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](#_Toc164098678)

[1 Teoretická část 8](#_Toc164098679)

[1.1 Rešerše 8](#_Toc164098680)

[1.1.1 RimWorld 8](#_Toc164098681)

[1.1.2 Age of Empires II 8](#_Toc164098682)

[1.1.3 Bad North 9](#_Toc164098683)

[1.2 Technologie 10](#_Toc164098684)

[1.2.1 Unity 10](#_Toc164098685)

[1.2.2 JetBrains Rider 10](#_Toc164098686)

[1.2.3 C# 10](#_Toc164098687)

[1.2.4 Blender 10](#_Toc164098688)

[1.2.5 Gimp 10](#_Toc164098689)

[1.2.6 Audacity 10](#_Toc164098690)

[2 Praktická část 11](#_Toc164098691)

[2.1 Návrhy 11](#_Toc164098692)

[2.1.1 Shader graf budov 11](#_Toc164098693)

[2.2 Produktizace 11](#_Toc164098694)

[2.2.1 Kontrola průchodnosti mapy 11](#_Toc164098695)

[2.2.2 Jak nepřátelé vidí mapu 12](#_Toc164098696)

[2.2.3 Automat jednotek 13](#_Toc164098697)

[2.3 Popis pro uživatele 15](#_Toc164098698)

[2.3.1 Hlavní menu 15](#_Toc164098699)

[2.3.2 Výběr mise 16](#_Toc164098700)

[2.3.3 Výběr jednotek 17](#_Toc164098701)

[2.3.4 Hra 17](#_Toc164098702)

[Závěr 19](#_Toc164098703)

[Použitá literatura 20](#_Toc164098704)

[3 Reference 20](#_Toc164098705)

[Seznam obrázků 21](#_Toc164098706)

[Seznam kódů 21](#_Toc164098707)

[Obsah média 22](#_Toc164098708)

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

### 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.

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

### 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)

Obsah obrázku Grafika, logo, grafický design, symbol

Popis byl vytvořen automatickyUnity 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

### JetBrains Rider

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

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

### C#

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

Obsah obrázku kruh, Grafika, Barevnost, kreslené

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

Obrázek 6 – Logo C#

### Blender

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

Obsah obrázku kreslené, nářadí, umění

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

Obrázek 7 – Logo Blender

### Gimp

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

Obsah obrázku Grafika, Barevnost, umění, kreativita

Popis byl vytvořen automatickyJednoduchý program, který je užitečný na úpravu textur a spritů.

Obrázek 8 – Logo Gimp

### Audacity

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

## 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

### Ukazatel trajektorie granátu

Velký problém dělal výpočet a následné vykreslení trajektorie granátu. Protože jsem nevěděl, jakou silou bude vůbec granát muset letět, aby dosáhl svého cíle.

Další číst bylo vykreslit trajektorii. Nejvhodnější byl nástroj připravený přímo v Unity s názvem Line renderer, který je přímo pro tyto účely perfektní. Stačilo mu přidat několik bodů kudy by granát mohl letět a Line renderer je spojí čárou.

1. LineRenderer line;
2. Unit unit;
3. Vector3 destination;
4. Vector3 direction;
5. float horizontalDistance;
6. float initialVelocityY;
7. void ShowTrajectory(float angle, float maxRange) {
8. direction = destination - unit.transform.position;
9. direction.y = 0;
10. horizontalDistance = Mathf.Min(Mathf.Sqrt(direction.x \* direction.x + direction.z \* direction.z), maxRange);
11. initialVelocityY = Mathf.Sqrt(Physics.gravity.magnitude \* (direction.y + Mathf.Tan(Mathf.Deg2Rad \* angle) \* horizontalDistance));
12. line.positionCount = Mathf.RoundToInt(horizontalDistance)\*3;
13. for (int i = 0; i < line.positionCount; i++) {
14. float time = i / (float)(line.positionCount-1);
15. float t = time \* Time.fixedDeltaTime \* line.positionCount;
16. Vector3 position = unit.transform.position + direction.normalized \* (horizontalDistance \* time) + Vector3.up \* (initialVelocityY \* t -1 \* Physics.gravity.magnitude \* t \* t);
17. line.SetPosition(i,position);
18. }
19. }

### 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

Obsah obrázku text, snímek obrazovky, Multimediální software

Popis byl vytvořen automaticky

Obrázek 13 - možnosti nastavení

### Výběr mise

Obsah obrázku text, mapa, snímek obrazovky

Popis byl vytvořen automaticky

Obrázek 14 - menu výběru misí

### 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

### 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. Lze zde vidět i životy jednotky, počet nábojů v zásobníku i energii pro běh.

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

Popis byl vytvořen automaticky

Obrázek 16 - snímek ze hry

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

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. Většina cílů byla splněna i když se muselo několik mechanik zmenšit v rozsahu, než v jakém byli plánovány a zamýšleny.

Základ hry je uživatelův tým jeho jednotek. Jednotky mohou mít různé specializace se jinými atributy jako je životy, přesnost, rychlost aj. Jednotky mohou mít i 2 zbraně a mezi nimi střídat během boje. Zbraně se také liší svými atributy, ale i stylem použití. Jsou 3 základní podle kterých se udávají další, brokovnice střílí více střel, samopal v dávkách a pistole po jedné střele.

Nepřátelé se bohužel nedrží souvislé taktiky, jak jsem doufal, a spoléhají se pouze sami na sebe. Nepřátel je o poznání více a mají nižší základní atributy. Také si udržují povědomý o stavu na mapě a v jaké situaci se nachází právě nepřítel a podle toho vybírá další akce, které provede.

