

Gymnázium, Praha 6, Arabská 14



Teamový Ročníkový Projekt

Robert Ambrož, Alexandr Belčenko, Tadeáš Jůn, Timofej Kiselev

Karoff

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne _____ Robert Ambrož _____

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne _____ Alexandr Belčenko _____

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne _____ Tadeáš Jůn _____

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne _____ Timofej Kiselev _____

Anotace

Karoff is a digital board game, created in the Unity game engine, by a team of four. The game, which is played by two players, either on one computer or across a network, consists of the players building tiles of terrain and buildings on said terrain. Built buildings, such as the Barracks, the Lumbermill, or the Sandstone Mine, give players bonuses in the form of resources (used on the construction of more buildings), or victory points. The first player to reach 25 victory points wins the game.

Karoff features two game modes (local multiplayer and network multiplayer), pixel art graphics, and dynamic chiptune music. Unity's multiplayer system UNet was used for the network multiplayer mode, while the graphics were made in Aseprite, and the music in FL Studio.

Karoff je digitální desková hra, vytvořená čtyřčlenným týmem v herním enginu Unity. Tuto hru hrají dva hráči, na jednom počítači, nebo přes internet. Hlavní pointou hry je stavba kostek terénu, a následně budov na tento terén. Budovy, jako například kasárny, pila, nebo pískovcový důl, dávají hráčům bonusy ve formě surovin (které jsou později použity na výstavbu dalších budov), nebo vítězných bodů. První hráč, který dosáhne 25 vítězných bodů, vyhrává hru.

Karoff obsahuje dva herní módy (lokální multiplayer, a multiplayer přes síť), pixel artovou grafiku a dynamickou chiptune hudbu. Síťový multiplayer byl vytvořen přes Unity UNet systém, grafika byla kreslena v programu Aseprite, a hudba složena v FL Studio.

Zadání

Cílem projektu je vytvořit funkční digitální deskovou hru s prvky stavění měst a okolních terénů v herním engine Unity 2019.1+, a programovacím jazyku, který tento engine používá - C#. Hra bude obsahovat retro pixelart grafiku a složité herní systémy, jako například počítání vítězných bodů podle několika podmínek stavby.

Alexandr Belčenko - hudba, zvukové efekty, vizuální efekty, teoretická dokumentace.

Tadeáš Jůn - game design, pravidla hry, pixelart grafika, případná pomoc s jazykem C# a herním engine Unity 2019.1+.

Timofey Kiselev - práce s herním engine, programování herním systémů, případná pomoc s game designem a pravidly hry.

Obsah

Anotace.....	3
Zadání	4
Obsah.....	5
1. Úvod.....	1
1.1 Původní plán	1
1.2 Hlavní myšlenka Karoff.....	2
2. Manuál.....	3
3. Pravidla hry	4
3.1 Budovy	4
4. O Unity	6
4.1 C# a Visual Studio	7
5. O vývoji	8
5.1 Začátek.....	8
5.2 Zadání	8
5.3 GitHub, Unity a VisualStudio	8
5.4 První prototyp	9
5.5 Osnova hry	10
5.6 Pixel Art Grafika	10
5.7 Testovací projekty	11
5.8 Sjednocení.....	11
5.9 Dokončení	12
6. Hudba	14
7. Local Multiplayer	16
8. Network Multiplayer	21
8.1 Port forwarding (Přesměrování portu).....	22
8.2 Hamachi a AnyDesk.....	22
9. Finální Program	25
10. Ovládání	29
11. Budoucí plány	30
12. Závěr	31
13. Nástroje	33

14.	Seznam Obrázků	34
15.	Zdroje.....	35

1. Úvod

1.1 Původní plán

Začátek našeho projektu se dá datovat až k dubnu roku 2019, kdy jsme složili tým Tadeáš Jůn, Timofey Kiselev a Alexandr Belčenko, a začali jsme vymýšlet ročníkový projekt, na kterém jsme chtěli pracovat během letních prázdnin.

Podstata projektu, tehdy nazvaného Despair, byla vytvořit MMORPG (massively multiplayer online role playing game), hru, která by běžela čistě přes web. Plánovalo se to uskutečnit pomocí React (<https://reactjs.org/>) a Flask (<https://flask.palletsprojects.com/en/1.1.x/>), ale nakonec ani jedno z toho v Karoff použito nebylo. Hráči by se mohli přihlásit či registrovat pomocí jednoduchého signup a login web rozhraní. Po úvodní registraci by byla hráči automaticky přiřazená strana – člověk, nebo démon.

Během jedné “partie”, která by probíhala přibližně tři měsíce, by se hráči snažili splnit hlavní úkol své strany. Lidé mají za úkol obránit Blackfell, jejich město, po celou dobu hry, zatímco démoni se toto město snaží dobít. Pro stranu lidí je největší problém sbírání surovin na obranu města; právě z tohoto důvodu se musejí vydávat na nebezpečné výpravy mimo městské hradby, kde na ně čekají mnohá nebezpečí, včetně hráčů hrající za demony.

Zatímco lidé sbírají suroviny, které poté využívají na upgrade (vylepšení) města Blackfell, démoni loví lidi, kteří se odvážili vystoupit z města ven, a pořádají skupinové nájezdy na městské hradby.

Jedna z původních komplikací projektu jako ročníkové práce byl fakt, že jsme nemohli používat nic, co jsme nevytvořili my, a co nebylo volně dostupné s otevřenou licencí. Toto byl jeden z důvodů (ale rozhodně ne ten hlavní), proč z Despair nakonec sešlo.

Rozhodli jsme se rozdělit si práci na první fázi projektu následovně:

- Tadeáš Jůn: grafika, game design
- Timofey Kiselev: back-end
- Alexandr Belčenko: front-end, hudba

Očividně by tento projekt vyžadoval hodně práce, protože by bylo potřeba vyřešit serverovou backend stránku hry, frontend rozhraní, mnoho game design problémů, a musela by se udělat relativně velká marketingová kampaň okolo hry Despair (protože aby byla hra zábavná, potřebovala by aspoň 100 hráčů na každé straně). Z těchto důvodů jsme chtěli na Despair dělat již přes velké letní prázdniny.

Bohužel z tohoto plánu nakonec sešlo, protože nebylo jisté, zdali budeme moct být v našem vybraném týmu. Jelikož tu byla možnost, že se nám tým na začátku roku rozhodí, rozhodli jsme se se začátkem práce počkat až na září. Ovšem tou dobou už nebyl dostatek času na projekt tak velkého rozměru jako původní Despair, a proto jsme se rozhodli zadání podstatně změnit, a stáhnout jeho scope, a to na momentální hru Karoff. Nicméně, Despair rozhodně hodně ovlivnil vývoj projektu Karoff, jak z game designového hlediska, tak z hlediska práce v týmu (a bohužel i negativně z hlediska morálky).

Myslíme si jako tým, že hra Despair má stále velký potenciál, a pravděpodobně na ní budeme v budoucnu společně pracovat. Je ovšem samozřejmé, že takový projekt bude potřebovat hodně času a financí, aby byl schopen fungovat. Doufáme, že Despair vytvoříme a vydáme v několika příštích letech, protože by reálně mohl mít velký úspěch u fanoušků web-based MMORPG her.

1.2 Hlavní myšlenka Karoff

Jelikož z Despair se muselo sejít, naše konečné zadání projektu obsahuje tvorbu “digitální deskové hry s prvky stavění měst a okolních terénu”. Náš hlavní cíl byl vytvořit hru, která se opravdu hrála co nejvíce jako desková hra. Proto jsme zvolili grafiku “kostek”, které představují terén a města. Jednou z inspirací byly oblíbené deskové hry, jako Carcassonne, či Osadníci z Katanu.

Valná většina deskových her se hraje ve více lidech, a proto jsme samozřejmě museli nějak vyřešit multiplayer, ať už lokálně (dva hráči hrají u jednoho počítače), nebo i po síti (každý z dvou hráčů má svůj vlastní počítač). Bylo nám jasné, že ideál by byl nabídnout uživatelům oba dva možné módy, a proto jsme rozdělili práci na local mode a network mode. Nakonec se nám podařilo zprovoznit oba herní módy, a hráči si mohou vybrat v hlavním menu hru. Každému z módů se později věnuje jedna kapitola.

2. Manuál

Jeden z velkých rozdílů typické počítačové hry a typické (evropské) deskové hry je to, že desková hra obsahuje manuál, nebo-li pravidla, prakticky dokument, který hru jasně vysvětluje hráčům, zobrazuje všechny hrací části, ukazuje situace příkladové partie a případně udává příběh a atmosféru hry.

Při zvažování, jestli taková pravidla zahrnout do Karoff, rozhodli jsme se vrátit se k našemu cíli, co nejvíce se přiblížit deskové hře, a proto jsme pravidla vytvořili. Manuál tvořil Tadeáš Jůn v designovém softwaru Figma.

Figma je program zdarma, který umožňuje jednoduchý workflow UX a UI designu.

Ve Figmě je vytvořen celý manuál hry, který obsahuje:

- úvodní stránku s logem a příběhem hry
- stránku s vysvětlením pravidel hry
- stránky s popisem a vysvětlením všech deseti budov
- stránku s popisem biomů a jejich ikonami

Tento manuál jsme původně chtěli vytisknout na silný papír, který se používá u pravidel fyzických deskových her, ale kvůli karanténě jsme nakonec manuál odevzdali jen jako PDF soubor.

