnd” file using XML-like technologies. Kanban will be used for agile project planning, with version control and documentation handled through GitHub.

KEYWORDS

Unity, C#, GameObject, Seed, Procedurally Generated, D&D

PODĚKOVÁNÍ

Děkuji Mgr. Petru Novotnému za cenné připomínky a rady, které mi poskytl při vypracování maturitní práce.

V Třebíči dne 20. února 2025 podpis autora

PROHLÁŠENÍ

Prohlašuji, že jsem tuto práci vypracoval/a samostatně a uvedl/a v ní všechny prameny, literaturu a ostatní zdroje, které jsem použil/a.

V Třebíči dne 20. února 2025

podpis autora

Obsah

[Úvod 6](#_Toc188485483)

[1 Použité Technologie 7](#_Toc188485484)

[1.1 Unity 7](#_Toc188485485)

[1.1.1 Object 7](#_Toc188485486)

[1.1.2 GameObject 7](#_Toc188485487)

[1.1.3 MonoBehaviour 7](#_Toc188485488)

[1.1.4 ScriptableObject 8](#_Toc188485489)

[1.1.5 Transform 8](#_Toc188485490)

[1.1.6 TextMesh Pro 8](#_Toc188485491)

[1.2 C# 8](#_Toc188485492)

[1.2.1 Historie 9](#_Toc188485493)

[1.3 Blender 9](#_Toc188485494)

[1.4 Git 9](#_Toc188485495)

[2 Vlastní souborový formát 10](#_Toc188485496)

[2.1 XML 10](#_Toc188485497)

[2.2 DND 10](#_Toc188485498)

[2.2.1 Struktura 10](#_Toc188485499)

[2.2.2 Head 10](#_Toc188485500)

[2.2.3 Body 10](#_Toc188485501)

[3 Table Top Role-Playing Games 11](#_Toc188485502)

[3.1 Historie 11](#_Toc188485503)

[3.2 Verze 11](#_Toc188485504)

[3.3 Mechaniky 11](#_Toc188485505)

[Závěr 13](#_Toc188485506)

[Seznam použitých zdrojů 14](#_Toc188485507)

[Seznam použitých symbolů a zkratek 15](#_Toc188485508)

[Seznam obrázků 16](#_Toc188485509)

[Seznam tabulek 17](#_Toc188485510)

[Seznam příloh 18](#_Toc188485511)

Úvod

Cílem této práce je tvorba webové aplikace pro vytváření map pro TTRPG hry typu Dungeons & Dragons. Aplikace bude využívat herní engine Unity a programovací jazyk C#, hlavně třídu GameObject. Pro agilní plánování projektu bude použita metoda Kanban na platformě Freelo. Správa úložišť, dokumentace a verzí bude zajištěna pomocí systému git, hostovaného na GitHubu.

Aplikace bude procedurálně generovat na základě uživatelem zadaných parametrů. Nejvýznamnějším parametrem bude unikátní seed, který bude základem pro generování mapy. Mezi další parametry bude patřit počet pater, četnost a typy dekorativních předmětů, s možností vybrat, které se mají zobrazit a které ne. Vytvořené procedurální místnosti bude možné prozkoumat přímo v aplikaci nebo exportovat do vlastního souborového formátu s příponou „.dnd“. Tento formát bude využívat technologie podobné XML a SVG pro snadné zpracování a čitelnost.

# Použité Technologie

## Unity

Unity je zevrubný herní engine, který umožňuje vytvářet 2D i 3D hry a interaktivní aplikace. Je známý svou flexibilitou, snadným použitím a širokou podporou platforem. Zakladatel společnosti stojící za Unity ji nazval: „sada nástrojů pro tvorbu her. Je to technologie, která zajišťuje grafiku, zvuk, fyziku, interakce, propojení do sítě.“ [1].

Umožňuje programovaní v jazyce C#, což je efektivní a široce používaný jazyk v herním vývoji. K dispozici jsou také různé nástroje a komponenty pro práci s audiem, síťovými funkcemi, vstupy a dalšími herními prvky.

