



Střední průmyslová škola strojní
a elektrotechnická a Vyšší odborná škola,
Liberec 1, Masarykova 3

2D MULTIPLAYEROVÁ HRA V UNITY

Maturitní práce

Autor
Obor
Vedoucí práce
Školní rok

Jan Zajíček
Informační technologie
Ing. Marek Pospíchal
2022/2023

Anotace

Práce pojednává o kompletním vytvoření akční hry pro více hráčů vytvořenou v herním engine Unity za použití knihovny Mirror Networking. Jejím zaměřením je možnost hraní s více lidmi přes internet. Tato práce vychází z původní arkádové hry Asteroids

Summary

This thesis is about the complete creation of a multiplayer action game created in the Unity game engine using the Mirror networking library. Its focus is the possibility of playing with multiple people over the internet. This work is based off the arcade game Asteroids.

Čestné prohlášení

Prohlašuji, že jsem předkládanou maturitní práci vypracoval sám a uvedl jsem veškerou použitou literaturu a bibliografické citace.

V Liberci dne 12. 3. 2023

.....

Jan Zajíček

Obsah

Anotace	2
Summary	2
Čestné prohlášení	2
Obsah	3
Úvod	1
1 Unity	2
1.1 Proč Unity	2
1.2 Pojmy v Unity	2
1.2.1 Asset	2
1.2.2 Game Object	2
1.2.3 Komponent	2
1.2.4 Transform	2
1.2.5 Collider	3
1.2.6 Trigger	3
1.2.7 Sprite Renderer	3
1.2.8 Rigidbody 2D	3
1.2.9 Prefab	3
1.2.10 Skriptovaný objekt	3
2 Herní enginy	4
2.1 Unity	4
2.2 UE4	4
2.3 Godot	5
3 Mirror Networking	6
3.1 Proč Mirror před Unity UNet/Netcode	6
3.2 Komponenty Mirror Networking	6
3.2.1 Network Manager	6
3.2.2 Network Identity	6

3.2.3	Network Transform.....	6
3.2.4	Network Start Position.....	7
4	Vývoj.....	8
4.1	Tvorba assetů	8
4.1.1	Skripty	8
4.1.2	Grafika	8
4.1.3	Začátek	8
4.1.4	Vytvoření hráče.....	9
4.1.5	Pohyb hráče.....	9
4.1.6	Střelba.....	10
4.1.7	Asteroidy (Nepřátelé).....	11
4.1.8	Cíl hry.....	12
4.1.9	Vytvoření a připojení do hry.....	12
4.2	Co mohlo být	13
	Závěr.....	14
	Seznam zkratk a odborných výrazů.....	15
	Seznam obrázků	16
	Použité zdroje	17
A.	Seznam přiložených souborů	I

Úvod

Video hry byly součástí mého života už od útlého věku, od různých FPS her, přes sandboxy (jako je slavná hra Minecraft), až po různé strategie a všechno okolo, a právě jedna z těchto her mě podnítila k vytvoření mé vlastní hry. Jedním z dalších důvodů je také nedostatek her, které by bavili mě a moji skupinu přátel.

Převážně veškeré zkušenosti, co se týče programování jsem nasbíral při svém čtyřletém pobytu na střední škole. Zajímat jsem se začal už na základní škole kde při některých hodinách informační technologie jsme zkoušeli základy programování například v jazyce Scratch. K tomu jsem také pár hodin si vyzkoušel napsat HTML soubor v notepadu. Na střední škole jsme se učily od základů programování přes webové stránky, desktopové aplikace, databáze až po umělou inteligenci. Vždy mě bavilo hrát hry s přáteli přes internet a hodně mě vždy zajímalo hostovat servery pro spoustu z těchto her, například již zmíněný Minecraft. Proto bych si chtěl i vyzkoušet jednu takovou multiplayer hru naprogramovat.