Figma byla také použita na tvorbu loga Karoff, které používá barvy #2d2c2c, a #ff652f, a zároveň #6c2f2f a #2e2e6a, což jsou barvy jednotlivých hráčů ve hře. Logo zobrazuje stylizované K, které má přidáný stín, což dodává logu hloubku, a iluzi 3D prostoru.

Logo je použité na první stránce manuálu, a samozřejmě jako ikona hry.



3. Pravidla hry

Jednou z hlavních inspirací pro koncept hry Karoff byla světoznámá hra Carcassonne.

Každá partie Karoff probíhá v několika desítkách tahů, a každý tah se skládá ze dvou fází. První fázi, stavění, hraje jen jeden ze dvou hráčů, ale jeho volby ovlivní následující vývoj hry pro oba hráče, zatímco druhá fáze, suroviny, se vždy vztahuje na oba hráče na základě jejich voleb v první fázi.

V první fázi příslušný hráč může:

- 1) přistavit kostku terénu, a to tak, že klikne na již postavený terén, a následně na jednu z přisedících kostek, které se objeví. Tato kostka bude mít stejný biom, jako kostka terénu, na kterou hráč prvně klik (např.: k poušti se přistaví nová poušť, k tundře nová tundra), nebo
- 2) postavit budovu na jednu z volných kostek terénu, postavených jakýmkoliv hráčem. Budovy jsou probrané do detailu níže a patří jen hráči co je postavil.

Druhá fáze probíhá naprosto automaticky, tj. nevyžaduje žádný input od jakéhokoliv z hráčů. V této fázi se oběma hráčům přidají suroviny, a to podle toho, kolik budov určitého typu má příslušný hráč postavených.

Po skončení druhé fáze se spustí nový tah, znovu probíhá první fáze, ale hraje ji nyní druhý hráč (pokud v minulém tahu stavěl terén/budovu červený hráč, nyní staví modrý, a opačně).

Tyto dvě fáze se opakují, dokud jeden z hráčů nedosáhne určeného počtu bodů, v ten moment hráč vyhrává.

3.1 Budovy

Deset typů budov, které Karoff obsahuje, jsou snad nejdůležitější částí hry. Hráči mohou budovy stavět na kostky terénu, čímž a) si zvýší počet surovin získaných ve druhé fázi tahu, nebo b) ihned získají určitý počet vítězných bodů.

Každá budova má svou cenu, udanou v libovolném počtu čtyř typů surovin – Sand (písek, surovina těžená v pouštích), Stone (kámen, surovina těžená v horách), Ice (led, surovina těžená v tundrách), či Lumber (dřevo, surovina těžená v lesích). Aby hráč mohl danou budovu postavit, musí vlastnit počet surovin, který budova stojí

Každý typ budovy je v Unity tzv. ScriptableObject, což je typ scriptu, který jednoduše umožňuje vytvářet nové budovy podle předem dané šablony. Díky tomu je relativně jednoduché do hry Karoff přidat nový typ budovy, aniž by se musel výrazně měnit kód.

Každý ScriptableObject budovy obsahuje tyto proměnné:

- BuildingName (jméno budovy),
- BuildingIcon (objekt typu Sprite, což je prakticky řečeno jakýkoliv importovaný .png nebo .jpg obrázek),
- Prices (čtyřprvkové pole, které určuje cenu budovy ve všech surovinách)
- PossibleBiomes (pole, které určuje, na které biomy ve hře se dá daná budova postavit)
- Gain (pětprvkové pole, které určuje, co hráč získá, když budovu postaví)

Systém ScriptableObjectů také umožňuje velmi lehkou editaci – například pro balancování hry stačí přepsat v jednom z polí prvek, a budova se automaticky upraví.

4. O Unity

Naši práci jsme dělali především v Unity, což je engine pro vývoj her. Velmi nám pomohl, jelikož podporuje C#. Pracuje se s ním lehce a vše máte po ruce. Jelikož tento program tvoří velkou část našeho projektu rád bych vám trochu přiblížil jeho historii a jak vlastně funguje jako takový. Unity může být jak pro amatéry, tak i pro zkušené programátory her

Herní engine Unity byl vytvořen firmou Unity Technologies, která má sídlo v San Franciscu, ale je americko-dánskou společností. Zpočátku se firma jmenovala Over the Edge Entertainment a byla vytvořena v roce 2004 Davidem Helgasonem, Davidem Francisem a Joachimem Antem. Nezačali ovšem hned z počátku jako herní engine. Vytvářeli dohromady hru jménem Gooball. Hra, avšak neměla komerční úspěch, ale členové viděli potenciál v usnadnění vytváření her, a tak změnili jejich cíl na vytvoření enginu pro ostatní vývojáře. Velký pozdvih získali při představení iPhone, protože většina vývojářů se zaměřovala na dělaní her na stolní počítače a konzole. Udělali tak podporu pro tvorbu her na iPhone. Velká většina her právě na iPhone byla vytvořena v jejich enginu a bylo tomu také dlouho poté. V Roce 2007 se následně společnost přejmenovala z Over the Edge Entertainment na Unity Technologies.

Nyní jejich snažení posunuli na 25 platforem, a to včetně virtuální reality. V roce 2010 vytvořili Asset store, kde mohli uživatelé nahrávat jakýkoliv svůj obsah a ostatní si pak následně mohli koupit jejich výrobky. Tvorba her byla tak ještě dostupnější, jestliže jste neuměl udělat hudbu do hry, koukli jste se do marketu s assetama a vybral jste si co se vám líbí. Zaplatil člověku, který muziku složil a vložil jste si to do vlastní hry, aniž by vás trápila jakákoliv autorská práva. Již v roce 2012 Unity hlásilo milion uživatelů, z nichž tři sta tisíc uživatelů používalo program alespoň jednou za měsíc. Následně v roce 2016 se engine pyšnil pět a půl milionem registrovaných uživatelů.

Všichni tvůrci chtěli dopřát tvorbu her co největší škále lidí, a to se jim také povedlo. Hlavním tahákem Unity byla jednoduchost a přehlednost. Nemuseli jste být kdejaký IT specialista abyste udělali hru na počítač, konzoli či mobilní telefon. Unity je velmi přátelský i pro lidi, kteří s tímto oborem nemají absolutní zkušenosti a chtějí si to vyzkoušet, jestliže u toho zůstali dostatečně dlouho mnohé se naučili a určitě netušili, jak lehké to může být, když máte po ruce úžasný program, který je velmi snadný na ovládání a tvorbu her

Některé hry, které hrajeme si ani neuvědomujeme, že byli stvořeny právě v Unity. Třeba takové Pokemon Go. Obrovský trhák, který donutil kohokoliv vytáhnout paty z baráku a jít chytat pokémony ven. Populární indie hra Ori and the Blind Forest, je také vytvořen pomocí tohoto enginu. Unity má velký dopad na tvorbu indie her, jelikož ho používá skoro každá skupinka indie vývojářů, ale na špičkové hry lidé používají mnohem dražší a lepší engine, ale těchto firem není tolik. Několik výjimek samozřejmě i zde existuje: například karetní hra Hearthstone je vytvořena na Unity.

Unity jsme pro Karoff vybrali proto, že se jedná o volně přístupný engine, je relativně jednoduchý na naučení, na internetu se dá najít snad nejvíce návodů (podstatně více než okolo jiných game enginů). Unity také podporuje Mac i Windows, což byl pro nás důležitý faktor, protože jeden ze členů pracuje na Macu.

4.1 C# a Visual Studio

C# (čteno C Sharp) je programovací jazyk vytvořený firmou Microsoft okolo roku 2000. Cílem tohoto jazyku je nabízet programátorům jednoduché, moderním a obecně využitelné rozhraní pro object-oriented programming.

Microsoft Visual Studio je IDE (integrated development environment) vytvořené ruku v ruce s C#, taktéž Microsoftem.

Tento jazyk a IDE je nezbytnou součástí herního enginu Unity. V minulosti Unity podporovalo i JavaScript, ale od roku 2017 musí standartny všechny být scripty napsané v jazyce C#.

5. O vývoji

5.1 Začátek

Na začátku, po konci letních prázdnin, jsme se museli rozhodnout o čem Karoff hra vlastně bude. Samotný název Karoff vznikl až později když jsme to už věděli, ale na začátku roku jsme potřebovali vymyslet deskovou hru na počítač. Už jsme si byli docela jistí, že budeme pracovat v Unity a taky jsme měli jistotu, že ho nebude problém používat ze zkušeností ročníkových prací z minulého roku. Řada známých stolních počítačových deskových her je dělaná právě v Unity, například již zmiňovaný Hearthstone, nebo Magic: The Gathering Arena. Jsou to především karetní hry a tomu jsme se chtěli vyhnout. Rozhodli jsme se vydat cestou stolních her toho typu, kde se hrací pole staví za průběhu samotné hry. Takových her je celá řada, ale hlavní inspirací pro nás byl právě Carcassonne a Osadníci z Katanu.