### Object

Object je nejzákladnější třída pro funkčnost a vývoj v Unity. Tato třída by se sama o sobě neměla používat přímo v kódu, protože neobsahuje tolik užitečných metod a vlastností jako její potomci. Nejvýznamnějšími potomky jsou GameObject, MonoBehaviour a ScriptableObject [2].

Třída Object dále obsahuje metody Instantiate a Destroy, které slouží k vytváření nebo mazaní instancí GameObjectů ve scéně [2].

### GameObject

GameObject je nejzákladnější třída v Unity pro vývojáře a game developery. Vše, co se nachází ve scéně Unity musí dědit z této třídy. Poskytuje funkcionality jako: hledání a vytváření odkazů a spojení, zasílání zpráv a proměnných mezi GameObjecty, přidávání, modifikaci a odstraňování komponent a nastavení parametrů a konfigurací [3].

Každý GameObject obsahuje proměnné: název, tag, vrstva, stav a transform. Tyto jednoduché proměnné se využívají k nalezení správné instance ve scéně [3].

### MonoBehaviour

Třída MonoBehaviour nám umožňuje připojit skript ke GameObjectu, takže všechny skripty od ní automaticky dědí. Přidává nám události jako Awake, Update a FixedUpdate. Awake se spustí pouze jednou při vytvoření instance, většinou na začátku scény. Update se spouští každý snímek. FixedUpdate se spouští při každém fyzikálním přepočtu, který je standardně 50krát za sekundu [4].

### ScriptableObject

ScriptableObject se používá při ukládání velkých objemů dat do paměti nebo na disk a přenosu dat mezi scénami. Pokud vytváříme několik instancí předem vytvořených GameObjectů, tzv. prefabů, tak ScriptableObject ukládá pouze data, která se liší od výchozích hodnot. Není možné jej přímo připojit ke GameObjectu, ale musí být uložen jako samostatný soubor, tzv. **Asset** [5].

### Transform

Transform je komponenta, která je automaticky přidána ke každému GameObjectu. Poskytuje informace o jeho pozici, rotaci a měřítku. Dále obsahuje metody a proměnné pro jednoduché hledání potomků a předků v hierarchii objektů [6].

Existuje alternativa jménem RectTransform, která se hlavně používá pro UI. Lokace v scéně se přes ní určuje pomocí šířky, výšky a pozice obdélníku [7].

### TextMesh Pro

TextMesh Pro není základní třídou Unity, ale rozšiřuje možnosti formátování a práce s textem. Přidává, nebo vylepšuje, také další UI komponenty, jako jsou tlačítka, textová pole, slidery, checkboxy a toggles, které jsou jednodušší na ovládání a optimalizovanější než základní komponenty Unity. Díky tomu je práce s textem a UI v Unity rychlejší a efektivnější [8].

## C#

C# je moderní, objektově orientovaný programovací jazyk [3]. Je známý svou jednoduchostí, efektivitou a širokou škálou použití. „Jazyk C# je odpovědí společnosti Microsoft na jazyk Java a byl vytvořen s cílem být víceúčelovým programovacím jazykem pro všeobecné použití.“ [9]

Jazyk je bohatý na funkce, jako jsou lambda výrazy, LINQ dotazy a asynchronní programování, což umožňuje vývojářům psát čistý a efektivní kód. Díky robustní knihovně .NET je C# ideálním nástrojem pro tvorbu multiplatformních aplikací, které běží na různých operačních systémech [10].

### Historie

C# vyšel v roce 2002 a od té doby je průběžně aktualizován. Každé 2 roky vychází nová verze. Mezi koncepty, které byly přidány později, patří například lambda výrazy, asynchronní programování a iterátory. Poslední verze rozšířila možnosti práce s parametry a asynchronním programováním [9].