1 Unity

1.1 Proč Unity

V dnešní době je na světě spousta různých herních enginů jako například Unreal Engine, Godot apod., proto může být obtížné vybrat ten správný pro daný projekt. Unity jsem si vybral z několika důvodů. Unity je jeden z méně komplexnějších enginů a jednoduše dokáže jeden člověk nebo malý tým udělat hru v dobré kvalitě. Dají se v něm vytvářet jak 3D tak 2D hry a podporuje cross-platform, což znamená že hry jdou vytvářet na většinu platforem jako je PC, console nebo mobile. Unity má i svůj obchod s assety, ve kterém si uživatelé mohou najít cokoli co potřebují pro svoji hru. Tohle může velmi usnadnit vývoj hry. V tomto projektu jsem využil z obchodu knihovnu s názvem Mirror Networking, s kterou zakomponuji online multiplayer do hry.

1.2 Pojmy v Unity

1.2.1 Asset

Všechny soubory vložené do projektu. Například sprity či skripty.

1.2.2 Game Object

Základní stavěcí jednotka v Unity. Samotný Game Object nic nedělá, ale pomocí komponent se dá určit jeho vzhled, chování a vlastnosti

1.2.3 Komponent

Vlastnost, která se připojuje na Game Object.

1.2.4 Transform

Jedna z mnoha komponent. Udává pozici, rotaci a velikost objektu.

1.2.5 Collider

Komponenta, která udělá z objektu překážku, do které mohou narážet ostatní objekty. Skripty na objektu mohou využít metody „OnCollisionEnter()“, která se zavolá pokaždé, když dojde ke kolizi.

1.2.6 Trigger

Trigger je Collider, který se nechová jako překážka, ale chová se jako zóna, která volá metody „OnTriggerEnter()“ a „OnTriggerExit()“, podle toho zdali objekt vejde či odejde

1.2.7 Sprite Renderer

Komponenta, co má na starost zobrazování spritu a jeho chování.

1.2.8 Rigidody 2D

Rigidbody přidává objektu fyziku. Tento objekt se poté může pohybovat předáním kinetické energie, gravitace nebo jinými metodami Rigidbody.

1.2.9 Prefab

Jedná se o uložený objekt, který má přednastavené komponenty a jejich hodnoty. Díky tomu ho můžeme nekonečně krát okopírovat do scény. Každá změna na prefabu se objeví na všech jeho variantách, ale změny na variantách se na prefabu neprojeví.

1.2.10 Skriptovaný objekt

Skriptovaný objekt je třída v unity, která pouze slouží k uchování dat. Tyto data uchovaná v této třídě poté může používat více objektů naráz. Tímto se snižuje spotřeba paměti.

2 Herní engine

Herní engine je softwarový framework primárně určený pro vývoj videoher a obvykle zahrnuje příslušné knihovny a podpůrné programy. Herní engine může také označovat vývojový software využívající tento framework, který obvykle nabízí sadu nástrojů a funkcí pro vývoj her. Vývojáři mohou herní engine používat ke konstrukci her pro herní konzole a jiné typy počítačů. Základní funkce, které herní engine obvykle poskytuje, mohou zahrnovat vykreslovací engine ("renderer") pro 2D nebo 3D grafiku, fyzikální engine nebo detekci kolizí (a odezvu na kolize), zvuk, skriptování, animace, umělou inteligenci, síť, streaming, správu paměti, vlákno, podporu lokalizace, graf scény a podporu videa pro kinematiku. Mezi herní enginey, patří například zde použité Unity, Unreal Engine 4, Godot, Clausewitz.

2.1 Unity

Unity je multiplatformní herní engine vyvinutý společností Unity Technologies. Byl použit pro vývoj her pro PC, konzole, mobily a web. První verze podporovala pouze OS X a byla představena na celosvětové konferenci Applu v roce 2005. Od té doby byl rozvinut o více než patnáct dalších platforem. Unity poskytuje možnosti vývoje pro 2D i 3D hry libovolného žánru a zaměření. Kromě grafického prostředí pro tvorbu, podporuje také tvorbu skriptů v jazyce C#. Firma Unity Technologies provozuje bezplatnou i placenou verzi programu. Hlavním konkurentem (především na trhu her pro PC a konzole) je Unreal Engine od společnosti Epic Games.