5.2 Zadání

Tým byl určen, ale téma ještě nebylo úplně jasné, a tak jsme šli do školní knihovny vymýšlet zadání práce. Zadání práce, které jsme nakonec měli můžete přechíst na začátku dokumentace, nebo na GitHubu.

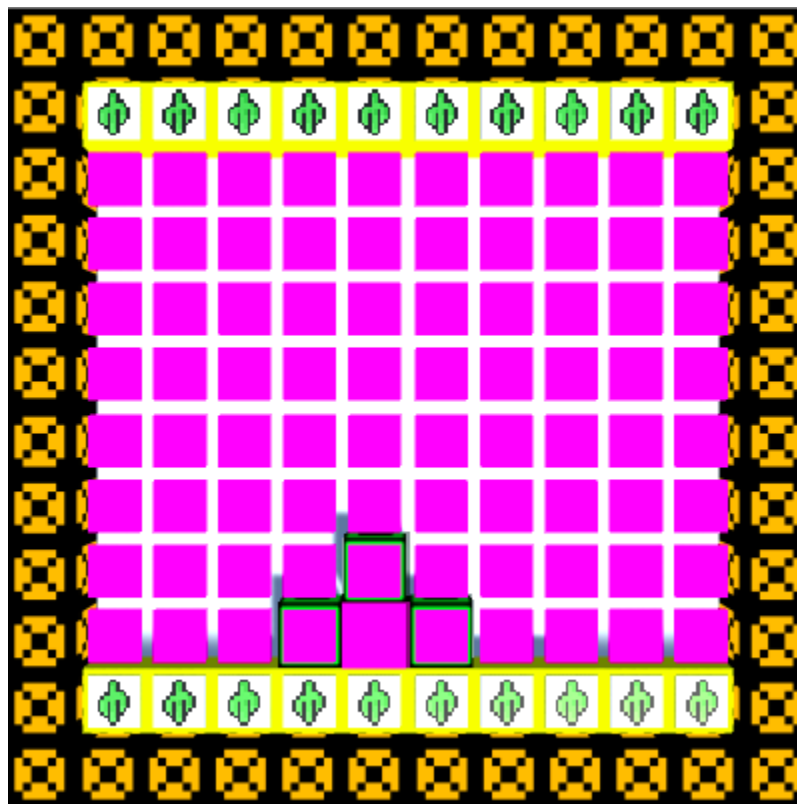
5.3 GitHub, Unity a VisualStudio

S GITem jeden člen týmu už pracoval a zbytek už alespoň slyšelo o GitHubu, takže s touto částí nebyl žádný velký problém a v případech kdy nějaký nastával na internetu už bylo známé řešení. Část týmu dokonce pro zjednodušení používala program GitHub Desktop, což je program, který svým GUI výrazně zjednodušuje základy práce s GITem.

Dále polovina týmu nainstalovala Unity verze lehce nad 2019.3 a Unity Hub. Unity Hub je program na spravování Unity projektů a verzí Unity. Unity používal hlavně 2 jazyky, Unity Script (speciální JavaScript pro unity) a C#. Používal i nějaké další jazyky, ale ty nikdo moc nepoužíval a teď se nepoužívá ani Unity Script, takže volba jazyku byla jasná. Jako IDE bylo zvoleno VisualStudio, protože Unity s ním velmi dobře spolupracuje a samo o sobě je to taky dobrý IDE od Microsoftu. Ve Visual Studiu se psaly všechny C# skripty pro Karoff.

5.4 První prototyp

Jako první jsme se dohodli jen na obecných principech hry, a ne konkrétně na tom, jak bude fungovat. Tudíž první prototyp hry se velmi lišil od finální verze. Například jako asi nejvýraznější rozdíl je to, že byl 3D. Vidět příklad toho jak vypadal můžete na obrázku. Jako druhý byla snaha udělat vše co ve hře je nějakou variací scriptable object re prezentujícího biome. Dalším rozdílem bylo to, že do hry bylo hodně obtížné přidat stavby budov a skoro nemožné přidat multiplayer. Hra obsahovala nejjednodušší mechaniky a už šlo na osnově toho říct co se má změnit a co bude lepší nechat. Hlavní změna pak byla to, že místo 3D se Karoff stal 2D hrou, protože 3D tam bylo naprosto zbytečné a jen způsobovalo zbytečné problémy. Ve hře, která je naprosto pořád jen pohled svrchu na hrací plochu je 3D naprosto zbytečný. Další změna byla to, že v dalších verzích byl odstraněn rámeček reprezentující desku, ve které se hra měla hrát a 4 startovní čtverce se generovaly na náhodných pozicích, místo toho, aby se generovaly jako krychle opakovaně v řadách po stranách hrací desky. Taky samotná struktura jak kódu, tak objektu tvořených v editoru Unity je nakonec mnohem lepší a lépe editovatelná, než byla v prvním prototypu. Ovládání se ale skoro nezměnilo.



1 – První prototyp

5.5 Osnova hry

Osnovou hry nazývám už hratelnou verzi hry, kde ještě není Network Multiplayer. Vizuálně se podle této verze dalo jen těžko poznat rozdíl od aktuální verze, ale zároveň v ní chyběla spousta velice důležitých funkcí. Ta verze už používala hotovou pixel art grafiku místo toho, aby jako prototyp užívalo to něco málo co už bylo hotové a jinak jednoduché křížky, prázdné textury, čistou stejnorodou barvu atd. Nebylo menu ani hudba. Nebyla dost dobře upravitelná pro Network Multiplayer, ale lokálně na jednom počítači by se již hrát dala, i když v případě, že jeden z hráčů vyhraje tak to nemá jak zjistit. Hra jen začínala od začátku a vypisovala výherce do konzole. Taky měla řadu chyb, které se postupně opravovaly.

5.6 Pixel Art Grafika

Součástí zadání Karoff bylo vytvoření “retro pixel art grafiky”; pixel art je druh umění, kde je práce složená jen z několika pixelů, hodně zvětšených. Časté rozlišení obrázků jsou 16x16 pixelů (toto rozlišení má například grafika hry Minecraft), 32x32 pixelů, nebo 64x64 pixelů. Pro větší práce, jako například pixel art obrazy krajin, měst, či jiných scén, se již volí jiné, nečtvercové rozlišení

Ve vývoji grafiky pro Karoff byl použitý editor Aseprite. Pro kreslení pixel artu se dá použít nespočet programů – často používaný na například Adobe Photoshop. Nicméně Aseprite, který je relativně mladý program, byl vytvořený přímo pro tvorbu pixel artu, a proto je také nejvíce přizpůsobený.

Hlavní součástí grafiky jsou ikonky všech biotů a budov ve hře. Celkem jich je 14 - kaktus pro biot pouště, sněhová vločka pro biot tundry, kámen pro biot hor, strom pro biot lesu, a pro budovy to jsou ikonky dřeva, pyramidy, kapky vody, dolařské helmy, masky morového doktora, iglů, kokosu, krumpáče, cílové vlajky, a meče.

Ikonky biotů byly zároveň pro přehlednost použité i jako ikonky surovin, které se v daných biomech dají sbírat/těžít.



2 - Ikonky

5.7 Testovací projekty

Když část týmu pracovala na osnově hry jiná část vytvořila nový projekt, ve kterém testovala, jestli UNet API pro Network Multiplayer vůbec bude fungovat. První testovací projekt byl snad tím nejjednodušším, co se s multiplayerem dá pokusit udělat, ale i to bylo dost obtížné poprvé udělat. Nakonec vznikl minimalistický projekt s nejzákladnějšími mechanikami, na jehož příkladu se dalo vyzkoušet, jestli to vůbec půjde s UNet API udělat. Tehdy ještě alespoň jednomu z týmu fungoval port forwarding takže to šlo dost jednoduše vyzkoušet a zjistit, že to funguje.

Tento první testovací projekt obsahoval 2 scény. První měla předvytvořené autory API minimalistický GUI umožňující pár funkcí, ze kterých podstatné byli začít hostovat, nebo zadat IP adresu a připojit se k hostu. Ve druhé scéně byla jedna předem vytvořená krychle (tehdy ještě nebylo zcela jasné, jestli se ke 3D nakonec nevrátí, i když nebyl důvod), ale na krychli byl pohled fixovaný přímo ze shora, takže vůbec nemusela to být krychle. Vidět byl jen čtverec. Čtverec změnil barvu a když hráč kliknul na ten čtverec vpravo od něj se vytvořil další. Tentokrát modrý. Bylo to nastavené tak, aby to počítalo tahy a hráč jehož kliknutí tvořilo nový čtverec se pokaždé střídal. Taky na každém čtverci byl skript, který pamatoval, jakou má barvu a v případech, že se client odpojí a znovu připojí, by se mu načetl i stav toho čtverce jako jeho aktuální barva.

Druhý testovací projekt už byl zcela 2D a zakládal se na už hotové osnově hry a testovacím projektu číslo jedna. V tomto projektu bylo cílem aplikovat úspěch prvního testovacího projektu na základní funkce hotové osnovy hry, jelikož udělat to hned v něm bylo velice obtížné a nastal zmatek. Avšak v tomto testovacím projektu se povedlo pomalu a postupně udělat základy na kterých se stavěl zbytek Network Multiplayeru. Ve chvíli, co uměl ty nejzákladnější funkce (přibližně na úrovni prvního prototypu), se začal vkládat do osnovy hry a propojovat s ní.