## Blender

Blender je open-source software, který slouží k řešení komplexních i jednodušších úloh při tvorbě 3D modelů. Umožňuje pracovat s topologií, UV unwrappingem, texturami a vytvářením vlastních materiálů. Tento nástroj je ideální pro modelování 3D objektů a texturování, ale také nabízí možnosti pro animaci a rendering. Je oblíbený jak mezi začátečníky, tak profesionály díky své široké škále funkcí [11].

## Git

Git je verzovací systém vyvinutý Linusem Torvaldsem v roce 2005 [12] při vývoji operačního systému Linux. Git umožňuje sledovat změny v kódu, vracet se k předchozím verzím a spolupracovat s více vývojáři na jednom projektu současně. Tento systém byl vytvořen jako alternativa k předchozím verzovacím nástrojům, které nebyly dostatečně flexibilní pro potřeby open-source vývoje, a od svého vzniku se stal nejrozšířenějším nástrojem pro správu verzí v softwarovém vývoji.

# Vlastní souborový formát

## XML

XML je formát souboru zapisovaný v plaintextu. Tento formát byl navržen tak, aby byl čitelný jak pro lidi, tak pro stroje. XML je značkovací jazyk, který využívá tzv. tágy k označení struktur a obsahu dat. Tyto tágy nejsou předefinované, na rozdíl od HTML. XML slouží k přenosu a ukládaní dat. [13]

XML soubory obvykle začínají deklarací verze a kódování. Následně obsahují hierarchickou strukturu složenou z otevíracích a uzavíracích tágu a atributu.

## DND

Stejně jako jeho „předchůdce“ XML je i formát .dnd značkovací jazyk využívající tágy. Název tohoto souborového formátu je inspirován hrou Dungeons & Dragons, pro kterou je tento projekt určen.

Formát .dnd byl navržen specificky pro potřeby tohoto projektu, aby umožňoval snadnou úpravu herního obsahu pomocí jednoduché syntaxe.

### Struktura

Na rozdíl od XML souboru používá DND strukturu hlavy (head) a těla (body). Výhodou této struktury je oddělení metadat od dat. To nám zajišťuje lepší přehlednost a organizaci souboru. Tato struktura také umožňuje snadnou rozšiřitelnost.

### Head

V části head DND souboru se nachází všechny parametry určené pro celý soubor, jako je pozice hráče, seed generátoru, verze programu a další uživatelem zadané proměnné.

### Body

V části body jsou uloženy informace o herním prostředí, především o místnostech, jejich velikosti, pozici a stavu použití vygenerovaných prvků.

# Table Top Role-Playing Games

Table Top Role-Playing Games, dále jen TTRPG, neboli stolní hry na hrdiny, je žánr fantasy a sci-fi her. Hráči se v nich vžívají do rolí hrdinů, kteří tvoří družinu dobrodruhu ve fantaskním světě tvořeným jedním z nich. Tento speciální hráč, většinou přezdívaný Pán Jeskyně nebo Dungeon Master, dále jen PJ, tyto příběhy, postavy a někdy i cele světy připravuje sám. To může být náročné, tudíž existuje hromada online i off-line nástrojů, které s některými repetetivními částmi pomáhají. Tento projekt se řadí mezi ne [14].

## Historie

“Kdysi velmi, velmi dávno v říši zvané Středozápad Spojených států, konkrétně ve státech Minnesota a Wisconsin, se scházela skupina přátel, aby navždy změnila herní historii” [14] píše Mike Mearls, vedoucí návrhu D&D. Hlavními tvůrčími osobnostmi byli přátelé Mike Mearls a Jeremy Crawford, kteří byli frustrovaní ne-interaktivními příběhy z knih a filmu. Tudíž se rozhodli vytvořit si novy žánr her, kde se nikdy nestane to stejně dvakrát.

“Dungeons & Dragons odstartovalo bující celosvětový fenomén. Je to první hra na hrdiny a zůstává nejlepší svého druhu” [14] píše Mearls.