2.2 UE4

Unreal Engine je herní engine, který byl vytvořen firmou Epic Games. Jeho první verze z roku 1998 byla použita ve hře Unreal. Od té doby byl Unreal Engine několikrát vylepšen a doplněn, aby mohl být použit v několika desítkách novějších herních titulů. Původně byl tento engine určen pouze pro střílečky z pohledu první osoby, ale našel využití i v některých MMORPG, RPG a adventurách. Jádro Unreal Enginu, napsané v programovacím jazyku C++, podporuje mnoho různých platforem, jako jsou Microsoft Windows, Linux, Mac OS, Mac OS X na PC. Nejnovější verze Unreal Enginu podporuje také platformy herních konzolí Dreamcast, Xbox, Xbox 360, PlayStation 2 a PlayStation 3 i PlayStation 4. Většina ostatních součástí enginu ve hře je napsána ve speciálním kódu

UnrealScript, díky kterému není nutné při vytváření případných herních modů zasahovat hluboko do jádra, ale postačí pouze změnit skripty.

2.3 Godot

Godot je 2D a 3D multiplatformní open source herní engine pod licencí MIT vyvinutý komunitou, byl interně používán v několika společnostech v Latinské Americe, předtím než byl uvolněn jako open-source a zpřístupněn veřejnosti.[2] Vývojové prostředí běží na Windows, macOS a Linux (oba 32 a 64 bit) a může vytvářet hry cílené na PC, konzole, mobily a webu.

3 Mirror Networking

3.1 Proč Mirror před Unity UNet/Netcode

V době na začátku práce jsem chtěl použít nativní knihovnu v Unity UNet. UNet měl ale složitou implementaci a špatně sepsanou dokumentaci. Proto jsem hledal náhradu a přišel jsem na Mirror networking, která mi více vyhovovala. Unity později v roce 2022 vydalo Netcode for GameObjects, ta jako nová nativní knihovna v Unity má jednodušší implementaci.

3.2 Komponenty Mirror Networking

3.2.1 Network Manager

Network Manager je základní komponentou multiplayer hry. Na jednom místě shromažďuje mnoho užitečných funkcí a zjednodušuje vytváření, spouštění a ladění multiplayer her. Mezi funkce Správce sítě patří správa stavu hry, spawnů různých game objectů, správa scén, informace o ladění.

3.2.2 Network Identity

Síťová identita je základem vysokoúrovňového rozhraní API pro síť Unity. Řídí jedinečnou identitu herního objektu v síti a používá ji k tomu, aby si síťový systém uvědomil herní objekt.

3.2.3 Network Transform

Network Transform synchronizuje polohu, natočení a měřítko herních objektů v síti. Herní objekt se komponentou Network Transform musí mít také komponentu Network Identity. Když přidáte komponentu Network Transform k hernímu objektu, přidá Mirror k tomuto hernímu objektu také komponentu Network Identity, pokud ji ještě nemá. Ve výchozím nastavení je Network Transform autoritativní pro server, pokud nezměníte směr synchronizace na Client To Server. Autorita klienta se vztahuje na (hráčské objekty i nehráčské objekty, které byly výslovně přiřazeny klientovi, ale pouze pro tuto komponentu. Je-li tato volba povolena, jsou změny polohy odesílány z klienta na server.

3.2.4 Network Start Position

Network Manager ve výchozím nastavení spawnuje hráče v bodě (0,0,0). Pokud chceme hráče spawnovat na jiném místě, uděláme nový game object s komponentou Network Start Position. Object poté umístíme na souřadnici kde chceme hráče spawnout. Při více startovních pozicích můžeme hráče spawnovat náhodně, kruhovým cyklem, nebo vlastní implementací.

4 Vývoj

4.1 Tvorba assetů

Veškeré assety, které nejsou součástí knihovny Mirror Networking a všechny zvuky, jsem vytvořil sám.