5.8 Sjednocení

Nejdříve byly vytvořené další 2 scény. Jedna byla vytvořena pro menu a druhá pro samotný multiplayer. Vytvořila se další složka MultiplayerScripts ve které se nachází kód podstatný pro network multiplayer. Network multiplayer využívá některé skripty ze složky Scripts ale naopak se to skoro neděje. Většina kódu v nové složce jsou potomky classy NetworkBehaviour a ne

MonoBehaviour, potomky, které je většina běžných Unity skriptu čteně většiny ze složky Scripts. Tyto Multiplayer skripty se liší jen minimálně od svých obyčejných verzí, ale ty změny jsou velice podstatné. Pak ale tu jsou skripty, co nemají žádnou alternativu mimo network multiplayer. Takové jsou například skripty, co ukládají momentální stav objektů, nebo jsou komponentem GameObject prefabu. Taky se dost výrazně změnily některé managery jako například TurnManager jak GameObject tak skript MultiplayerTurnManager který je jeho komponentem. Trochu podrobněji jak to vše souvisí však budu popisovat v jedné z následujících kapitol.

Velký problém bylo udělat do NetworkMultiplayeru přiřítání surovin hráčům, aby to fungovalo správně. Nastávala velmi nepříjemná chyba, o které se nakonec nezjistilo odkud se bralo a místo toho se udělal další pomocný skript. S tímto skriptem vše začalo fungovat, jak mělo.

5.9 Dokončení

Poté co ve hře fungovaly oba režimy, jak hra na jednom počítači Local Multiplayer, tak Network Multiplayer začal čas na dokončení hry. Při testování se zjistilo, že je jednoduché prohrát, když hráč udělá jedinou chybu a pak nebude moct nic udělat. To se děje proto, že jeden hráč vždy začíná poté co druhý a může se snadno stát, že se dostanou do situace, kde jeden musí do nekonečna stavět druhému (pojmy první a druhý se nemyslí, že reálně v tom pořadí začaly, ale jen situaci v tahu) a ten jen využívá toho co postaví první, aniž by mohl cokoli udělat. Nebylo tak na vybranou a tuto chybu jsme museli řešit a vyřešit. Řešením bylo udělat možnost nic vybrané kolo nedělat. Možnost Skip Turn, ale nemohla být zdarma. Naopak, jelikož cílem hry je vyhrát a ne neprohrát (čehož by se dalo, jak v každé jiné stolní hře, dosáhnout tím, že jeden z hráčů odmítne hrát a nikdy neukončí svůj tah), poplatek za vynechání kola musí být velký. Poplatek musí jasně znevýhodňovat hráče co je výhře blízko víc než toho, co je v horší situaci. Řešením bylo udělat, aby se hráči, co nechce na svém tahu nic dělat odečítaly 3 body potřebné k výhře a několik procent (40 %) všech surovin co momentálně má. Suroviny se odečítají do přiřítání na konci tahu, takže nebude je ztrácet úplně k nule, ale dřív nebo později se to ustálí i když už mu nebudou přibývat nové, dokud bude stále přeskakovat svůj tah. Tímto způsobem hráči budou dřív nebo později riskovat a pomoci protivníkovi v naději na vlastní vítězství.

Dále se musel řešit problém, že na určitých rozlišení obrazovky se menu, ve kterém se vybírá budova, kterou hráč staví, zobrazovalo špatně. To se taky muselo opravit přepsáním kódu na

zobrazování menu. Udělalo se to na úkor jednoduchosti přidávání do hry nových budov, ale to složitosti tomu už tehdy přidal network multiplayer, takže to už nebyl takový problém.

Taky se musela vytvořit možnost návratu do menu, stejně jako samotné vlastní menu, protože nechat tu věc co předepsali pouze pro debug vývojáři API rozhodně nešlo. Naštěstí to nebyl takový problém. Problém bylo to, že tlačítkům v menu se poté co se z něj jednou odešlo, po návratu ne vždy přidělovaly funkce. Přidělování funkcí tlačítkům se muselo doplnit na různá místa v kódu, ale poté už se ani jednou nestalo, že by to nefungovalo.

Nebyly to jediné chyby, co se musely opravit, ale jestli se tady bude mluvit o každé sebemenší chybě tak ta kapitola nikdy neskončí, a proto se půjde na poslední zajímavější část hry co jsme museli přidat, a tou částí je hudba, kterou jsme pro naši hru vytvořili.

6. Hudba

K tvorbě hudby bylo využito hlavně FL Studio od firmy Image Line.

FL Studio je jedním z nejpobulárnějších DAW poslední doby. DAW je zkratka pro Digital Audio Workspace, neboli digitální audio pracovní plocha. Je to obecně přístroj či software, který se používá k nahrávání, úpravě a vydávání hudby.

Jiné známé programy jsou například Ableton Live nebo Logic Pro. FL studio bylo vybráno hlavně proto, že člen týmu již tento software vlastnil, a byl již zkušený s jeho využíváním.

Většina DAW programů potřebuje ke své operaci takzvané VST pluginy (Virtual Studio Software), které se vyskytují v mnoha formách. Nejčastěji to jsou nástroje a syntetizátory, které může člověk použít, pokud zrovna nevlastní nějaký nástroj. Také se s pomocí nich dá dojít k zajímavým efektům, které by jinak byly nemožné.

K vytvoření hudby byly použity 2 VST. Jeden jde slyšet ve všech tří verzích muziky, jako syntetizátor a druhý jako bicí v posledním úryvku. Syntetizátor byl zakoupen v rámci projektu, ale bubny byly staženy z překvapivě velkého repositáře VST zdarma pluginů vst4free. Taky bylo použito MIDI (Musical Instrument Digital Interface) kláves, pro vymýšlení a nahrávání různých melodií a rytů.

Inspirací pro samotnou melodii a styl pak přišlo hlavně ze starých strategických her jako je Age of Empires, Stronghold, nebo Sid Meier's Civilization. Většina písničky je ve stupnici D-dur, kromě basové části, která obsahuje jednu jinou notu a náleží tedy k G-dur.

Samotné skládání pak nebylo moc složité, nýbrž časově náročné. Každá z melodií zabrala více než 2 hodiny pro kompozici a úpravu. Po dokončení poslední bylo zjištěno, že pokud zvuk není ve sluchátkách, kde tento problém záhadně mizí, tak je v pozadí slyšet velmi nepříjemná ozvěna, způsobená nastavením VST, která se tedy musela u každé skladby manuálně opravit.

Celková hudba se tedy skládá ze tří skladeb, které mají stále rychlejší tempo, a mění se podle stavu hry. Třetí je pak také jediná skladba, která obsahuje něco jiného než samotný syntetizátor, což jsou bubny, které by měly navozovat pocit naléhavosti a konce.

Konečná délka všech tří skladeb dohromady je asi tři a půl minuty. Jsou ale vhodné k stálému opakování, jakožto většina hudby použité ve starších hrách.

7. Local Multiplayer

Jak již bylo řečeno, Karoff podporuje dva herní módy – Local Multiplayer, a Network Multiplayer. Tato kapitola se zabývá Local Multiplayer módem. Local Multiplayer se hraje na jednom počítači, kde se dva hráči střídají na tzv. “horkém křesle”.

Tento mód je z programovacího hlediska základ celé hry, protože na něm přímo staví druhý mód, Network Multiplayer, probraný v další kapitole.

Local Multiplayer mód (dále v této kapitole jen “Hra”) původně obsahovala dvanáct scriptů – Biome, BiomeBuilding, BiomeType, Building, BuildingDisplay, BuildingManager, BuildingSpace, CameraMovement, MapBuilder, ResourceManager, TurnManager a WinManager.

Následuje krátký popis každého ze scriptů, a jak spolupracují, aby vytvořili prakticky celou Hru.

Biome

Biome je script, který zaručuje komunikaci mezi BiomeType (což je ScriptableObject), a ostatními scripty. Je připojený ke každé kostce, kterou herní deska obsahuje.

BiomeBuilding

BiomeBuilding umožňuje hráčům zvolit kostku pomocí kliknutí, čímž aktivuje tzv. “buildingSpaces”, což jsou okolní kostky, kde hráč následně může postavit novou kostku terénu. Tento script přirozeně obsahuje i funkci pro zrušení zvolení kostky, v případě, že si hráč svůj tah rozmyslí.

BiomeType

BiomeType je ScriptableObject, tudíž obsahuje jen několik proměnných, které určují jméno daného biomu, jeho ikonku, a jeho barvu. Karoff obsahuje čtyři instance tohoto typu, a to Desert, Mountains, Snow, a Forest. Tyto instance určují všechny biomy, které Hra obsahuje. Díky ScriptableObjectům by bylo opět velmi jednoduché přidat úplně nový biom, čímž by se hra podstatně rozšířila.

Building

Building je také ScriptableObject, který umožňuje tvorbu nových typů budov, jak bylo probráno výše.