## Verze

DnD je pouze jedna z variant tohoto populárního žánru. DnD samotné má 5 edicí, kdy každá má svoji oddanou komunitu. Pro vice komplexní fantasy zážitek existuje Pathfinder a jeho edice [15] .Neexistují pouze fantasy TTRPG hry, velmi populární jsou sci-fi verze jako Traveler, Starfinder nebo Cyberpunk. Hororové témata jsou taktéž hojná jako třeba Mörk Borg nebo Call of Cthulhu [16].

## Mechaniky

Většina stolních her na hrdiny má svá komplexní pravidla a mechaniky, ale vše je založeno na 20 stěnné kostce [14]. Tato kostka a jine velkosti určují úspěch nebo neúspěch při různých akcích, jako je boj, hádanky nebo vyjednávání. Hráči mají volnou ruku nad prozkoumáváním světa, kde hledají magické a jinak užitečné předměty.

Každý systém má vlastní pravidla, která určují míru realismu a hloubku hry. Například Dungeons & Dragons klade důraz na hrdinský boj, Mörk Borg se zaměřuje na smrt a pokus o přežití v apokalyptickým světě.

# Scény

Aplikace je rozdělena do 3 základních scén, které fungují nezávisle na sobě a každá plní specifickou funkci v rámci aplikace.

## MainMenuScene

MainMenuScene slouží jako úvodní obrazovka aplikace, kde uživatel vybírá různé možnosti, jako je začátek nové hry nebo základní nastaveni hry.

## GeneratingScene

V GeneratingScene se zpracovávají vstupy od uživatele, které jsou následně uloženy do odpovídajících datových struktur. V této scéně probíhá samotné procedurální generování, kde se na základě uživatelem zadaných parametrů vytváří struktura mapy a další herní elementy.

## LevelScene

LevelScene je dynamicky generovaná scéna, která při kompilaci neobsahuje žádný obsah. Až při spuštění aplikace se z uložených dat (DNDFileData) vytvoří konkrétní úroveň (level) podle definovaných pravidel a parametrů. Tato scéna se tedy přizpůsobuje aktuálním potřebám uživatele.

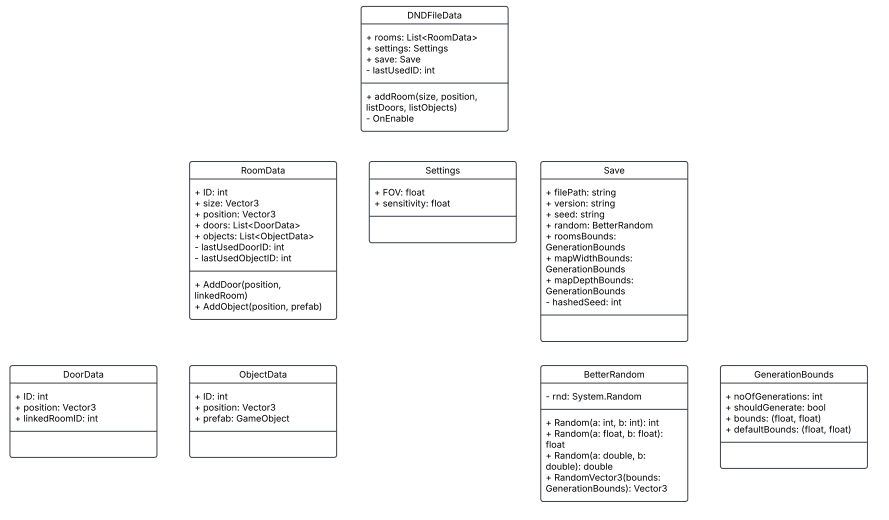
# Důležité Skripty

Tato aplikace má několik skriptu a tato část ukáže ty nejdůležitější, které stoji za hlavni funkčností a generováním.

## UIScript

**UIScript** zastřešuje všechno UI v projektu. V **GeneratingScene** dynamicky zapíná a vypíná potřebné UI elementy podle uživatele a jeho dat. V samotné **LevelScene** dynamicky ovládá menu a nastavení v nich. Propojuje menu nastavení s hodnotami ve scéně, jako je **FOV** nebo **senzitivita**. V tomto skriptu je umístěno i ovládaní Input map, které jsou specifické pro menu a herní level.

