



SPŠ ÚSTÍ NAD LABEM

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 15.1. 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 řešerše o konkurenci i jakou roli se hráli 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
1 Rešerše	8
1.1 Rimworld	8
1.1.1 Popis	8
1.1.2 Inspirace	8
1.2 Age of Empires II	8
1.2.1 Popis	8
1.2.2 Inspirace	8
1.3 Bad North	9
1.3.1 Popis	9
1.3.2 Inspirace	9
2 Technologie	10
2.1 Unity	10
2.2 JetBrains Rider	10
2.3 C#	10
2.4 Blender	10
2.5 Gimp	10
3 Praktická část	11
3.1 Návrhy	11
3.1.1 Shader graf budov	11
3.2 Produktizace	11
3.2.1 Kontrola průchodnosti mapy	11
3.3 Popis pro uživatele	15
3.3.1 Výběr jednotek	15
3.3.2 Hra	15
Závěr	18
Použitá literatura	19
Seznam obrázků	20
Seznám kódů	21

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 systému je příležitost vyzkoušet spoustu různých druhů prací.

Po úspěšném nainstalování 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 zastavenost 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 bodama. 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.

Modely ve hře budou vytvářené i animování pomocí 3D modelovacího programu Blender. Texturování právě z kratší doby na vývoj bude řešeno alternativním způsobem, například pomocí shaderů.

Dokončený projekt bude zveřejněn ke stažení.

1 Rešerše

1.1 Rimworld

1.1.1 Popis

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.

1.1.2 Inspirace

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 management celé kolonie je skvělý.



Obrázek 1 - snímek ze hry Rimworld - [odkaz](#)

1.2 Age of Empires II

1.2.1 Popis

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.

1.2.2 Inspirace

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 - [odkaz](#)

1.3 Bad North

1.3.1 Popis

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 ostrůvků se pak stane krvavé bojiště.

1.3.2 Inspirace

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.

Zaujalo mě i místo odehrávání bojů, které jsou na malém vygenerovaném ostrovu. Ten doplňuje styl grafiky.



Obrázek 3 - snímek ze hry Bad North - [odkaz](#)

2 Technologie



2.1 Unity

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 možností, protože má s tímto programem zkušenosti a dříve už používal. Značná výhodou je také používání C# vhodné pro autora.

Obrázek 4 – Logo Unity – odkaz



2.2 JetBrains Rider

JetBrains Rider je multiplatformní editor C# skriptů, 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 – odkaz



2.3 C#

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 skripty také využívá C#.

Obrázek 6 – Logo C# – odkaz



2.4 Blender

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 – odkaz



2.5 Gimp

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.