BuildingDisplay

Úkolem scriptu BuildingDisplay je zobrazit všechny budovy (vytvořené přes Building ScriptableObject) do menu stavění, ze kterého si poté hráč může vybrat budovu, kterou postavit. BuildingDisplay také řeší samotné stavění budov, a to tak, že odebírá daný počet surovin hráči, a postaví budovu na vybranou kostku.

Je to jeden z nejdůležitějších scriptů Hry, protože bez možnosti stavět budovy nemají hráči prakticky žádnou možnost hrát (krom nekonečného rozšiřování mapy stavěním nových a nových kostek terénu).

Tento script je připojený k individuálním zobrazením budov v menu stavění.

BuildingManager

BuildingManager úzce navazuje na předešlý script, BuildingDisplay. Při začátku hry pro každou instanci Building (tzn. pro každý možný typ budovy ve Hře) přidá jeden objekt typu BuildingDisplay do menu stavění.

Prakticky řečeno tento script načítá všechny budovy do menu stavění, aby to nebylo nutné dělat ručně. Díky této automatizaci je možné lehce přidávat nové typy budov.

BuildingSpace

BuildingSpace umožňuje stavbu nových kostek terénu. Je to script, který je připojených k “potenciálním sousedům” každé kostky na hrací desce. Jakmile hráč vybere nějakou kostku, všichni čtyři sousedi (jeden z každé strany) se aktivují. Poté proběhnou metody scriptu BuildingSpace, a to pro každého souseda individuálně.

První metoda schová každého souseda, který je blokován již postaveným terénem. Díky tomu se nemůže stát, že by hráč postavil kostku terénu tam, kde již taková kostka existuje.

Druhá metoda scriptu BuildingSpace jednoduše poslouchá a čeká, kdy a jestli hráč klikne na tento objekt. Pokud ano, postaví na jeho pozici nový terén stejného biomu, jako je původní kliknutá kostka, odvybere tuto kostku, schová všechny ostatní sousedy, a ukončí první fázi tahu.

CameraMovement

Obrazovky hráčů mají samozřejmě omezenou velikost, a obvyklá velikost partie Karoff se na ní většinou nevejde. Proto existuje script CameraMovement, který umožňuje pomocí kurzoru myši pohybovat kamerou po scéně. Má samozřejmě omezení, aby hráči nemohli odjíždět kamerou až do nekonečna.

CameraMovement má proměnnou offset, což je prakticky velikost místa, ve kterém, pokud do něj hráč přesune myš, se kamera začne pohybovat.

MapBuilder

MapBuilder je poměrně jednoduchý script, který jednoduše volí náhodné pozice pro všechny začáteční kostky. Momentálně Hra obsahuje čtyři začáteční kostky (jednu od každého biomu), ale dá se jednoduše přidat nová začáteční kostka, script si s ní automaticky poradí; to se hodí například v případě přidání nového typu biomu.

Náhodná generace pozic začátečních kostek se zdá jako maličkost, ale reálně dodává variaci, která pomáhá partiím Karoff být zajímavé a zábavné i po několika hrách

ResourceManager

Další script, bez kterého by se Hra rozhodně neobešla; ResourceManager si drží informace o surovinách každého hráče, a o tom, kolik má každý hráč získávat surovin v druhé fázi každého tahu. Také vypisuje počty všech surovin do příslušných položek v UI tak, aby každý hráč viděl své suroviny (a suroviny druhého hráče).

TurnManager

TurnManager, jak již jméno napovídá, řeší všechno okolo jednotlivých tahů, a to jak první, tak druhou fázi. Script si drží informaci o tom, či tah zrovna probíhá, mění barvy UI elementů, aby tuto informaci zobrazovaly, a, když se mu řekne, ukončí první fázi tahu, ve druhé fázi přidá potřebný počet surovin hráčům, zkontroluje, jestli náhodou nějaký z hráčů již nevyhrál Hru, a pokud ne, začne první fázi dalšího tahu.

WinManager

V neposlední řadě tu je WinManager, který ukončuje hru, pokud jeden z hráčů dosáhl potřebného počtu bodů. Také se zde dá nastavit verze Hry, kde se hraje na konečný počet tahů, místo bodů. Tuto verzi jsme bohužel nestihli důkladně protestovat, a proto není zahrnutá v hlavním menu Karoff (nicméně by měla být plně funkční, a dá se spustit z Editoru Unity). Také ovládá změnu hudby.

Všechny tyto scripty spolu v průběhu hry spolupracují, aby vytvořily co nejlepší herní prožitek pro oba hráče. Celkem obsahují něco okolo 800 řádků kódu (bylo na dobu kdy se tato kapitola psala); to je podstatně méně, než by bylo potřeba pro tvorbu hry mimo engine Unity.

Je tomu tak díky tomu, že Unity samo řeší hodně základních funkcí, jako fyziku, vykreslování objektů na obrazovku atd.

Network Multiplayer, druhý game mode Karoff, umožňuje pro hru přes síť, přes hráče, kteří nejsou fyzicky u sebe, ale oba vlastní hru Karoff. O tomto módu pojednává další kapitola.

8. Network Multiplayer

Network Multiplayer narozdíl od dřív zmiňovaného Local Multiplayeru nemusí oba hráči hrát na jednom počítači, i když můžou. Network multiplayer probíhá prostřednictvím internetové sítě, kde jeden hráč vystupuje jako Host a druhý jako Client. Host je zároveň taky Client, ale i Server, a právě k tomu se připojuje druhý Client.

K připojení Client potřebuje znát IP adresu Host hráče a jaký Port zvolil pro hru. Pro Network Multiplayer bylo použito API od vývojářů Unity jménem UNet nebo HLAPI (The Multiplayer High Level API). Toto API bylo zvoleno z důvodu, že ostatní varianty buď nebyly úplně zdarma, nebo měli nevyhovující licence. Jako nejlepší volba nám přišlo právě toto API taky z důvodu, že je Open Source a je od vývojářů Unity enginu, takže v teorii mělo s ním dobře fungovat. Nakonec se však zjistilo, že bylo vyvíjeno pro verze ještě do Unity 2019.3. což vedlo k tomu, že práce s ním nebyla jednoduchá. Nastaly neočekávané chyby, kterým z důvodu, že API už se neudrhuje a dokumentace pro něj je z části smazaná stejně jako nejsou tutoriály co by ho rozebírali výrazně dál než po úplné základy, bylo lepší se vyhnout.

Nakonec se dokonce musel vytvořit nový projekt, ve kterém byli testovány jednotlivé základní herní mechaniky, které byly přenášeny do hry až poté co se ověřili jako funkční v tom zjednodušeném testovacím projektu. I přes všechny problémy se nám však povedlo Network Multiplayer zprovoznit a je plně funkční.

Jsou 4 způsoby, jak hrát Networking Multiplayer. Naprosto nejjednodušší možnost je aby oba hráči byli na tom samém počítači. V tom případě stačí aby hra byla spuštěna ve dvou oknech a v jedné uživatel zmáčknul tlačítko Start Hosting a ve druhé Connect To Host. IP adresu stačí nechat localhost a Port může změnit. Základní Port byl zvolen 25565 jelikož ho používali Minecraft servery a první, tehdy ještě 3D, verze hry jeho kostkovanou grafiku lehce připomínala. Tento způsob hraní, kdy je hra zapnutá dvakrát na jednom počítači je ale zbytečná a nepohodlná.

Další jednoduchou možností je hrát v jedné lokální síti. V tom případě také stačí aby jeden hráč zmáčkl Start Hosting tlačítko, druhý zadal lokální IP adresu prvního a zmáčkl Connect To Host.

Tyto 2 varianty jsou dobré pro testování hry, ale ve skutečnosti to není to k čemu Network Multiplayer má sloužit, takže další způsoby už budou zajímavější.

8.1 Port forwarding (Přesměrování portu)

Někteří lidé by mohli zkusit hned použít předchozí metodu pro hraní Network Multiplayeru, aniž by byli v jedné lokální síti. Některým by to možná i vyšlo, ale pro ostatní je tato a následující část. Tato část se bude hodit pro lidi co mají administrátorský přístup ke svému routeru. Port forwarding je potřeba, aby Client byl přesměrován a mohl se připojit k portu na soukromé adrese v lokální síti. Nejběžnější způsob, jak zapnout Port forwarding je zadat do vyhledávače IP adresu routeru (obvykle 192.168.1.1), přihlásit se jako administrátor. Dále najít záložku WAN, dále Virtual Server / Port Forwarding a tam zadat údaje které potřebuje včetně Portu který potřebujete. Tento způsob byl popsán tak jak by to probíhalo u mě, ale nemusí být pro všechny stejný, takže v případě, že nefunguje obraťte se na Google. Je to velice běžná otázka, na kterou na internetu najdete spoustu odpovědí. Ne všichni ale mají tu možnost a pro ty co nemají, něco jim nefunguje, nebo se chtějí obejít bez toho, je následující část.

