Střední průmyslová škola, Ústí nad Labem, Resslova 5



Anarchy Squad

Dokumentace k ročníkové práci

**Autor:** Matěj Bezděka

**Třída:** 4ITB

**Vedoucí práce:** Bc. Vratislav Medřický 2023/2024

Prohlášení

Prohlašuji, že jsem ročníkovou práci na téma „Anarchy Squad“ vypracoval samostatně a s použitím uvedené literatury a pramenů.

V Ústí nad Labem dne 18. 4. 2024

Poděkování

Chtěl bych poděkovat Bc. Vratislavu Medřickému za vedení mé ročníkové práce, cenné rady a odborný dohled.

Anotace

Dokument pojednává o ročníkové práci Anarchy Squad, slouží také jako jeho popis. Obsahuje také rešerše o konkurenci i jakou roli sehráli v inspiraci. Nadále je připravená i praktická část s návrhy s produktizací a návodem pro uživatele, na konci i závěr o projektu a jeho přípravě.

Klíčová slova

Unity, Blender, 3D, hráč, nepřátelé, strategie, generování, hra, vylepšení

Obsah

[Úvod 7](#_Toc163932299)

[1 Teoretická část 8](#_Toc163932300)

[1.1 Rešerše 8](#_Toc163932301)

[1.1.1 Vybírání tevhnoligií 8](#_Toc163932302)

[1.2 Technologie 9](#_Toc163932303)

[1.2.1 Unity 9](#_Toc163932304)

[1.2.2 JetBrains Rider 9](#_Toc163932305)

[1.2.3 C# 9](#_Toc163932306)

[1.2.4 Blender 9](#_Toc163932307)

[1.2.5 Gimp 9](#_Toc163932308)

[2 Praktická část 10](#_Toc163932309)

[2.1 Návrhy 10](#_Toc163932310)

[2.1.1 Shader graf budov 10](#_Toc163932311)

[2.2 Produktizace 10](#_Toc163932312)

[2.2.1 Kontrola průchodnosti mapy 10](#_Toc163932313)

[2.3 Popis pro uživatele 14](#_Toc163932314)

[2.3.1 Výběr jednotek 14](#_Toc163932315)

[2.3.2 Hra 14](#_Toc163932316)

[Závěr 17](#_Toc163932317)

[Použitá literatura 18](#_Toc163932318)

[Seznam obrázků 19](#_Toc163932319)

[Obsah média 20](#_Toc163932320)

Úvod

Za ročníkovou práci chci vytvořit videohru se jménem Anarchy Squad. Hru jsem si vybral, protože je to trávím s hrami spoustu času, navíc díky komplexnosti vývoji her je příležitost vyzkoušet spoustu různých druhů prací spojený nejen s programováním.

Po úspěšném naistalováni hry a následném spuštění se uživatel ocitne v hlavním menu hry, tam bude mít na výběr nastavení grafických možností, ovládání i zvuku. Dále bude mít hráč možnost hrát dříve rozehranou hru nebo začít novou.

Po zvolení nové hry se uživatel dostane na obrazovku s mapou. Na mapě budou zobrazeny náhodné mise s různými prvky jako je velikost herní desky, množství nepřátel i jejich obtížnost, nebo zastavěnost plochy. Cíle na mapě budou také různé jako zabrání bodu nebo eliminace všech cílů.

Po vybrání mapy se uživatel bude rozhodovat, jak si postavit tým. Na výběr bude počet jednotek. Statistiky jednotek jako životy, rychlost, přesnost, obrana. Dále je nutné vybrané jednotky vybavit zbraní. V základu budou tři typy zbraní, první střílí jednu kulku za čas, druhá střílí v dávkách a poslední několik kulek najednou. Jako poslední bude si vybrat předměty které jednotka bude mít a počet. Vše bude mít cenu bodů které omezí, aby uživatel neměl vše na začátku hry.