4.1.1 Skripty

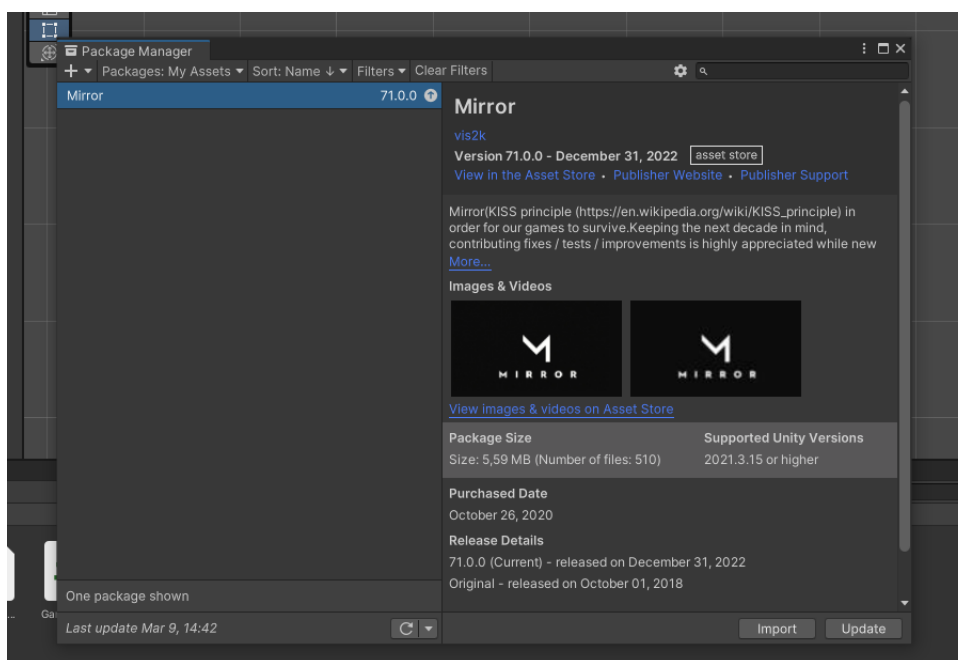
Všechny skripty jsou napsány v programovacím jazyce C#. K jejich psaní jsem využil vývojové prostředí Microsoft Visual Studio 2022.

4.1.2 Grafika

K vytvoření spritů jsem použil software Aseprite. Je to nástroj pro úpravu obrázků a animací. Primárně je určený pro tvorbu pixel art.

4.1.3 Začátek

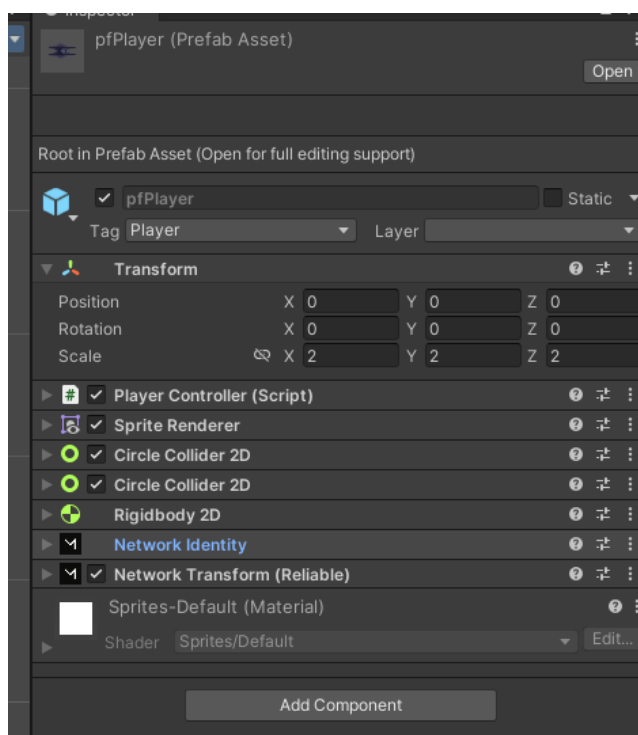
Jako první po založení projektu v Unity, jsem zašel do internetového obchodu s assety, kde jsem si vyhledal Mirror Networking a poté jsem jej pomocí Package Manageru naimportoval do projektu.