8.2 Hamachi a AnyDesk

V našem případě se používal port forwarding právě do té doby, než přestal fungovat. V nastaveních se nezměnilo nic, ale port byl closed i když předtím vše fungovalo. Nebylo to poprvé co se mi to stalo a jelikož v té době byla vyzkoušená jen první hodně zjednodušená verze projektu, museli jsme hledat alternativu. Oba dále popsané programy jsou k vyzkoušení zdarma

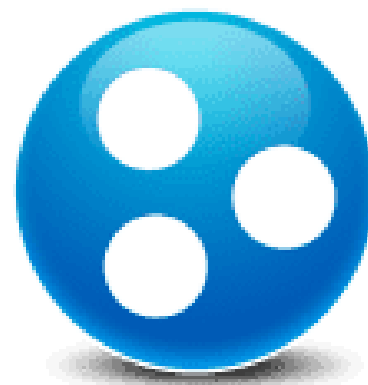
Jako první jsme vyzkoušeli AnyDesk (<https://anydesk.com>) což je remote desktop software tudíž umožňuje se připojit k cizímu počítači a sledovat, nebo dokonce ovládat jeho obrazovku a vyměňovat soubory. Zkoušeli jsme ho kvůli funkcí TCP tunneling a jeho VPN (které, ale vypadá to, že je zatím jenom na androidu a linuxu), ale nakonec i když se povedlo připojit k web serveru s Karoff to nevyšlo. Pravděpodobně jsme jen něco špatně nastavili, ale AnyDesk nám v budoucnu stejně velmi pomohl zjistit, jak to na druhé straně vypadá.



3 – AnyDesk logo

Další, co jsme zkusili, byl program Hamachi (<https://www.vpn.net/>). Hamachi se specializuje právě na tento typ problému. Hamachi vytváří společné VPN (Virtual Private Network nebo Virtuální privátní síť) a uživatelé tak budou moct komunikovat jako kdyby byli v jedné uzavřené lokální síti. Po zapnutí bude Hamachi vyžadovat registraci a poté bude stačit zmáčknout

Network, poté create a new network, poté vytvořit název a heslo. Oba uživatelé se k vytvořené network musí v Hamachi připojit a Client pak musí kliknout pravým na položku reprezentující druhého uživatele, který bude vystupovat jako Host, a v settings (v details/settings) u položky traffic nastavit Allow All. V případě, že to nepůjde v nastaveních samotného Hamachi (settings/Peer Connections) u traffic filtering nastavit na Allow All. Až Client povolí traffic u Hosta vše dál probíhá skoro stejně. Host zmáčkne Start Hosting. Client v Hamachi zas klikne na Hosta pravým a dá copy IPv4 address a vloží ji do pole pro IP adresu, nastaví stejný Port jako měl nastavený Host a dá Connect To Host.



4 – Hamachi logo

Tou poslední variantou jsme testovali, a tak jsme hráli Karoff až do odevzdání. V případě nějakých problému s Hamachi se můžete také obrátit na Google, jelikož se Hamachi hodně používalo pro nejrůznější účely už poměrně dlouhou dobu je na internetu spousta návodu jak na to.

Poznámka pro uživatele firewallu. Firewall může hraní hry přes NetworkMulitplayer překážet. Víme jistě, že v případě hraní přes Hamachi, host musí povolit Karoff v nastaveních firewallu (allow app through firewall) a povolit VPN pro hamachi (allow private network).

Allow apps to communicate through Windows Defender Firewall

To add, change or remove allowed apps and ports, click Change settings.

What are the risks of allowing an app to communicate?



Allowed apps and features:

Name	Private	Public
<input checked="" type="checkbox"/> karoff.exe	<input checked="" type="checkbox"/>	<input type="checkbox"/>

5 – Firewall settings

Na závěr k Network Multiplayeru chci říct, že odpojit se od Hosta může Client tím, že na klávesnici stisknete ESC a pak se může opětovně připojit stejně jako to udělal předtím a hra se mu načte do stavu v jakém momentálně je (všechny změny co provedli, dokud byl ve hře zůstanou a všechny co provedl Host, zatímco nebyl, pokud Host byl zrovna na řadě se taky načtou). Je to uděláno proto, že jeden z uživatelů může mít špatné připojení k internetu, a proto to Clienta může odpojit, Client se může odejít sám, restartovat hru, dát si pauzu a nic se nestane. Ale jestli hru vypne Host tak se hra neuloží. Host je zároveň Client i Server, a tudíž mu nehrozí, že se Serverem ztratí spojení. Taky se může stát, že Client se připojí a vidí jen šedou obrazovku. To je způsobeno tím, že počet hráčů je omezený a Client nějakým způsobem zmáčknul tlačítko připojení víckrát naráz, než se stihlo deaktivovat. Nakonec se ukázalo, že možnost Clienta se znovu připojit je velmi praktická už při testování, jelikož snad nikdo z nás tu nemá zrovna stabilní připojení a s tím problémem by se uživatelé mohli setkat také.

9. Finální Program

V této kapitole se shrne, co vše jsme vytvořili, kde se to dá najít a lehce přiblíží co to dělá, protože kdyby se popisovala celá hra krok po kroku hra by byla mnohem delší a myslím, že kód to říká lépe než slova.

Práci našeho týmu najdete na GitHubu na odkaze <https://github.com/gyarab/Karoff>. Hned v root adresáři Karoff uvidíte z toho, co nedogeneroval git ani operační systém 3 důležité věci. V zipu se nachází 2 verze hry. V zipu jménem Mac Karoff se nachází buildlá verze pro mac ve formátu APP a v Windows Karoff složku kde je vše potřebné k spuštění buildlé pro windows verze souborem ve formátu EXE. Dále je tam složka Karoff ve které je samotný Karoff project.

Ve složce Karoff se nachází spousta souboru. Soubory .sln a .csproj byli automaticky vytvořené Visual Studiem a zbytek Unity. Ve složkách vytvořených Unity je hodně důležitých souboru, například package instalované v Unity package manageru jako součást Unity jako TextMeshPro a UNet. Samotná naše práce se nachází ve složce Assets.

Ve složce assets je taky hodně souboru například soubory meta, které vytvořilo Unity pro každý asset a podle dokumentace Unity v těch souborech ukládá ID, které dalo jednotlivým assetům. V assets složce je ještě složka dříve zmiňovaného TextMeshPro kterou jsme nevytvořili my a v Music je složka Audio s veřejně dostupnými všem zvuky kliknutí, ale zbytek je čistě práce našeho týmu.

Ve složce Data se nachází složky Biomes a Buildings. Obsah těchto složek jsou právě data jednotlivých scriptable objectu o kterých již řeč byla, ale pro případ, že nevíte připomenou, že ScriptableObject je typ classy, která na rozdíl od klasického Unity MonoBehaviouru není komponentem žádného GameObjectu. Namísto toho vývojář může vytvářet instance této classy ve složkách hry, kde tyto instance ukládají různá data. Což tady právě pozorujeme.

- Ve složce Images najdete obrázky použité ve hře. Jsou rozdělené podle toho, co zobrazují.
- Dále následuje složka MultiplayerScripts obsahující obrovské množství skriptu.

BuildingSpacesColor nastavuje barvu vašich preview pro stavení na červenou, když je na řadě druhý hráč a na bílou, když jste na řadě vy.

MenuNetworkManager se stará o menu. Nastavuje funkce tlačítkům, přijímá obsah textových polí a zapíná hry. Taky poslouchá, jestli hráč nezmáchl tlačítko ESC aby odešel zpět do menu nebo ze hry.

MultiplayerBiome dělá fakticky to samé co obyčejný Biome popsáný v kapitole LocalMultiplayer, jen v NetworkMultiplayeru.

SyncTile ukládá jako stringy a bool o tom jaký biome, starting a sprite má GameObject na kterém se nachází a podle těch nastavuje sprite a biome toho GameObjectu. Může se to zdát zbytečné, host a client mají každý vlastní verzi GameObjectu a jejich stav se musí synchronizovat. Právě o to se stará tento skript.

MultiplayerBiomeBuilding, MultiplayerBuildingDisplay, MultiplayerBuildingManager, MultiplayerBuildingSpace, MultiplayerMapBuilder, MultiplayerResourceManager a MultiplayerTurnManager dělají fakticky to samé co dělaly jejich již popsáné v kapitole o LocalMultiplayeru verze, jen to dělají poté co určí, jestli je hráč, co je zavolal Host nebo Client a jestli jsou zrovna na řadě. V případě, že jsou zavolají jednu z funkcí PlayerCommands skriptu, který vykoná zbytek. Někdy jejich proměnné jsou nastavené skriptem zvaným ze skriptu PlayerCommands, nebo mají proměnnou synchronizovanou mezi oběma hráči, jako například MultiplayerTurnManager má synchronizovaný číslo momentálního tahu, tak že se nestane, že se bude u dvou hráčů lišit. Každopádně jejich funkcionál se výrazně neliší, protože plní stejnou funkci jen jiným způsobem.

MultiplayerWinManager dělá taky fakticky to samé co obyčejný WinManager, jen taky přizpůsobuje svou aktivitu network multiplayeru jako například to, že ukazuje oboum hráčům, že vyhrál hru ten co dosáhl potřebného množství bodů jako první.

MultiplayerCameraMovement ovládá pohyb kamery stejně jako to udělal skript CameraMovement, ale v tomto skriptu se pohyb kamery dal pohyb kamery vypnout dříve než u CameraMovement, takže je příkladem toho, že některé novinky přichází do LocalMultiplayeru z NetworkMultiplayeru a ne vždy naopak.