## DNDFileData

**DNDFileData.cs** obsahuje několik pomocných tříd na ukládání dat a jednoduchou práci s nimi. Abstraktní třída **BaseEntityData** zastřešuje všechny ukládané objekty do scény. Dědí od ní **RoomData**, **DoorData** a **ObjectData**, přičemž každá je specializovaná na jiný druh objektu. Třídy **Settings** a **Save** slouží pro ukládání dat zadaných uživatelem při generování místností. Třídy **GenerationBounds** a **Bounds** využívají parametr obecného typu. To nám pomáhá určovat typ při kompilaci. Slouží k ukládání rozsahu pro generování náhodných čísel.

Třída **BetterRandom** je založená na třídě Random z namespace System, ale je rozšířená o funkce Random a RandomVector3. Random používá jako parametry a návratovou hodnotu parametr obecného typu. RandomVector3 je přetížená pro 2D a 3D. 2D se využívá u procedurálního generování místností. 3D se používá pro objekty.

## Creator

Creator je prefab, který ovládá generovaní a převádění dat do souboru a datových struktur. Obsahuje dvě třídy: DNDFileScriptCreator a DNDSceneScriptCreator.

### DNDFileScriptCreator

**DNDFileScriptCreator** obsahuje dvě hlavní části, a to metodu PrepareSave a CreateFile s pomocnými metodami.

**PrepareSave** stojí za samotnou generaci dat. Využívá algoritmus vzorkování s odmítáním na generovani mistnostni a dále multiple spanning tree da genrovani propoju mezi mistnosti a dvere

**CreateFile** používá třídu XMLWriter na zapisování do souboru. Metody WriteGenerationBounds, WriteBounds a WriteVector3 odstraňují duplicitní kód při psaní opakovaných datových struktur.

V tomto souboru je nadefinovaná pomocná třída **RectangleF**, která se používá při generování místností a zjišťování, jestli se překrývají, pomocí metody Overlaps.

### DNDSceneScriptCreator

**DNDSceneScriptCreator** obsahuje metody na načítání dat, převádění souborů do datové struktury a vytvoření scény samotné.

Metoda **ParseDNDFile** převede **.dnd** soubor zpět do datové struktury, aby se s ní dalo jednoduše manipulovat.

**MakeMap** a pomocné metody **InstantiateRoom** a **InstantiateDoor** převedou datovou strukturu třídy **DNDFileData** do scény. Využívají Unity metodu **Instantiate** s přetížením pro rodiče, aby se zachovala logická hierarchie.

**Load** metody využívají třídu **Resources** a načítají modely a prefaby ze specifikovaných složek. Tato načtená data ukládají do slovníku, aby se jednoduše a intuitivně používala v jiných metodách.

# Procedurální Generace

Procedurální generování je klíčovým mechanismem aplikace, který umožňuje dynamické a náhodné vytváření herních úrovní a map. Tento proces využívá algoritmy a matematické funkce k vytvoření herních prvků, jako jsou terény, objekty a struktury, na základě parametrů zadaných uživatelem.

Generování začíná příjmem dat od uživatele v GeneratingScene, kde se definují požadované vlastnosti mapy. Na základě těchto parametrů se pak provádí výpočet a generování prostředí, což zahrnuje jak náhodné, tak i určité predikabilní prvky pro zajištění vyváženosti.

Jedním z nejznámějších příkladů použití procedurálního generování je hra Minecraft, která využívá techniku Perlinova šumu k vytváření realistických a přírodních terénů. Perlinův šum je matematická funkce, která generuje plynulé a přirozeně vypadající vzory, což se ideálně hodí pro generování hor, řek, lesů a dalších přírodních prvků. Tento typ šumu je základem pro vytváření "náhodného" světa, který vypadá konzistentně a realisticky, i když je generován automaticky.