Mapa je podle cíle celá náhodně generovaná. Mapy se liší obtížností, velikostí a hustotou zastavění. Tudíž každý uživatel nejspíš nenarazí na stejnou mapu dvakrát. Budovy mají shader, který jsem vytvořil poprvé a povedl se. Bohužel budovy se už neliší ničím než výškou, na modely nevyšel čas.

Rozhraní a grafika byla zjednodušena, aby ladila s celkovým pocitem hry. Je tedy jednoduchá, ale lehce pochopitelná a co je potřeba zvýraznit je dobře vidět. Vše je laděné do policejních barev: modrá, červená a šedá.

Práce na projektu mě naučila spoustu věcí a prohloubila vědomosti zejména v Unity, kde jsem se naučil pracovat se scriptable object a základní tvorbu shader textur.

A navíc jsem se naučil pracovat s programem Audacity pro výtvor hlasů pro jednotky.

Projekt jako takový mi dal cenné zkušenosti do budoucího programování, zejména pracování s objektové programován. I práce s Unity a jak se využívají nástroje v něm jako fyzika předmětů, eventy, nebo i ukládání pomocí Playerprefs. Projekt ale zasahoval i do spoustu jiných odvětví jako je správa vlastního času i že se musím s volnočasovými aktivity uskromnit, i že pracuji na dlouhodobém projektu mi pomáhá se zlepšovat v syntaxi a nutí mě to si dělat pořádek v souborech, v kódu i v plánování práce.

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.**

**7. *kinguin.net.* [Online] [Citace: 14. 4 2024.] https://www.kinguin.net/cz/category/136251/rimworld-biotech-dlc-eu-v2-steam-altergift.**

**8. *steam.com.* [Online] [Citace: 14. 4 2024.] https://store.steampowered.com/app/688420/Bad\_North\_Jotunn\_Edition/.**

**9. *steam.com.* [Online] [Citace: 14. 4 2024.] https://store.steampowered.com/app/688420/Bad\_North\_Jotunn\_Edition/.**

**10. Watabou. *Github.io.* [Online] [Citace: 14. 4 2024.] https://watabou.github.io/city.html.**

**LAGUE, Sebastian. *Character Creation* [online]. 30. 10. 216 [cit. 2023-06-16]. Dostupné z:** [**https://www.youtube.com/watch?v=ZwD1UHNCzOc&list=PLFt\_AvWsXl0djuNM22htmz3BUtHHtOh7v&index=12&ab\_channel=SebastianLague**](https://www.youtube.com/watch?v=ZwD1UHNCzOc&list=PLFt_AvWsXl0djuNM22htmz3BUtHHtOh7v&index=12&ab_channel=SebastianLague)

**Samyam. *Control Rebinding with the New Input System* [online]. 11. 4. 2021 [cit. 2023-06-16]. Dostupné z:** [**https://www.youtube.com/watch?v=csqVa2Vimao&ab\_channel=samyam**](https://www.youtube.com/watch?v=csqVa2Vimao&ab_channel=samyam)

***Unity User Manual* [online]. [cit. 2023-06-15]. Dostupné z:** [**https://docs.unity3d.com/Manual/index.html**](https://docs.unity3d.com/Manual/index.html)

**CORPVS. *Electronic Music Songpack* [online]. 7. 7. 2022 [cit. 2023-06-16]. Dostupné z:** [**https://assetstore.unity.com/packages/audio/music/electronic/electronic-music-songpack-214055**](https://assetstore.unity.com/packages/audio/music/electronic/electronic-music-songpack-214055)

**Dustyroom. *Retro Aesthetics* [online]. 16. 11. 2020 [cit. 2023-06-16]. Dostupné z:** [**https://assetstore.unity.com/packages/tools/particles-effects/retro-aesthetics-79538**](https://assetstore.unity.com/packages/tools/particles-effects/retro-aesthetics-79538)

**Unity Technologies. *Partiple Pack* [online]. 8. 3. 2023 [cit. 2023-06-15]. Dostupné z:** [**https://assetstore.unity.com/packages/vfx/particles/particle-pack-127325**](https://assetstore.unity.com/packages/vfx/particles/particle-pack-127325)

Seznam obrázků

[Obrázek 1 - snímek ze hry Rimworld - (7) 9](#_Toc164098391)

[Obrázek 2- snímek ze hry Age of Empires II - (8) 10](#_Toc164098392)

[Obrázek 3 - snímek ze hry Bad North - (9) 10](#_Toc164098393)

[Obrázek 4 – Logo Unity 11](#_Toc164098394)

[Obrázek 5 – Logo Jetbrains Rider 11](#_Toc164098395)

[Obrázek 6 – Logo C# 11](#_Toc164098396)

[Obrázek 7 – Logo Blender 11](#_Toc164098397)

[Obrázek 8 – Logo Gimp 11](#_Toc164098398)

[Obrázek 9 - Logo Audocity 11](#_Toc164098399)

[Obrázek 10 - Graf shaderu budov 12](#_Toc164098400)

[Obrázek 11 - vizualizace hodnot polí 14](#_Toc164098401)

[Obrázek 12 - Hlavní menu 17](#_Toc164098402)

[Obrázek 13 - možnosti nastavení 17](#_Toc164098403)

[Obrázek 14 - menu výběru misí 18](#_Toc164098404)

[Obrázek 15 - Menu tvorby čety 18](#_Toc164098405)

[Obrázek 16 - snímek ze hry 19](#_Toc164098406)

[Obrázek 17- ukázka trajektorie granátu 19](#_Toc164098407)

Seznam kódů

[Kód 1 - Flood-fill algoritmus 13](#_Toc164098386)

[Kód 2 - Automat jednotek 15](#_Toc164098387)

[Kód 3 - Stav jednotky při přebíjení 16](#_Toc164098388)