GYMNÁZIUM, STŘEDNÍ PEDAGOGICKÁ ŠKOLA, OBCHODNÍ AKADEMIE A JAZYKOVÁ ŠKOLA s právem státní jazykové zkoušky ZNOJMO, příspěvková organizace

669 02 Znojmo, Pontassievská 3 • tel.: 515158101 • fax: 515225230 • e-mail: info@gpoa.cz • www.gpoa.cz, IČ:49438816

Maturitní práce

Žák: David Beneš
Třída: E4.C
Školní rok: 2017/2018
Obor vzdělání: Informační technologie

Název maturitní práce: Tvorba počítačové hry

Vedoucí maturitní práce: Mgr. Milička Tomáš
Oponent maturitní práce:

Sociální partner: GYMNÁZIUM, STŘEDNÍ PEDAGOGICKÁ ŠKOLA, OBCHODNÍ AKADEMIE A JAZYKOVÁ ŠKOLA s právem státní jazykové zkoušky ZNOJMO, příspěvková organizace

Práce odevzdána dne: (datum)

Podpis žáka: (podpis)

Prohlášení

Prohlašuji, že jsem maturitní práci vypracoval samostatně pod vedením Mgr. Tomáše Miličky a uvedl v seznamu literatury a zdrojů veškerou použitou literaturu a další informační zdroje.

Místo datum

David Beneš

Anotace

Maturitní práce se zaměřuje na programování v jazyce C# v herním enginu Unity. V práci je popsán vývoj hry a veškeré problémy, které s vývojem souvisí.

Klíčová slova

hra, csharp, unity, programování

Obsah

1 Úvod **4**
1.1 Vymezení cílů . . . 4

2 Využité programy **5**
2.1 Unity . . . 5
2.1.1 Představení . . . 5
2.1.2 Důvod pouzití . . . 5
2.1.3 Výhody . . . 5
2.1.4 Nevýhody . . . 5
2.1.5 Alternativy . . . 5
2.2 Visual Studio . . . 5
2.2.1 Představení . . . 5
2.2.2 Důvod pouzití . . . 6
2.2.3 Výhody . . . 6
2.2.4 Nevýhody . . . 6
2.2.5 Alternativy . . . 6

3 Průběh vývoje **7**
3.1 Výběr žánru . . . 7
3.2 Ovládání postavy . . . 7
3.3 Schopnosti . . . 7
3.3.1 Blink . . . 7
3.3.2 Neviditelnost . . . 7
3.3.3 Cooldown . . . 8
3.4 Nepřítel . . . 8
3.5 Cíl úrovní . . . 8

4 Problémy vývoje **9**

1 Úvod

Téma jsem si vybral, protože mě vždy zajímalo jak se hry vyvíjí. Dále jsem chtěl vědět, jestli vůbec takovou hru dokážu naprogramovat. Také jsem zastáncem názoru, že programování se dá naučit jedině praxí.

1.1 Vymezení cílů

Mým hlavním cílem je naprogramovat hru pomocí jazyka C#. Neměla by být moc obtížná na naučení, ale měla by být těžká na pokoření (Bushnell). Game design by měl být jednoduchý a přehledný, ať se hráč jednoduše vyzná kde se právě nachází. Hra by se měla zaměřovat na schovávání se před nepřáteli, a ne na jejich zabíjení.

2 Využité programy

2.1 Unity

2.1.1 Představení

Unity je herní engine. Je používaný pro vývoj jak indie (nezávislých), tak AAA her. Je v něm možná naprogramovat hru v jazyce C#, který je rozšířen o mnoho Unity knihoven, ale i v JavaScriptu.

2.1.2 Důvod pouzití

Unity jsem si vybral, protože se mi hned na první pohled zalíbilo. Má krásný a jednoduchý design. Také znám spoustu her, které jsou za pomocí Unity enginu vytvořené. Velikým důvodem pro výběr byla také dostupnost v jazyce C#, což ne všechny herní enginy nabízejí.

2.1.3 Výhody

- Jednoduchost
- Je zdarma pro osobní i komerční účely (do $100k)
- Dostupnost jazyka C#

2.1.4 Nevýhody

- Pro úplné nástroje je třeba zaplatit

2.1.5 Alternativy

- Unreal Engine
  - Od společnosti Epic Games
  - Více se zaměřuje na systém akce a reakce, než se zaměřuje na programování
- CryEngine
  - Od společnosti Crytech

2.2 Visual Studio

2.2.1 Představení

Pro upravování C# skriptů jsem použil program Visual Studio Community 2017 Corporation. Tento program vytvořil microsoft, proto má skvělou podporu jazyka C#.