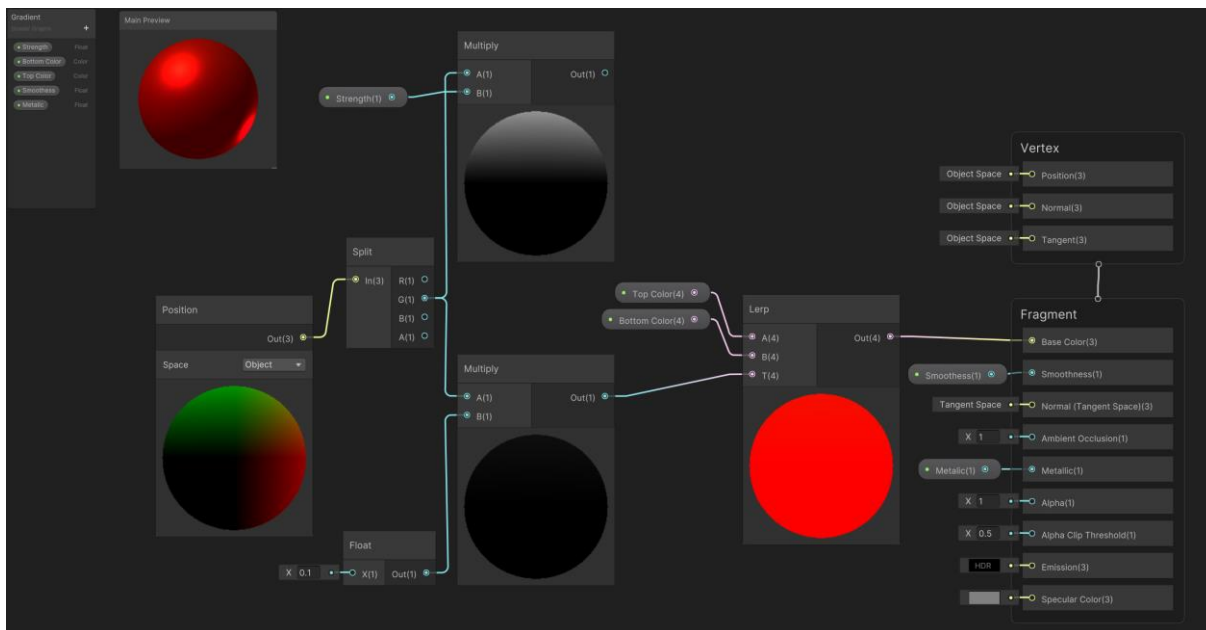
Obrázek 8 – Logo Gimp – odkaz

3 Praktická část

3.1 Návrhy

3.1.1 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.



Obrázek 9 - Graf shaderu budov – vlastní obrázek

3.2 Produktizace

3.2.1 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 očekávaný 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.

```

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;
7.
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 &&
neighborY < obstacleMap.GetLength(1)) {
16.                            if (!mapFlags[neighborX, neighborY] &&
!obstacleMap[neighborX, neighborY]) {
17.                                mapFlags[neighborX, neighborY] = true;
18.                                queue.Enqueue(new Coord(neighborX,
neighborY));
19.                                accessibleTileCount++;
20.                            }
21.                        }
22.                    }
23.                }
24.            }
25.        }
26.    }
27.
28.    int targetAccessibleTileCount = (mapSizeX * mapSizeY -
currentObstacleCount);
29.    return targetAccessibleTileCount == accessibleTileCount;
30. }

```

Kód 1 - Flood-fill algoritmus

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.     }
15.
16.     protected abstract void UpdateState();
17.
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.     }
```