PlayerID je jeden z důležitých pro NetworkMultiplayer skriptu. Původem je z prvního testovacího projektu, ale od té doby byl výrazně zjednodušen. Teď sice furt přiřazuje sobě id podle toho, jaký ID má jeho komponent NetworkIdentity (z API), ale již neztrácí čas tím, že by volal nějaké další

funkce a synchronizoval své id s druhým hráčem. Druhý hráč nemá potřebu vědět kdo je druhý hráč, když je hra stavěná na 2 hráče stačí když hráč vždy bude vědět, jestli je host nebo client. Takové řešení taky pomáhá v případech, kdy třeba kvůli pomalému internetu (u testování mezi členy týmu se to stávalo docela běžně), client ztratí spojení se serverem a musí se připojit znovu. V tomto případě, když se připojí může pokračovat kde skončil hned. Jiné skripty taky rozeznávají hráče podle toho, jestli je host nebo client. Aby neztrácel čas, ale dál to vědět tento skript nastaví jméno GameObjectu reprezentujícího hráče na host nebo klient v souvislosti s tím kdo je. Je výhodný mít tento skript co nejrychlejší, protože ostatní musí odkládat svůj start, aby už stihl tímto dát najevo kdo je.

SyncResources je skript co synchronizuje obrovské množství surovin obou hráčů a jejich multipliu. Taky MultiplayerResourceManager ze kterého je dostává většina jiných funkcí je v NetworkMultiplayeru dostává odsud. Taky každou sekundu je nastavuje ResourceManageru znovu pro případ, že by nastala chyba, nebo se něco změnilo. Ještě odpovídá za odebrání surovin a pointu při přeskočení tahu.

A na konec z této složky to nejzajímavější a tím je PlayerCommands script.

Je velice zajímavý a důležitý, dokonce bych řekl, že nejdůležitější ze všech v této složce. Tento skript je taky jako PlayerID skriptem přímo na GameObjectu reprezentujícím daného hráče, který se vytvoří, když se hráč připojí. To důležité ale na tom je to, že obsahuje řadu public funkcí volaných z nejrůznějších skriptu v této složce. Z nich se pak volají [Command] (cmd), které posílají příkaz serveru (což je v tomto případě host), je to nutný zaprvé, protože ne na vše hráči mají oprávnění a také některé věci bez ohledu na oprávnění může dělat jenom server. Taková věc je například vytvoření nového GameObjectu podle prefabu (k prefabům se dostanete až když budeme ve složce prefabu). Jestli se o to pokusí hráč tak to vytvoří existují jen pro něj variantu toho GameObjectu. Na tom principu funguje preview u stavení, nebo otevřenost menu, které vidí jen ten hráč co kliknul na už postavenou kostičku, otevřel BuildingMenu. Server nastaví authority (fakticky nastavení přístupu k GameObjectu) a zavolá RPC (Remote Procedure Call) funkci, která probíhá u všech hráčů. Touto funkcí může server nastavovat hráčům proměnné včetně SyncVar na synchronizaci a volat funkci co proběhne u obou hráčů, například změna tahu.

Je čas se posunout na další složku a tou je složka Music. V té jsou .wav soubory s vytvořenou námi hudbou a složka Audio s různými veřejnými zvuky kliknutí (které jsme nedělali my, ale jsou všem veřejně dostupné).

Další složka je složka Prefabs, která obsahuje prefaby. Prefab je objekt, od kterého se odvíjí tzv. klony – jakákoliv změna v master objektu změní danou věc i ve všech klonech. Klony se ale dají individuálně upravovat, bez efektu na ostatní klony. Hodně zjednodušeně se dá říct, že uložené prefaby jsou šablony pro klony, co se podle nich vytváří.

Následuje složka Scenes, která reprezentuje jednotlivé scény hry jako je menu, MultiplayerScene a LocalScene. Když hráč zapíná, nebo vypíná hru probíhá výměna načtených scén.

Následující složka obsahuje Skripty z větší části již popsané v LocalMultiplayer kapitole, takže tady se budu věnovat jen těm novým přidaným v době, kterou popisuje podkapitola "Dokončení" kapitoly "O vývoji".

Hned na začátku je nová složka AudioManager, kde jsou skripty ovládající hudbu, kromě přepnutí jedné hudby na jinou. Přepnutí jedné hudby na jinou provádí WinManager.

Další z nových skriptů je CameraShake, který lehce zatřese kamerou, když hráč postaví nový dílek mapy. Je to udělané pro estetický pocit vlivu na herní svět stejně jako particles nebo zvuk při kliknutí. Obsah toho kódu využívá nejen LocalMultiplayer, ale i NetworkMultiplayer.

SkipTurn je posledním novým velkým skriptem. Dokáže podle názvu momentálně aktivní scény poznat, jestli je LocalMultiplayer hra, nebo NetworkMultiplayer a pokud local tak provede sám celý skipTurn, a pokud není zkontroluje, jestli hráč, co ho zavolal je zrovna ten, co je na řadě a pokud ano zavolá funkci z PlayerCommands co zaručí dokončení přeskočení tahu a odečtení bodů a surovin za to.

Toto bylo to, co jsme vytvořili a pohromadě to tvoří naši hru. Teď víte, kde hledat jednotlivé části a máte představu o tom, jak to funguje.

10. Ovládání

Představa o tom, jak to funguje, vám ale neřekne, jak hru ovládat.

Ovládání je jednoduché.

Mezerníkem (SPACE) se zapne a zase vypne posouvání kamery. Kamera se posouvá, pokud je kurzor myši blízko okraje obrazovky, když je posouvání zapnuté.

Tlačítko ENTER vám přeskočí váš aktuální tah, pokud je opravdu váš, ale bude vás to stát 40% každé suroviny a 3 body. Do mínusu nejedete, toho se bát nemusíte, ale stejně musíte pečlivě spočítat, jestli se to teď vyplatí.

Tlačítko ESC umožní ukončit hosting a vrátit se do menu, pokud jste byli host (veškerý progress hry bude navždy ztracen). Odpojit se a vrátit se do menu, pokud jste byli Client. V tomto případě se můžete zase do hry připojit, pokud je furt host neukončil. Nebo se prostě vrátíte do menu, pokud to byla LocalMultiplayer hra.

Jestli ESC zmáčknete v menu, hra se ukončí.

Ve hře, když jste na řadě můžete kliknout na již postavenou kostičku a zobrazí se vám kam ji můžete postavit. Když kliknete na jedno z preview co se objevily postaví se tam nová a váš tah se ukončí.

Když jste ve hře na řadě a 2krát kliknete na postavenou kostičku bez ikonky otevře se building menu, které můžete buď scrollovat kolečkem myši, nebo posouvat tím, že do něj kliknete, držíte a táhnete myší.

Když kliknete na něco z building menu, dá se to postavit na vybranou kostičku a máte dostatek surovin tak se postaví a zvýší se vám multiplier dané suroviny. Nakolik se zvýší multiplier se píše vlevo dole a cena vpravo z boku.

11. Budoucí plány

I když je Karoff plně hratelná hra, je spousta polí, kde by se mohla zlepšit. Samozřejmě plánujeme jako tým práci na projektu dále pokračovat, ale pravděpodobně přibereme nějaké další členy.

Například grafika by se mohla o dost zlepšit, kdyby každá kostka terénu byla detailní ilustrace, každá budova komplexní 3D model, který by se na kostku terénu postavil, a každá akce by vyvolala nějakou menší vizuální reakci (hráči pak mají pocit, že mají na svět hry větší vliv, a více je hra poté baví).

Hra by se určitě dále dala přizpůsobit na mobilní zařízení, aby hráči mohli hrát například v metru, v autě, nebo ve škole (jen o přestávkách). Takový port by vyžadoval hodně práce, i přes to, že Unity porty na Android a iOS zařízení podporuje.

Muselo by se například předesignovat UI, aby zobrazovalo veškeré potřebné informace v malém rozlišení mobilních obrazovek. Ovládání, které momentálně běží téměř výhradně myší, by se také muselo celé přepsat.

Dále je možné jednoduše přidat libovolný počet nových biomů a budov (díky tomu, že je systém hry k tomuto optimalizovaný). Tímto způsobem by se dala relativně exponenciálně rozšířit komplexita hry, s malým množstvím investované práce. Jediný problém by tu bylo balancování hry, a proto by se muselo provést hodně alpha a beta testování. Taky by bylo dobré přidat a otestovat nové režimy.