Tento přístup umožňuje generování neomezeného množství unikátních map, což zvyšuje opakovatelnost a herní hodnotu. Tento proces také šetří prostor v paměti, protože není potřeba uchovávat celou mapu – místo toho se vždy generuje nová na základě aktuálních podmínek.

Procedurální generování může být aplikováno na různé aspekty hry, od terénu, přes umístění objektů až po chování NPC postav, což dává hráčům novou zkušenost s každým hraním.

# Klíčové Algoritmy

## Vzorkování s odmítáním

Vzorkování s odmítáním, anglicky **Rejection Sampling**, je algoritmus na procedurální generovaní platných dat. Funguje tak, že generuje náhodné hodnoty (např. polohy místností) a ty, které nesplňují podmínky (např. překrývají se s jinými místnostmi), odmítne a zkouší to znovu. Tento proces opakuje, dokud nenajde vhodnou hodnotu

Aplikace tento algoritmus používá k generovaní místnosti do DNDFileData.

## Multiple Spanning Tree

Multiple Spanning Tree (MST) je algoritmus využívaný pro generování grafových struktur s více kostrovými stromy (spanning trees). Tento přístup je obzvláště užitečný při tvorbě nelineárních propojených systémů, například cest mezi místnostmi v procedurálně generovaných mapách.

Nejprve se vytvoří úplný nebo částečně propojený graf, kde uzly představují místnosti a hrany možné spojnice mezi nimi. Použije se standardní algoritmus pro hledání minimální kostry (např. Kruskalův nebo Primův algoritmus), čímž se vytvoří propojený strom bez cyklů. Tento strom zajišťuje, že každá místnost je dosažitelná. Pro lepší propojení a větší variabilitu se přidávají další hrany podle různých pravidel (např. náhodně, podle vzdálenosti nebo podle určitého vzoru), čímž se vytvářejí alternativní cesty a cykly. Výstupem je graf s několika vzájemně propojenými komponentami, které umožňují větší rozmanitost možných tras a cest mezi místnostmi.

# Port na web

## WebGL

V této části bude projekt využívat WebGL, což je technologie pro vykreslování 3D grafiky přímo v prohlížeči bez nutnosti instalace jakýchkoli doplňků. WebGL je založen na OpenGL ES 2.0 a umožňuje vykreslovat 3D objekty a animace ve webových prohlížečích, což znamená, že aplikace bude přístupná přímo přes internet bez potřeby externího softwaru. Tento přístup je ideální pro interaktivní aplikace a hry, které mají běžet na široké škále zařízení.

## Hostovaní

Pro tento projekt bude použita platforma itch.io, která umožňuje vývojářům snadno hostovat a distribuovat své hry přímo na internetu. itch.io podporuje nahrávání her v různých formátech, včetně HTML5 her, které využívají WebGL pro grafiku a interaktivitu. Projekt bude tedy exportován jako HTML5 aplikace, což zaručí kompatibilitu napříč prohlížeči bez potřeby stahování klienta.

Po exportu bude aplikace nahrána na itch.io, kde ji budou moci uživatelé jednoduše přistupovat a hrát přímo v prohlížeči. itch.io také nabízí nástroje pro nastavení verzí hry, což umožňuje jednoduše aktualizovat projekt bez nutnosti měnit URL.