```

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.}

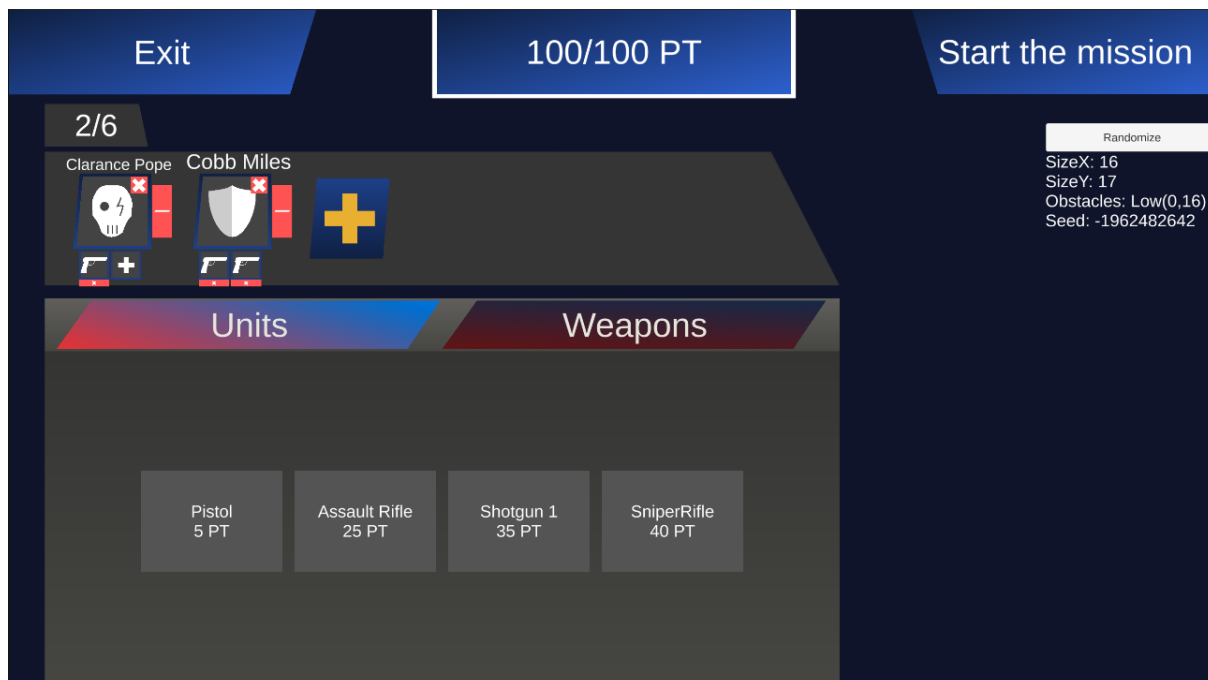
```

Kód 3 - Stav jednotky při přebíjení

3.3 Popis pro uživatele

3.3.1 Výběr jednotek

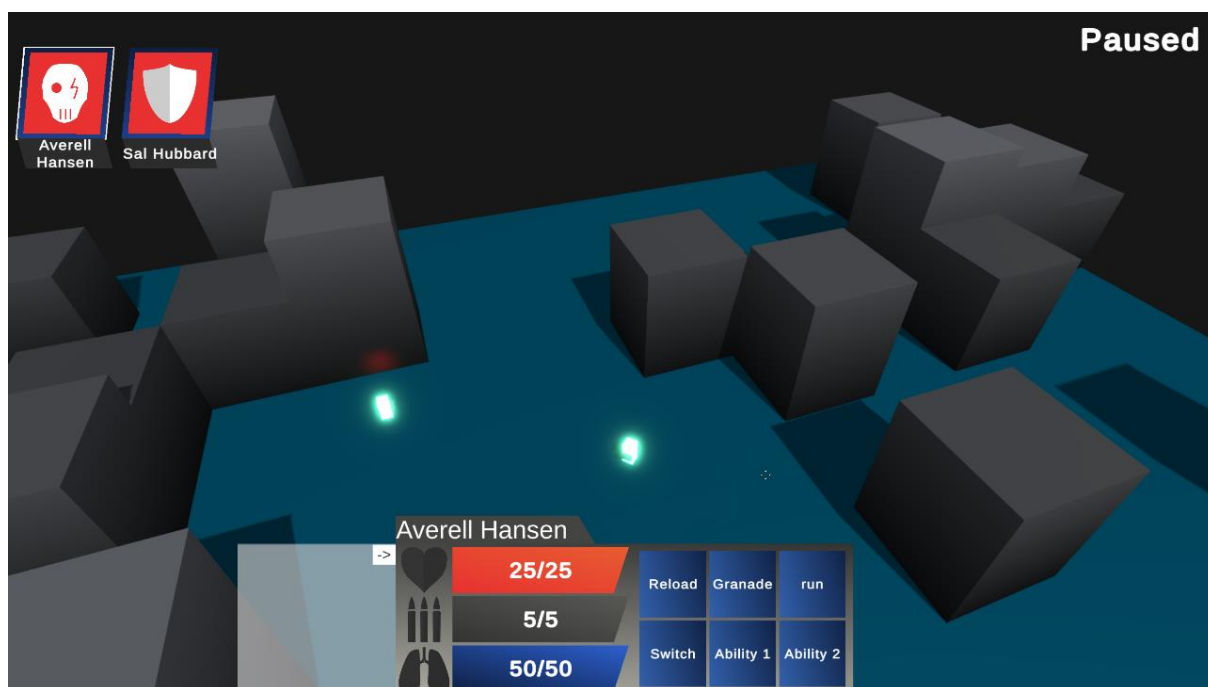
Obrazovka kde si uživatel vybírá vlastní tým, který pošle do akce. Je omezen body. Každá položka stojí několik bodů. Cena položky naznačuje sílu položka a jak užitečná může být. K lepšímu vybavení se uživatel dostane postupným hraním v pozdějších fázích hry.



Obrázek 10 - Ukázka výběru jednotek – vlastní obrázek

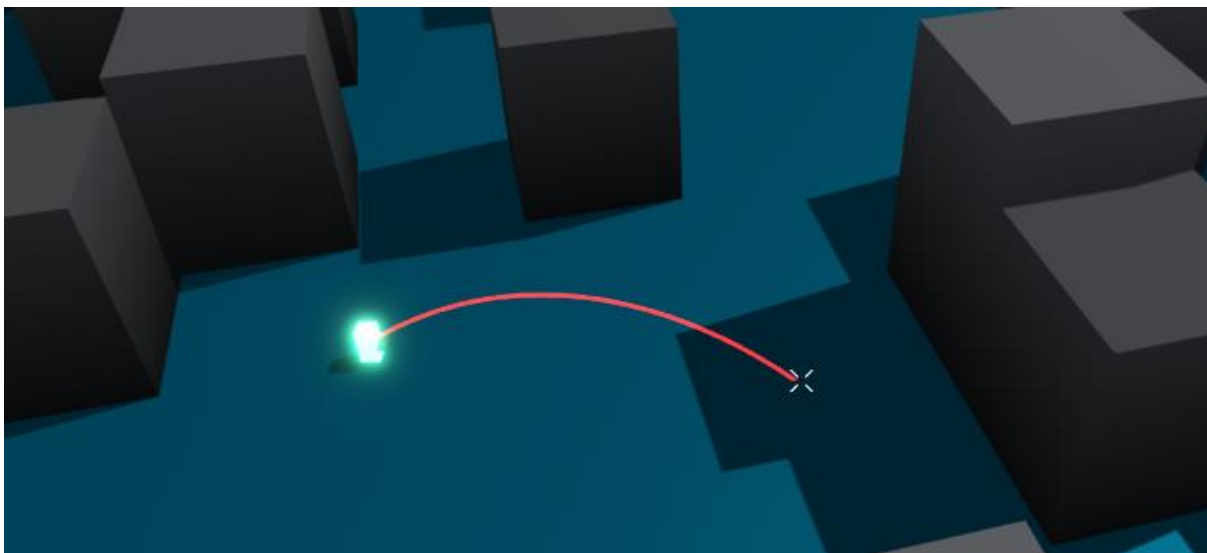
3.3.2 Hra

Samotná hra, kde se objeví jednotky hráčem zvolené na hracím poli. Jednotky tedy pomocí hráče plní úkoly právě zadané jako „běž“, „střílej na nepřítele“, „hod' 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.



Obrázek 11 - 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.



Obrázek 12 - ukázka trajektorie granátu – vlastní obrázek

Závěr

Aktuální stav projektu je na dobré cestě. Pohyb kamery je jednoduchý a umožňuje pohyb kamkoli uživatel potřebuje vidět. Jednotky mají své příkazy i jako běh, přebíjení nebo granát s ukazatelem dopadu. Mapa je celá náhodně generovaná a umožňuje mi obrovské množství kombinací map s různým množstvím zastavění plochy.

I UI se podařilo sjednotit pod jednoduchý styl s vlastníma shaderama i ikonami. Bohužel se vývoj prodloužil a nečekané zbrzdění znamenalo uchýlit se k několika kompromisům, které ale nepřekáží dalšího postupu vývoje projektu dopředu.

Použitá literatura