Obrázek 1 – Otevřený package manager s Mirror networking

4.1.4 Vytvoření hráče

Jako první je potřeba vytvořit nový Game Object, do kterého přidáme všechny komponenty, se kterými budeme pracovat. Komponenty, jež jsem přidal na tento Game Object jsou: Rigidbody 2D aby byl object pod vlivem fyzikálního engine, dva Circle Collidery, jeden pro kolize a druhý pro trigger, Network Identity, která umožní serveru vidět na hráče a Network Transform, který bude synchronizovat pozici a rotaci hráče. Vytvořil jsem také nový skript „PlayerController.cs“, který jsem přidal na tento Game Object.



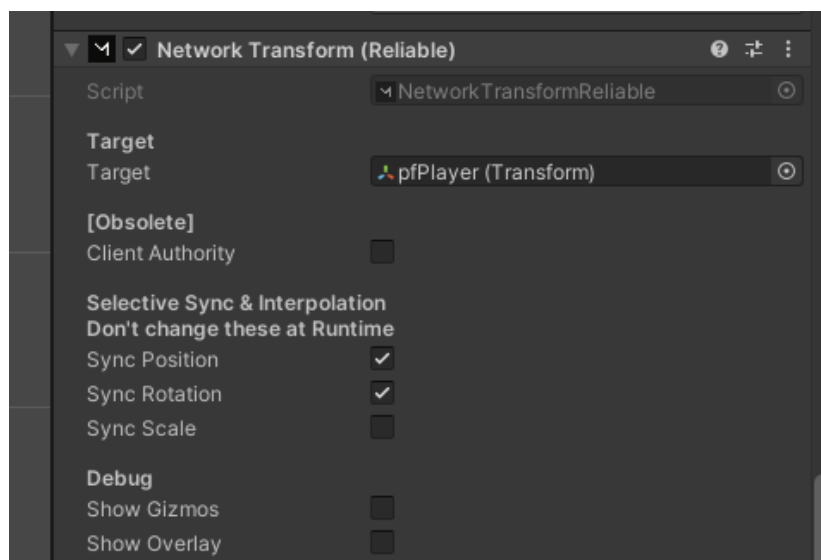
Obrázek 2 – Game Object hráče

4.1.5 Pohyb hráče

Pohyb hráče prošel několika iteracemi přes průběh vývoje. Přecházel jsem mezi klasickým pohybem WASD a otáčením za kurzorem myši, později jsem pohyb změnil na WASD pohyb kde klávesy A a D hráče neposouvají doleva či doprava, ale otáčejí jej. Později jsem ještě odstranil možnost posouvat se dozadu klávesou S. Při pohybu dopředu hráč po dobu držení klávesy W nabírá rychlost. Po odmačknutí klávesy hráč nezastaví hned ale postupně se zpomalí k zastavení (podobné skluzu na ledu).

```
//Pohyb vpřed
if (Input.GetKey(KeyCode.W))
{
    Vector3 move = Vector3.right * Time.deltaTime * speed;
    rigidbody.AddRelativeForce(move);
}
if (rigidbody.velocity.magnitude > maxSpeed)
{
    rigidbody.velocity=rigidbody.velocity.normalized*
maxSpeed;
}
```

Aby server a všichni klienti věděli kde se kdo nachází a jakou mají rotaci musíme mít zaškrtnuté Sync Position a Sync Rotation v Network Transform hráče, který synchronizaci pozic a rotací zajistí.