Až uživatel zvolí svou skupinu bude přenesen na samotné bojiště. Kde uživatel bude mít za úkol splnit předem daný cíl, nebo v případě kdy budou jednotky smrtelně ohroženy bude potřeba utéct. Za správně splněnou misi bude uživatel odměněn body. V případě nesplnění úkolu nebo eliminace všech jednotek uživatel dostane méně. Při hraní bude uživatel mít i zpomalení nebo zastavení času.

Nepřátelé budou chytřejší než jen, vidím a střílím. Budou dodržovat hromadnou taktiku a postupovat dohromady. Tedy pokud se jim daří, pokud budou mít málo životů nebo budou ve velkém nebezpečí budou dělat unáhlená rozhodnutí, budou moct i utéct z boje.

Dokončený projekt bude zveřejněn ke stažení.

# Teoretická část

## Rešerše

### Vyběr technologií

#### Herní Engine

#### Editor kódu

#### Modelovací software

### Inspirace a konkurence

#### RimWorld

Rimworld je hra ve které má hráč na starost svou vlastní kolonii na planetě plné pirátů, gangů a jiných nepřátel. Musí se vyzbrojit připravit kolonii na obranu ale i udržovat morálku, mít plné sklady jídla i provádět výzkum nových technologií.

Líbilo se mi provedení, jak funguje boj a je využíváno celé prostředí, na první pohled se jeví jako velmi jednoduchý, ale je počítáno se spousty faktory, na které hráč jen tak nepomyslí. Také samotný managment celé kolonie je skvělý.



Obrázek 1 - snímek ze hry Rimworld - (7)

#### Age of Empires II

Strategická hra, kde je úkol zničit nepřítele. Bude potřeba vybudovat dostatečně silnou ekonomii pro dostatečně silnou armádu, která bude moct přemoci nepřítele. Hráč si navíc může vybrat z mnoha civilizací, které mají různé bonusy a jsou lepší v jiných ohledech.

Age of Empires nabízí široké možnosti hratelnosti díky velkému množství budov, vylepšení a jednotek k dispozici.



Obrázek 2- snímek ze hry Age of Empires II - (8)

#### Bad North

Nezávislá hra od jednoho člověka. Hráč brání ostrov proti vlnám nájezdníků. K dispozici má hráč generály se skupinou vojáků, ti pak můžou být vylepšováni a měnit styl boje. Hratelnost je jednoduchá ale obtížnost se velmi rychle stupňuje a z malých generovaných ostrůvku se pak stane krvavé bojiště.

[](https://store.steampowered.com/app/688420/Bad_North_Jotunn_Edition/)Hra má jen pár prvků za to je na každý kladen velký důraz a každý je velmi důležitý, aby se hráč dostal na poslední ostrov je nucen využít vše co má.

Obrázek 3 - snímek ze hry Bad North - (9)

Zaujalo mě i místo odehrávání bojů, které jsou na malém vygenerovaném ostrovu, doplněn zvoleným stylem grafiky.

## Technologie

### Unity is a new 3D game engine with a new identityUnity

Unity je multiplatformní herní engine vyvinutý společností Unity Technologies. 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#. (1)

Unity byl pro autora nejlepší možnost, protože má s tímto programem zkušenosti a dříve už používal. Značná výhoda je také používá C# vhodné pro autora.

Obrázek 4 – Logo Unity

### Obsah obrázku Grafika, logo, grafický design, symbol Popis byl vytvořen automatickyJetBrains Rider

JetBrains Rider je multiplatformní editor C# scriptů, které vychází z .NET IDE a je založené na platformě IntelliJ a ReSharper. (2)

Tento program byl autorem zvolen, protože už ho autor dříve využíval, Jeho výhodou nad programem Visual studio je, že se zaměřuje pouze na C#.

Obrázek 5 – Logo Jetbrains Rider

### 20 Years Anniversary of C# Programming Language. What the Future Holds. - AvengaC#

C# je objektově orientovaný programovací jazyk od společnosti Microsoft, který umožňuje vývojářům vytvářet aplikace běžící na platformě .NET. (3)

Zvolen byl C#, protože byl již dříve autorem používán. Unity, kde jsou veškeré scripty také využívá C#.