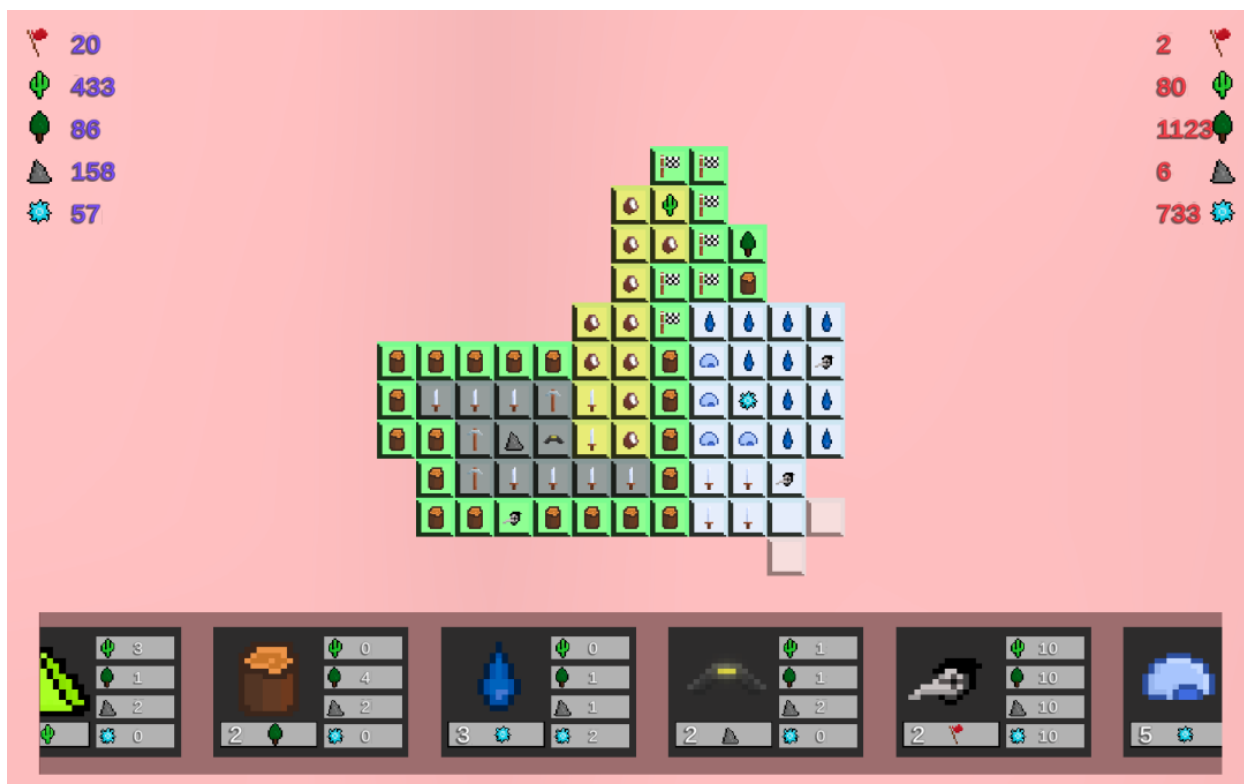
Také by se do hry daly přidat nastavení, aby si hráč mohl přizpůsobovat hru víc sobě a tutoriál aby hru mohl rychle pochopit i ten kdo nemá dokumentaci.

Unity posledním rokem (2019/2020) mluví o novém network multiplayer systému, který má fungovat mnohem lépe, než zastaralé multiplayer API (u net/hlapi), které Karoff momentálně používá. Nový systém pravděpodobně vyjde někdy během roku 2021, a až se tak stane, rozhodně bychom chtěli Karoff na tento systém přesunout.

12. Závěr

Na závěr chci vysvětlit možnou nejasnost, že v git pushe od uživatele tfshaman jsou pushe od člena týmu jménem Timofey Kiselev a od uživatele Darezar od člena týmu Tadeáš Jůn. Je to tím, že již jsme účet měli a neustále se přepínat mezi nimi je zbytečné.

Zadání bylo splněno a u hry se musí dokonce docela dost přemýšlet. Jako příklad uvedu screenshot od konce jedné z her mezi členy týmu.



6 – Screenshot hry

V této hře vyhrál v následujícím tahu modrý hráč. Nejdříve to šlo vyrovnaně, ale červený hráč v jednu chvíli udělal chybu a místo zachování monopolu na vodu si vzal poušť. Tato chyba ho nakonec stála výhru, protože druhý hráč chytrým přeskakováním tahů. Poté dokázal uzavřít monopol na kámen, protože červený nemohl ani postavit kostku na, kterou by se daly postavit doly, aniž by ji modrý nezabral, ani si dovolit mu získat víc dřeva a vody na které vedl. A tak když červený neměl žádný rychlý zdroj kamene a pochopil, že může získat jen 2 body jednou za 5 tahu, kdežto ztratit musí alespoň 12 za stejnou dobu, pochopil, že už nemá šanci na výhru a dovolil modrému vyhrát.

Jak vidíte u hry se musí poměrně dost přemýšlet, protože jedna chyba blízko začátku hry může prohrát celou hru a musíte včas pochopit, jestli to, co zrovna děláte nepomáhá protivníkovi víc než vám. Taký musíte správně odhadnout, jestli teď se vyplatí ztratit část surovin a pointu a počkat, nebo vás to bude stát víc než protivníka a jen vás příliš to neoddálí od výhry.

Během práce bylo hodně problémů, které se postupně museli řešit, ale hlavně, že z toho každý člen týmu odnesl nějaké ponaučení pro budoucí týmové, a nejen projekty a vznikly už určité šablony pro spolupráci a komunikaci se zbytkem týmů. K ponaučení na budoucnost je ještě jedna důležitá poznámka, že co se může pokazit se nejspíš pokazí, a tak ani dokončování dokumentace se nevyplácí odkládat na posledních pár dní. Výsledkem této spolupráce vznikla hra, kterou jsme si mezi sebou v týmů rádi zahráli a doufáme, že se bude líbit i ostatním.

13. Nástroje

Unity a package unet/textmeshpro https://unity.com/
Unity hub https://docs.unity3d.com/Manual/GettingStartedInstallingHub.html
Visual Studio https://visualstudio.microsoft.com/
FL Studio https://www.image-line.com/flstudio/
Figma na manual, https://www.figma.com/
Aseprite na ikonky https://www.aseprite.org/
GithubDesktop https://desktop.github.com/
Google docs https://www.google.com/docs/about/
Ms Word https://www.microsoft.com/en-us/microsoft-365/word
Git https://git-scm.com/
Github https://github.com/

14. Seznam Obrázků

1 – První prototyp	9
2 - Ikonky	10
3 – AnyDesk logo.....	22
4 – Hamachi logo	23
5 – Firewall settings	23
6 – Screenshot hry	31

15. Zdroje

- A Shell in the Pit. (2013, May 8). Retrieved from YouTube:
https://www.youtube.com/watch?v=eY4_tcoCkYg
- AnyDesk. (n.d.). Retrieved from anydesk.com:
https://anydesk.com/_static/img/favicon/anydesk_icon.png
- Brackeys. (n.d.). Retrieved from YouTube: <https://www.youtube.com/watch?v=6OT43pvUyfY>
- Brackeys. (2017, May 31). Retrieved from YouTube: <https://www.youtube.com/watch?v=6OT43pvUyfY>
- Brackeys. (2018, February 25). Retrieved from YouTube:
<https://www.youtube.com/watch?v=9A9yj8KnM8c>
- Donion Tech. (2017, June 26). Retrieved from YouTube:
<https://www.youtube.com/watch?v=x7DIllpgGs4s>
- Donion Tech. (2017, June 26). Retrieved from YouTube:
<https://www.youtube.com/watch?v=9VW7ctwvNok>
- Euvino, R. L. (2001, October 19). Retrieved from YouTube:
<https://www.youtube.com/watch?v=JqmxljAOgJg&>
- GTGD. (2015, July 10). Retrieved from YouTube: <https://www.youtube.com/watch?v=9w2kwGDZ6wM>
- Kenney. (n.d.). Retrieved from kenney.nl: <https://www.kenney.nl/assets/ui-audio>
- Lana, J. (n.d.). Retrieved from Testovadlo: <https://gyarab.ddns.net/>
- lifeforstock. (2019). Retrieved from freepik.com: https://www.freepik.com/free-photo/abstract-white-color-canvas-wallpaper-textures-surface_4011913.htm#page=1&query=canvas%20texture&position=3
- LogMeIn. (n.d.). Retrieved from logmein.com: <https://banner2.cleanpng.com/20190228/phy/kisspng-logmein-hamachi-virtual-private-network-logmein-i-5c77ff209a2fd0.7202092115513679686316.jpg>
- Max Project. (2019). Retrieved from <https://maxproject.wixsite.com/>:
<https://maxproject.wixsite.com/maxproject/vsti/>
- Reaper1121. (2015, June 21). *answers.unity.com*. Retrieved from unity.com:
<https://answers.unity.com/questions/990807/how-do-i-get-sprite-renderer-from-all-child.html>
- Rutledge, A. (n.d.). Retrieved from tweakbench.com: <https://www.tweakbench.com/vst-plugins/triforce>
- sanks007. (2013, September 11). *answers.unity.com*. Retrieved from unity.com:
<https://answers.unity.com/questions/534582/accessing-variable-defined-in-a-script-attached-to.html>

Sisso. (2013, March 06). <https://answers.unity.com/>. Retrieved from unity.com:
<https://answers.unity.com/questions/411793/selecting-a-game-object-with-a-mouse-click-on-it.html>

Unity Technologies. (2020, 04 29). docs.unity3d.com. Retrieved from
<https://docs.unity3d.com/Manual/Profiler.html>

Unity Technologies. (2020, April 29). docs.unity3d.com. Retrieved from unity.com:
<https://docs.unity3d.com/Manual/UNet.html>

Unity Technologies. (2020, April 29). <https://docs.unity3d.com/>. Retrieved from unity.com:
<https://docs.unity3d.com/Manual/index.html>