Obrázek 3 – Network transform komponent se zaškrtnutými Sync Position/Rotation

4.1.6 Střelba

Pro střelbu používám coroutine, která je v nekonečném zacyklení a kontroluje kdy hráč zmáčkne klávesu C pro střelbu. Po detekci zmáčknutí se zavolá jen z dvou metod „CmdFireProjectile()“ s Mirror dekorátorem [Command]. Tento dekorátor určuje metodu jako příkaz, který mohou používat pouze klienti. Když jej klient zavolá tak se tato metoda spustí pouze na serveru. V tomto případě tato metoda zavolá další metodu

„RpcFireProjectile()“, která teprve vytvoří nový projektil a vystřelí jej. Tato metoda má další dekorátor [ClientRpc]. Metody s tímto dekorátorem může použít pouze server, aby mohl říci nějakému klientovi, aby spustil tuto metodu.

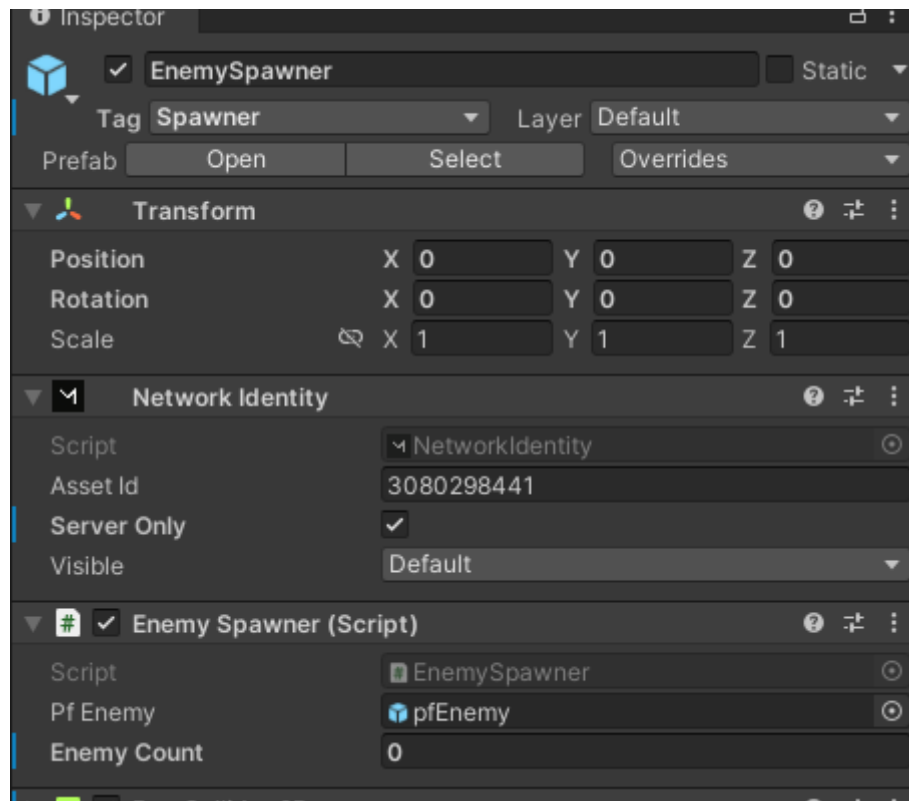
```
[Command]
public void CmdFireProjectile()
{
    RpcFireProjectile();
}
[ClientRpc]
public void RpcFireProjectile()
{
    if (NetworkServer.activeHost)
    {
        GameObject projectile = Instantiate(pfProjectile,
        projectileSpawner.transform.position, transform.rotation);
        Destroy(projectile, 7);
        NetworkServer.Spawn(projectile);
    }
}
```

V metodě „RpcFireProjectile()“ se vytvoří nový objekt, neboli projektil který hráč vystřelí. Aby ale server věděl o tomto projektilu musíme jej také přidat do jeho vědomí. To se dá udělat pomocí metody „Spawn()“ ze třídy NetworkServer. Samotný projektil so vytváříme je prefab se všemi komponentami předpřipravený.