Obrázek 6 – Logo C#

### Obsah obrázku kruh, Grafika, Barevnost, kreslené Popis byl vytvořen automatickyBlender

Blender je populární 3D modelovací program, ve kterém lze vytvářet profesionální modely (např. do her), animované filmy nebo například fyzikální simulace. (4)

S tímto programem se Autor už několikrát setkal, a proto byl zvolen na tvorbu modelů i jejich následné animování.

Obrázek 7 – Logo Blender

### Obsah obrázku kreslené, nářadí, umění Popis byl vytvořen automatickyGimp

GIMP je multiplatformní grafický editor sloužící ke tvorbě a úpravě obrázků. Obsahuje širokou škálu standardních nástrojů. (5)

Jednoduchý program, který je užitečný na úpravu textur a spritů.

Obrázek 8 – Logo Gimp

### Obsah obrázku Grafika, Barevnost, umění, kreativita Popis byl vytvořen automatickyAudacity

Audacity je všestranný open-source audio editor, který umožňuje nahrávání, úpravy a mixování zvuků. Je ideální jak pro úplné začátečníky, tak i hudebníky a obecně každého, kdo potřebuje upravit audio soubory s profesionálními nástroji, ale zdarma. (6)

Obrázek 9 - Logo Audocity

# Praktická část

## Návrhy

### Shader graf budov

Shader graf z Unity. Shader je využíván na budovy kde se jim podle výšky postupně mění barva z jedné na druhou. Podobný princip je využíván i na 2D sprity využívané v grafickém rozhraní hry.

Obsah obrázku snímek obrazovky, text, 3d modelování, Multimediální software

Popis byl vytvořen automaticky

Obrázek 10 - Graf shaderu budov – vlastní obrázek

## Produktizace

### Kontrola průchodnosti mapy

Když se generuje mapa je důležité, aby se jednotky někde nezasekli nebo se nemohli někam dostat co by mohlo být důležité pro průběh hry. Protože je mapa menší můžeme si dovolit použít velmi jednoduchý způsob kontroly. Nazývá se Flood-fill, tento způsob zkontroluje průchodnost po každém generování budovy. Od prvního základního čtverce se kontroluje každá ze 4 stran čtverce, kontroluje, jestli je zastavěný, pokud ne, zapíše si souřadnice a přidá do počtu nalezených přístupných míst a následně pokračuje na další zapsaný čtverec. Pokud se na konci zjistí že očekáváný počet volných míst je větší tak není mapa průchodná a překážka se nevygeneruje.

Nevýhody tohoto algoritmu je velmi špatná škálovatelnost a na větší pole, než je 40 na 40 se začíná čas generace mapy značně prodlužovat. Na paměť není algoritmus šetřivý. Je to tedy velmi hloupý algoritmus, ale na tyto účely je dostačující.

### Jak nepřátelé vidí mapu

Kód 1 - Flood-fill algoritmus

1. **bool** MapIsFullyAccessible(**bool**[,] obstacleMap, **int** currentObstacleCount) {
2. **bool**[,] mapFlags = new **bool**[obstacleMap.GetLength(0), obstacleMap.GetLength(1)];
3. Queue<Coord> queue = new Queue<Coord>();
4. queue.Enqueue(playerSpawn);
5. **mapFlags[playerSpawn.x, playerSpawn.y] = true;**
6. **int** accessibleTileCount = queue.Count;
8. **while** (queue.Count > 0) {
9. Coord tile = queue.Dequeue();
10. **for (int x = -1; x <= 1; x++) {**
11. **for** (**int** y = -1; y <= 1; y++) {
12. **int** neighborX = tile.x + x;
13. **int** neighborY = tile.y + y;
14. **if** (x == 0 ^ y == 0) {
15. **if (neighborX >= 0 && neighborX < obstacleMap.GetLength(0) && neighborY >= 0 &&**
16. neighborY < obstacleMap.GetLength(1)) {
17. **if** (!mapFlags[neighborX, neighborY] && !obstacleMap[neighborX, neighborY]) {
18. mapFlags[neighborX, neighborY] = **true**;
19. queue.Enqueue(new Coord(neighborX, neighborY));
20. **accessibleTileCount++;**
21. }
22. }
23. }
24. }
25. **}**
26. }
28. **int** targetAccessibleTileCount = (mapSizeX \* mapSizeY - currentObstacleCount);
29. **return** targetAccessibleTileCount == accessibleTileCount;
30. **}**