2.2.2 Důvod pouzití

2.2.3 Výhody

- Je zdarma pro osobní i komerční účely pro jedince
- Dostupnost jazyka C#
- Jednoduché debugování

2.2.4 Nevýhody

- Není multiplatformní (neexistuje verze pro linux)

2.2.5 Alternativy

- monodevelop
- atom
- vs code

3 Průběh vývoje

3.1 Výběr žánru

Na začátku jsem musel vymyslet v jakém duchu budu hru vyvýjet. Jelikož mám velice rád stealth hry **?**, rozhodl jsem se vytvořit v tomto duchu. Nechtěl jsem, aby hráč začil nějakou výzvu, a tak ve hře nejsou žádné zbraně. Pro záživnější průbeh jsem se rozhodl zahrnout do hry nějaké superschopnosti.

3.2 Ovládání postavy

Hráč ovládá postavu pomocí základních WASD tlačítek. Pomocí myši míří a rozhlíží se po úrovni. Pro pohyb jsem využil Vector3 **?** a to tak, že jsem k pozici postavy přičetl jednotkový vektor a vynásobil rychlostí, kterou jsem si vybral. Pro otáčení hráče ke kurzoru jsem prvně musel zjistit, kde je kurzor. To jsem udělal pomocí kamery, která je na scéně. Ta dokáže vypočítat, kam na jaký bod kurzor míří. Aby se postava nedívala do země, přičetl jsem výšku postavy k tomuto bodu. Pro samotné otočení postavy k bodu jsem musel využít quaternion **?**. Ten sice zjistí, jak daleko se musí postava otočit, ale poté je nutno ho převést na tzv. eulerAngles **?**. Díky tomu pak víme, o kolik se na jednotlivých osách postava musí natočit.

3.3 Schopnosti

Při výběru schopností jsem se nechal inspirovat ze stealth her a moba her. Hlavně pak Dishonored a Dota2.

3.3.1 Blink

Blink je teleportace na krátkou vzdálenost. Pomáhá hráči k rychlému přesunu. Prvně jsem se rozhodl vytvořit vizuální zobrazení pro hráče. Vytvořil jsem čáru, které začala v postavě a končila na kurzoru hráče (pro získání kurzoru jsem využil stejný postup jako při otáčení postavy). Na konci čáry jsem přidal kostku, aby hráč věděl, kam se přesune. Poté jsem řešil maximální vzdálenost teleportu. Vytvořil jsem si proměnou, která uchovávála desetinné číslo, které určuje maximální vzálenost. Pomocí IF podmínky jsem zjistil, jestli vzdálenost postavy od konce čáry je delší než je tato proměná. Pokud ano zkrátím čáru na maximální vzdálenost ve směru kurzoru. Poté jsem musel zabránit teleportaci skrz zdi. Prvně jsem zjistil, jestli je na konci čáry zeď pomocí Physics.Raycast **?**. Pokud ano, tak čáru zkrátil na vzdálenost hráče od této zdi. Nakonec jsem musel vyřešit samotný teleport. To bylo velice jednoduché, protože stačilo nastavit pozici postavy na pozici, která je na konci čáry.

3.3.2 Neviditelnost

Pro neviditelnost jsem použil globální proměnnou, kterou jsem si vytvořil. Tu jsem pak předával nepříteli, aby "nevnímal" hráče. Težší bylo zajistit, aby neviditelnost trvala nějakou dobu a pak se hráč opět zviditelni. Vytvořil jsem while cyklus a proměnou, která uchovávala čas, který může být hráč

neviditelný. Každým průchodem cyklu jsem od této proměné odečetl sekundu pomocí Time.Deltatime **?**. Dokud byla proměná větší jak 0, tak byl hráč neviditelný. Aby hráč věděl, kdy je neviditelný musel jsem změnit barvu postavy. Prvně jsem získal materiál postavy a pak nastavil průhlednost nastavil na 1/3. Až se hráč opět zviditelnil, nastavil jsem zpět hodnotu 1.

3.3.3 Cooldown

Aby hráč nemohl být neviditelný donekonečna, a nemohl se stále teleportovat musel jsem vytvořit cooldown. Pro cooldown jsem využil podobný cyklus jako pro neviditelnost. Jediné co jsem musel přidat byla proměná, která uchovává, jestli hráč muže schopnost použít. Před cyklem jsem ji nastavil na false a po cyklu opět na true. Aby šlo vidět, kolik času zbývá, rozhodl jsem se ho zakomponovat do UI.

3.4 Nepřítel

3.5 Cíl úrovní

4 Problémy vývoje

Reference

Nolan Bushnell. Theorem: Easy to learn, difficult to master. http://www.wolfsheadonline.com/bushnells-theorem-easy-to-learn-difficult-to-master.

Microsoft Corporation. Visual studio community 2017. https://www.visualstudio.com/vs/community.