4.1.7 Asteroidy (Nepřátelé)

Jako nepřátelé tu jsou asteroidy. O vytváření a počty těchto asteroid se stará script „EnemySpawner.cs“ ten pokaždé vybere náhodný bod a směr, kde a kam asteroid poletí. Také udržuje jejich počet a rychlost jejich vytváření. Hráč se těmto asteroidům snaží vyhýbat a zničit jich co nejvíce. Pokud hráč do nějakého narazí přijde o jeden ze svých 3 životů. Proměnná, která udržuje počet životů má dekorátor [SyncVar]. Proměnné, které tento dekorátor mají se server snaží synchronizovat mezi všechny klienty. Pokud jeden z hráčů o všechny životy přijde je do konce hry mimo a může pouze sledovat ostatní. Object se skriptem „EnemySpawner.cs“ má také Network Identity, kde má zaškrtnuté

Server Only. To zakáže vytvoření tohoto objectu na všech klientech a vytvoří se pouze na serveru. Bez toho by každý klient viděl dva asteroidy na jednu, jeden, co si klient vytvoří sám a druhý co vytvoří server, což je nechtěné chování.



Obrázek 4 – Object Enemy Spawner Server Only

Object EnemySpawner má jeden veliký collider nastavený jako trigger, který používám jako prostor kde se mohou vytvářet nové asteroidy a také jako hranice kam mohou docestovat. Pokud asteroid vyjde z prostoru EnemySpawner otočí svůj směr zpátky do herního prostoru. Nemusí se tak asteroid likvidovat a vytvářet nový. Asteroidy také mohou narážet mezi sebou což zničí oba asteroidy najednou.

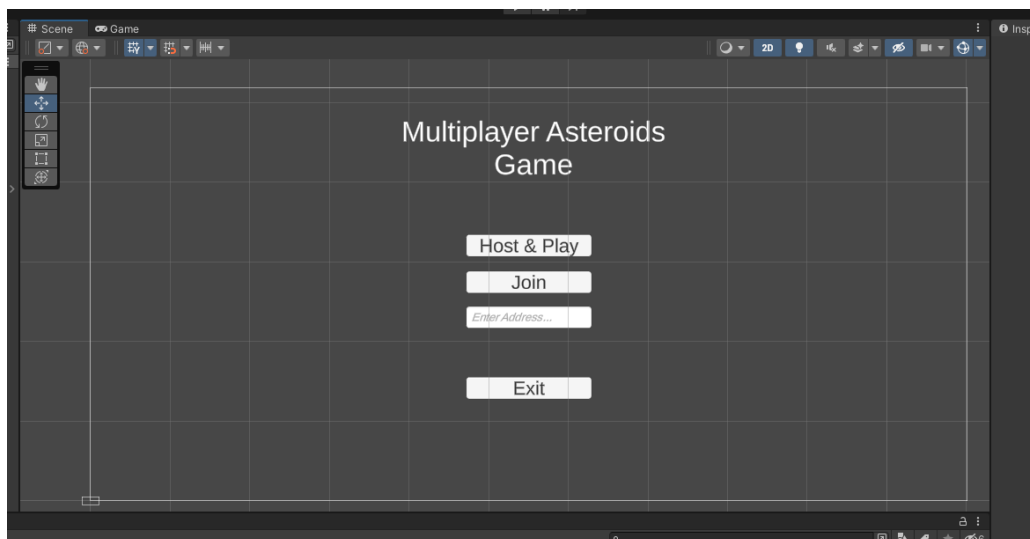
4.1.8 Cíl hry

Cílem jedné hry je přežít jako poslední hráč na hracím poli. Postupem času jsou asteroidy rychlejší a častěji se vytvářejí.

4.1.9 Vytvoření a připojení do hry

Po spuštění hry se objeví menu hry. Zde jsou dvě tlačítka a jeden textové pole. První tlačítko pošle hráče do herní scény a spustí server na portu 27015. Druhé tlačítko se snaží hráče připojit na IP adresu napsanou v textovém poli hned pod ním. IP adresa

musí být napsaná ve formě „1.2.3.4:27015“. Pokud hráč, který hostuje hru nemá veřejnou IP adresu bude si muset tento port muset přesměrovat. To je specifické k routeru hráče.