1. *Wikipedia*. [Online] [https://cs.wikipedia.org/wiki/Unity_\(hern%C3%AD_engine\)](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. Github. *jména měst*. [Online] [Citace: 15. 1 2024.] <https://github.com/dominictarr/random-name>.
7. itch.io. *Sprity cursorů*. [Online] [Citace: 15. 1 2024.] <https://kaboff.itch.io/mouse-cursors-pack>.
8. *Křestní jména*. [Online] [Citace: 15. 1 2024.] <https://www.cs.cmu.edu/Groups/AI/areas/nlp/corpora/names/male.txt>.
9. *Shader návod*. [Online] [Citace: 15. 1 2024.] <https://learn.unity.com/tutorial/shader-graph-gradients-2019-3#>.
10. *Unity dokumentace*. [Online] [Citace: 15. 1 2024.] <https://docs.unity3d.com/Manual/index.html>.
11. Lague, Sebastian. Youtube.com. *Generace mapy*. [Online] [Citace: 15. 1 2024.] https://www.youtube.com/playlist?list=PLFt_AvWsXI0ctd4dgE1F8g3uec4zKNRV0.

Seznam obrázků

Obrázek 1 - snímek ze hry Rimworld - odkaz	8
Obrázek 2- snímek ze hry Age of Empires II - odkaz	9
Obrázek 3 - snímek ze hry Bad North - odkaz	9
Obrázek 4 – Logo Unity – odkaz	10
Obrázek 5 – Logo JetBrains Rider – odkaz	10
Obrázek 6 – Logo C# – odkaz	10
Obrázek 7 – Logo Blender – odkaz	10
Obrázek 8 – Logo Gimp – odkaz	10
Obrázek 9 - Graf shaderu budov – vlastní obrázek	11
Obrázek 10 - Ukázka výběru jednotek – vlastní obrázek	15
Obrázek 11 - snímek ze hry – vlastní obrázek	16
Obrázek 12 - ukázka trajektorie granátu – vlastní obrázek	17

Seznám kódů

Kód 1 - Flood-fill algoritmus	12
Kód 2 - Automat jednotek.....	13
Kód 3 - Stav jednotky při přebíjení.....	14