Závěr

Seznam použitých zdrojů

1. HAAS, John. *A History of the Unity Game Engine*. Online, ‎. Worcester: WORCESTER POLYTECHNIC INSTITUTE, 2014. Dostupné z: <http://www.daelab.cn/wp-content/uploads/2023/09/A_History_of_the_Unity_Game_Engine.pdf>. [cit. 2025-01-14].
2. UNITY TECHNOLOGIES. *Fundamental Unity types*. Online. UNITY TECHNOLOGIES. Unity 6 User Manual. 2025. Dostupné z: <https://docs.unity3d.com/Manual/fundamental-unity-types.html>. [cit. 2025-01-14].
3. UNITY TECHNOLOGIES. *GameObject*. Online. UNITY TECHNOLOGIES. Unity 6 User Manual. 2025. Dostupné z: <https://docs.unity3d.com/Manual/class-GameObject.html>. [cit. 2025-01-14].
4. UNITY TECHNOLOGIES. *MonoBehaviour*. Online. UNITY TECHNOLOGIES. Unity 6 User Manual. 2025. Dostupné z: <https://docs.unity3d.com/Manual/class-MonoBehaviour.html>. [cit. 2025-01-14].
5. UNITY TECHNOLOGIES. *Scriptable Object*. Online. UNITY TECHNOLOGIES. Unity 6 User Manual. 2025. Dostupné z: <https://docs.unity3d.com/Manual/class-ScriptableObject.html>. [cit. 2025-01-23].
6. *Transform*. Online. UNITY 6 USER MANUAL. Unity Technologies. 2025. Dostupné z: <https://docs.unity3d.com/Manual/scripting-transform.html>. [cit. 2025-01-23].
7. *RectTransform*. Online. UNITY 6 USER MANUAL. Unity Technologies. 2025. Dostupné z: <https://docs.unity3d.com/6000.0/Documentation/ScriptReference/RectTransform.html>. [cit. 2025-01-16].
8. UNITY TECHNOLOGIES. *TextMesh Pro Documentation*. Online. UNITY TECHNOLOGIES. Unity Documentation. 2025. Dostupné z: <https://docs.unity3d.com/Packages/com.unity.ugui@2.0/manual/TextMeshPro/>. [cit. 2025-01-13].
9. FRIEDMAN, Janice. *When Was C# Created? A Brief History*. Online. C# Station. 2020. Dostupné z: <https://csharp-station.com/when-was-c-sharp-created-a-brief-history/>. [cit. 2025-01-22].
10. MICROSOFT. *Prohlídka jazyka C#*. Online. MICROSOFT. Microsoft Learn. 2024. Dostupné z: <https://learn.microsoft.com/cs-cz/dotnet/csharp/tour-of-csharp/overview>. [cit. 2025-01-13].
11. BLENDER. *About Blender*. Online. BLENDER. Blender 4.3 Reference Manual. 2025, 10.1.2025. Dostupné z: <https://docs.blender.org/manual/en/latest/getting_started/about/>. [cit. 2025-01-13].
12. BROWN, Zack. *A Git Origin Story*. Online. LINUX journal. 2018. Dostupné z: <https://www.linuxjournal.com/content/git-origin-story>. [cit. 2025-01-14].
13. W3 SCHOOLS. *XML Introduction*. Online. W3 SCHOOLS. W3 Schools. 1999. Dostupné z: <https://www.w3schools.com/xml/xml_whatis.asp>. [cit. 2025-01-09].
14. MEARLS, Mike a CRAWFORD, Jeremy. *Prirucka Hrace*. 10. Vydání. Wizards of the Coast, 2018. ISBN 978-0-7869-6560-1.
15. PAIZO. *Gamemastering*. Online. D20PFSRD. 2009. Dostupné z: <https://www.d20pfsrd.com/gamemastering>. [cit. 2025-01-16].
16. *10 TTRPGs That Define The Genre*. Online. CBR. 1995, s. 1. Dostupné z: <https://www.cbr.com/iconic-genre-defining-ttrpgs/>. [cit. 2025-01-15].

Seznam použitých symbolů a zkratek

|  |  |  |
| --- | --- | --- |
| Symbol | Vysvětlení |  |
| DnD | Dungeons & Dragons |  |
| TTRPG | Table Top Role-Playing Games |  |
|  |  |  |

Seznam obrázků

[Obr. 2.1 Obsah 10](#_Toc147493921)

[Obr. 2.2 Příklad umístění legendy obrázku 12](#_Toc147493922)

Seznam tabulek

[Tab. 2.1 Legenda k tabulce 12](#_Toc147493615)

*Pathfinder Roleplaying Game Core Rulebook*. 2nd E. Paizo, 2009. ISBN 978-1-60125-150-3.

Seznam příloh

Prázdná šablona maturitní práce