Další problém při generování mapy je přidávání nových nepřátel, nebo kam má nepřítel začít běžet, pokud chce jít do bezpečí a vyhnout se uživatelským jednotkám.

Spočítá se vzdálenost každého políčka od nejbližší jednotky k danému poli (žluté pole). Podle této metriky mohou nepřátelé vyhodnotit kam jít v případě nebezpečí, nebo pokud se chtějí vzdálit přebíjet zbraň, ale jestli chce zaútočit tak si rovnou vyberou nebližší jednotku podle vzdálenosti trasy a vydají se k ní. Také lze umístit na nejvzdálenější pole (vyznačena fialově) i pole s nižší hodnotou umístit nové jednotky, které se neobjeví v zádech uživatelovi jednotky, což by mohlo být nefér a frustrující. Šedá pole vyznačují budovy s hodnotou -1.

Výpočet využívá princip flood-fill algoritmu, bylo by zbytečné neustálé přepočítávání, a navíc by zbytečně výpočet zatěžoval proces, proto je volán každou sekundu namísto každého snímku hry.

### Obsah obrázku snímek obrazovky, vzor, čtverec, Barevnost Popis byl vytvořen automaticky

Obrázek 11 - vizualizace hodnot polí – vlastní obrázek

### Automat jednotek

Nedílná součást jednotek a jejich chování je tento automat. Každá jednotka začíná v „Normal state“ a podmínka, nebo uživatelskými příkazy mění v jakém stavu se právě nachází. Tyto stavy způsobují že kód je izolován od ostatních stavů a je ve třídě pouze relevantní pro daný stav. Protože jednotka může mít pouze jeden stav, tudíž funkce update způsobuje pouze aktivuje update v daném stavu a neprovádí se další stavy.

Stavy mají navíc 3 fáze. Stav vždy začne ve fázi „enter“, pokračuje po iteraci do fáze „update“. Dokud se nesplní podmínka pro změnu stavu nebo vnější vliv přepne bude chtít přepnout stav silou. Poslední fáze je „exit“ kdy končí svou funkci a přepíná fázi na „enter“ v jiném stavu. Pokaždé se musí vykonat aspoň stavy enter a exit.

Pomocí tohoto systému a použití abstraktní třídy Stavu je velmi jednoduché přidávat další stavy bez velkého zásahu do kódu, nebo na náročnost celého systému.

1. **public** **abstract** **class** UnitState : MonoBehaviour
2. {
3. **protected** **enum** stateStages {
4. entry, update, exit
5. **}**
6. **protected** stateStages currentStage = stateStages.entry;
7. UnitState nextState;
8. **protected** SquadUnit unit;
9. **public** UnitState(SquadUnit unit) {
10. **this.unit = unit;**
11. }
12. **protected** **virtual** **void** Enter() {
13. currentStage = stateStages.update;
14. }
16. **protected** **abstract** **void** UpdateState();
18. **protected** **virtual** **void** Exit(UnitState state) {
19. nextState = state;
20. **currentStage = stateStages.exit;**
21. }
22. **public** UnitState Process() {
23. **switch** (currentStage) {
24. **case** stateStages.entry:
25. **Enter();**
26. **break**;
27. **case** stateStages.update:
28. UpdateState();
29. **break**;
30. **case stateStages.exit:**
31. **default**:
32. currentStage = stateStages.exit;
33. **return** nextState;
34. }
35. **return this;**
36. }
37. **public** **void** ForceChangeState(UnitState newState) {
38. Exit(newState);
39. }
40. **}**