Obrázek 5 - Méně scéna z pohledu editoru

4.2 Co mohlo být

Tato hra je pouhý základ toho, co by mohla být. Přistavět by se dalo spousta věcí. Například více nepřátel, určitě něco, třeba nějaká vesmírná loď, co by hráče pronásledovala a snažila se jej sestřelit nebo do něj narážet. Také by se dali přidat různé bonusy pro hráče. Mezi nimi by mohly být různé sbíratelé věci co by mohli hráči dostat z nepřátel. Například více životů, štít, nebo třeba nová zbraň pro jeho vlastní loď. Také by zde mohli být různé styly hraní, kde by mohli hráči hrát kooperativně, či dokonce moci i střílet proti sobě. Určitě je tu spousta různých cest kudy by se vývoj hry mohl vydat.

Závěr

Cílem této práce bylo zjistit a naučit se, jak se dělá síťování multiplayerových her v prostředí herního enginu Unity. Ve světě tvoření online videoher je spousta věcí, co si člověk musí hlídat a je velice jednoduché, aby se cokoliv pokazilo. V tomto případě jsem použil způsob tvorby hry tak, že jsem nejprve udělal jednoduchou verzi, která neměla žádné síťování. Později když jsem začal tuto funkcionalitu doplňovat tak všechno co se mohlo pokazit se pokazilo. Vynořilo se spousta problémů, kvůli kterým jsem musel předělat spoustu věcí. Důležité je, pokud se vytváří online multiplayer hra, je začít se síťováním hned od úplného začátku. Tato práce je ale pouhé minimum, co se dá vytvořit. Téma online multiplayer her mě velice zajímá, jak z pohledu hraní, tak i z pohledu hlubšího porozumění, jak takové hry fungují a moc rád bych se chtěl do tohoto tématu hlouběji ponořit.

Seznam zkratk a odborných výrazů

FPS

First Person Shooter – Střílečí hra z pohledu první osoby

Sandbox

V překladu „pískoviště“ označuje hry, které často nemají daný cíl. Pouze hráči dají nástroje k tvoření, se kterými si hráč může stvořit co si zamane.

HTML

HyperText Markup Language – Značkovací jazyk pro tvorbu webových stránek

Notepad

Poznámkový blok – jedna z výchozích aplikací v systému Windows

Platform

Prostředí, ve kterém se spouští software, např. herní konzole nebo operační systém

Cross-platform

Software, který může fungovat na více typech platform

WASD

Klávesy na klávesnici, často používané pro pohyb

Seznam obrázků

Nenalezena položka seznamu obrázků.

Použité zdroje

1. mirror-networking. *Mirror Networking*. [Online] [Citace: 10. 11 2021.]
<https://mirror-networking.gitbook.io/docs/>.
2. unity.com. *Coroutines*. [Online] Unity Technologies.
<https://docs.unity3d.com/Manual/Coroutines.html>.
3. answers.unity.com. *answers.unity.com*. [Online] Unity Technologies.
<https://answers.unity.com/questions>.
4. Unity. unity.com. *Documentation*. [Online] Unity Technologies. [Citace: 10. 11 2021.] <https://docs.unity3d.com/Manual/index.html>.
5. forum.unity.com. *Forums*. [Online] Unity Technologies. forum.unity.com.

A. Seznam příložených souborů

Na přiloženém datovém nosiči se nacházejí následující soubory a složky:

- **MP2022-Zajíček-Jan-P4-2D_Multiplayerová_hra_v_Unity.docx** – editovatelná verze dokumentace maturitní práce
- **MP2022-Zajíček-Jan-P4-2D_Multiplayerová_hra_v_Unity.pdf** – tisknutelná verze dokumentace maturitní práce
- **Aplikace** – zdrojové kódy