Kód 2 - Automat jednotek

Kód 3 - Stav jednotky při přebíjení

1. **public** **class** ReloadUnitState : UnitState {
2. **float** reloadTime;
3. **float** currentCooldown = 0;
4. **bool** reloaded = **false**;
5. **UnitState previousState;**
6. **public** ReloadUnitState(SquadUnit unit, **float** reloadTime, UnitState previousState) : **base**(unit) {
7. **this**.reloadTime = reloadTime;
8. **this**.previousState = previousState;
9. }
10. **public ReloadUnitState(SquadUnit unit, float reloadTime) : base(unit) {**
11. **this**.reloadTime = reloadTime;
12. previousState = new NormalUnitState(unit);
13. }
14. **protected** **override** **void** Enter() {
15. **unit.InvokeStartReloading(reloadTime);**
16. **base**.Enter();
17. }
18. **protected** **override** **void** UpdateState() {
19. currentCooldown += Time.deltaTime;
20. **unit.InvokeReloading(reloadTime-currentCooldown);**
21. **if** (currentCooldown > reloadTime) {
22. unit.weapon.Reloaded();
23. reloaded = **true**;
24. Exit(previousState);
25. **}**
26. }
27. **protected** **override** **void** Exit(UnitState state) {
28. currentCooldown = 0;
29. unit.InvokeReloading(-1);
30. ***//prevents repeating reload while reloading***
31. **if** (state is AttackUnitState && !reloaded || state is ReloadUnitState) {
32. currentStage = stateStages.update;
33. **return**;
34. }
35. **base.Exit(state);**
36. }
37. }

## Popis pro uživatele

### Hlavní menu

Při zapnutí hry uvítá uživatele hlavní menu. Zde si uživatel může vybrat z nabídnutých možností. Jako je ukončit program, nebo otevřít okno s nastavením hlasitosti hudby, efektů, zvuků a celkové hlasitosti i grafických možností.

Obsah obrázku text, snímek obrazovky, Elektricky modrá

Popis byl vytvořen automaticky

Obrázek 12 - Hlavní menu – vlastní obrázek

Obsah obrázku text, snímek obrazovky, Multimediální software

Popis byl vytvořen automaticky

Obrázek 13 - možnosti nastavení – vlastní obrázek

### Výběr mise

Obsah obrázku text, mapa, snímek obrazovky

Popis byl vytvořen automaticky

Obrázek 14 - menu výběru misí – vlastní obrázek

### Výběr jednotek

Obrazovka kde si uživatel vybírá vlastní tým, který pošle do akce. Je omezen body. Každá položka stojí několik bodů. Cena položky naznačuje sílu položka a jak užitečná může být. K lepšímu vybavení se uživatel dostane postupným hraním v pozdějších fází hry.

Obsah obrázku text, snímek obrazovky, software, Multimediální software

Popis byl vytvořen automaticky

Obrázek 15 - Menu tvorby čety – vlastní obrázek

### Hra

Samotná hra, kde se objeví jednotky hráčem zvolené na hracím poli. Jednotky tedy pomocí hráče plní úkoly právě zadané jako „běž“, „střílej na nepřítele“, „hoď granát“ a jiné pomocí pravého tlačítka myši. Hráč může zvolit více jednotek najedou pomocí klinutím na ikonu v levém horním rohu, nebo kliknutí přímo na jednotku a zadávat úkoly hromadně. Při označení pouze jedné jednotky se ukáže rozšířená lišta možností, které má jednotka k dispozici.

Obsah obrázku snímek obrazovky, text, 3d modelování, Multimediální software

Popis byl vytvořen automaticky

Obrázek 16 - snímek ze hry – vlastní obrázek

Zde si hráč zvolil možnost házet granát, spustil se tedy režim házení granátu a ukazatel přibližně ukazuje kam granát dopadne a následně bouchne.

Obsah obrázku snímek obrazovky, 3d modelování, Počítačová hra, Multimediální software

Popis byl vytvořen automaticky

Obrázek 17- ukázka trajektorie granátu – vlastní obrázek

Závěr

Ročníková práce s názvem Anarchy squad dopadla obstojně. Povedlo se vytvořit hru kde si uživatel může zahrát velitele malé čety jednotek, která se dostane do střetu s ozbrojenými nepřáteli v městské oblasti na život a smrt. Uživatel si tedy musí dát pozor na nebezpečí a

Uživatelova četa, která se může skládat až ze 6 různých jednotek. Jednotky se liší svým zbraní a specializací jakou uživatel vybere dané jednotce. Existuje několik druhů zbraní, které se liši svými atributy stejně jako specializace. Jednotky mají svou umělou inteligenci a uživatel pouze zadává rozkazy co májí dané jednotky provézt za akce jako Jdi tam, přebij, střílej na cíl apod.

Ročníková práce s názvem Anarchy squad dopadla obstojně.

Nepřátelé

Mapa

Rozhraní

Grafika

Práce na projektu mě naučil

Závěr je **souvislý text**, ve kterém rozeberete, co se vám z uvedených cílů povedlo/nepovedlo. Nemělo by se jednat o odrážky/body! Většinou se závěr píše tak, že si otevřete Úvod a sepisujete, jak jste naplnili původní očekávání.

Zároveň v závěru můžete připsat, co vám projekt dal, kdo vám pomáhal, jestli pro něj máte nějaké plány do budoucna, jestli je/bude někde nasazen atd.

Délka závěru by měla být přibližně stejná jako délka Úvodu (**cca 1 stránka**). Neměl by obsahovat obrázky.

Použitá literatura

# Reference

1. ***Wikipedia.* [Online] https://cs.wikipedia.org/wiki/Unity\_(hern%C3%AD\_engine).**

**2. *JetBrains.* [Online] https://www.jetbrains.com/rider/.**

**3. *C#.* [Online] https://www.techtarget.com/whatis/definition/C-Sharp.**

**4. *Blender.* [Online] https://www.itnetwork.cz/blender/zaklady/uvod-do-tvorby-3d-grafiky-v-blenderu.**

**5. *Gimp.* [Online] https://www.slunecnice.cz/sw/gimp-windows/.**

**6. *dostupnysoftware.cz.* [Online] [Citace: 14. 4 2024.] https://www.dostupnysoftware.cz/produkty-zdarma/61-audacity.html.**

Seznam obrázků

^^^ zde bude seznam obrázků + stránka, kde se nachází (automaticky generovaný) -> buďte od té lásky a po vygenerování upravte font tak, aby to byl jednotný styl s ostatními (doporučuji využívat předvytvořené styly)

Obsah média

Zde přidejte stručně adresářovou strukturu (např jako víceúrovňový seznam) pro všechny důležité soubory na přiloženém médiu,

Médium by mělo být fyzicky označené **jménem, třídou, školním rokem!** Zároveň by médium mělo být v dokumentaci zajištěno tak, aby nevypadávalo, ale zároveň aby se dalo vyndat a použít.

Médium by mělo obsahovat následující:

* Případný export databáze
* Spustitelný build (nebo aspoň odkaz, kde se nachází spustitelná verze)
* Dokumentace v PDF + nějakém dalším editovatelném formátu (docx, odt…)
* Prezentace připravená k obhajobě

**Závěrečné poznámky:**

* Dokumentace může obsahovat různá poděkování
* Před exportem do PDF nechte znovu přegenerovat všechny generované seznamy a zkontrolujte, že je vše v pořádku
* Před tiskem si dokumentaci exportujte do PDF a zkontrolujte odsazení atd
* Dokumentace může být černobílá
* Dokumentace může být tisknutá oboustranně nebo jednostranně
* **V pololetí se dokumentace netiskne!**
* Vytištěná dokumentace by měla být svázána kroužkovou vazbou s průhlednou přední stranou a neprůhlednou stranou zadní (barva zadní strany a vazby je na vás)

**V případě dotazů k dokumentaci kontaktujte vedoucího práce nebo vyučujícího předmětu Projekty!**