

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

VÝVOJ STRATEGICKÝCH HIER V UNITY
BACHELOR THESIS

2022
PETER HORVÁTH

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

VÝVOJ STRATEGICKÝCH HIER V UNITY
BACHELOR THESIS

Študijný program: Informatika
Študijný odbor: Informatika
Školiace pracovisko: Katedra Aplikovanej Informatiky
Školiteľ: RNDr. Andrej Lúčný, PhD.
Konzultant: Mgr. Silverster Buček

Bratislava, 2022
Peter Horváth



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Peter Horváth
Študijný program: informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Tvorba strategických hier v Unity
Development of strategy games in Unity

Anotácia: Táto analyticko-implimentačná práca analyzuje aké konkrétne softwarové prostriedky potrebujeme pre implementáciu strategickej hry. Najprv všeobecne popíše aké konkrétne herné objekty a scenáre potrebujeme v prípade strategickej hry riešiť. Pritom sa môže opierať o analýzu existujúcich strategických hier, napríklad Rais. Potom popíše akými technickými prostriedkami v jednom z najpoužívanejších prostredí na tvorbu hier – Unity ich je vhodné implementovať (štandardné herné objekty a assety, všeobecne dostupné assety z asset store a vlastné prefabrikáty). Následne tieto použije pre implementáciu pilotnej strategickej hry.

Cieľ: Cieľom práce je analyzovať prostriedky tvorby strategických hier v Unity a vyskúšať ich na konkrétnej hre tohto typu.

Literatúra: Tomáš Holan: Unity, CZ NIC, 2021
Developerská web stránka Unity www.unity.com
Tiago de Souza Lima Gomes: "Computer Game Studies: Research and Educational Designs", dostupné na: <http://www.bocc.ubi.pt/pag/gomes-tiago-teixeira-marcelo-computer-games-studies.pdf>

Poznámka: Platforma: Unity. Práca vzniká v spolupráci s doktorandom na VŠMU, ktorý do nej dodáva herný scenár a grafické prevedenie pilotnej hry.

Kľúčové slová: počítačová hra, Unity, assety, strategy game

Vedúci: RNDr. Andrej Lúčny, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.

Dátum zadania: 06.10.2021

Dátum schválenia: 11.10.2021

doc. RNDr. Daniel Olejár, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Pod'akovanie: Týmto by som sa chcel poďakovať školiteľovi, ktorý bol nápomocný s každým aspektom a vďaka jeho pohotovým reakciám a radám žiadny problém nespôsobil dlhé zdržanie. Zároveň by som sa chcel poďakovať konzultantovi pri tejto práci, ktorého pomoc bola nesmierne prínosná pri implementačnej časti.

Abstrakt

Táto práca rozoberá povahu a vlastnosti strategických hier a poskytuje náhľad na prístup k dizajnu a vývoju strategickej hry. Jedna z myšlienok prezentovaných v tejto práci je, že videohry sú funkčný softvér, ktorý navyše obsahuje špeciálne aspekty, ako umenie alebo gameplay, a preto ku vývoju videohier by sa malo pristupovať ako ku vývoju softvéru, narozdiel od prístupov ku tvorbe videohier ako ku tvorbe umenia.

Ďalej táto práca popisuje niektoré všeobecné princípy vývoja videohier, zaoberá sa prvkami dizajnu a tomu, ako dizajn môže viesť ku konkrétnej implementácii niektorých častí hry, a konkrétne nástroje na implementáciu. Konkrétne sa pozeráme na vývoj hier v Unity, takže prostriedky sú viazané na vývoj projektu v prostredí Unity.

V neposlednom rade, sme vytvorili návrh dizajnu strategickej videohry a následne sme pomocou Unity implementovali prototyp založený na tomto dizajne.

Kľúčové slová:

počítačová hra, Unity, assety, strategická hra.

Abstract

This thesis discusses the nature of strategy videogames and describes certain aspects of the design and development processes of such games. One of the key opinions that this thesis represents, is that videogames are a functional software, with specific properties, like art and gameplay, and as such game development should be treated as software development, rather than artwork creation.

This thesis then proceeds to describe stages of general videogame development, and discusses nuances of the transition from a video game design to implementation. Next, it also describes chosen software tools commonly used to development of strategy games, focused around Unity as the main tool for this development.

And lastly, we have created a design of a potential strategy game, and subsequently design a prototype in Unity, that is based around this design.

Keywords:

computer game, Unity, assets, strategy game

Obsah

Úvod	1
1 Strategické hry ako žáner a ich vlastnosti	3
1.1 Delenie Strategických hier podľa štýlu interakcie	6
1.1.1 Real-time strategy	6
1.1.2 Turn-based strategy	6
1.2 Stratégia a Taktika	7
1.3 Delenie Strategických hier podľa kľúčových vlastností	9
1.3.1 Auto-battlers	9
1.3.2 Tower defense	10
1.3.3 Real-World-Time	10
1.3.4 4X	12
1.3.5 Grand strategy	13
1.4 Známe strategické hry	14
1.4.1 StarCraft	14
1.4.2 Into the Breach	17
1.4.3 XCOM	19
1.4.4 Factorio	20
1.4.5 Dune	23
1.4.6 Heartstone	24
1.4.7 Frostpunk	26
1.4.8 Total War	26
2 Princípy tvorby strategických hier	29
2.1 Princípy vývoja softvéru používané pri tvorbe strategických hier	29
2.1.1 Životný cyklus vývoja softvéru	29
2.1.2 Životný cyklus vývoja hier	31
2.2 Prvky dizajnu strategických hier	40
3 Prostriedky vývoja strategických hier	43
3.1 Unity	43

3.1.1	Dokumentácia	43
3.1.2	Editor	43
3.2	CSharp	46
3.3	VisualStudio	46
3.4	Git	46
4	Implementácia pilotnej hry a vyhodnotenie práce	49
4.1	Vývojové štádia a dizajn hry	49
4.2	Popis prototypu	53
4.3	Porovnanie s podobnými prácami	55
	Záver	57
	Príloha A	61

Zoznam obrázkov

1.1	Koláž rôznych rts hier	6
1.2	Batalj, TBS videohra	7
1.3	Dota auto-chess	9
1.4	Kingdom Rush	11
1.5	Divoké kmeny	12
1.6	Age of Wonders: Planetfall	13
1.7	Field of Glory: Empires	14
1.8	Titulný obrázok StarCraft 2	15
1.9	Obrázok herných prvkov SC2 z pohľadu hráča	16
1.10	Titulný obrázok Into the Breach	18
1.11	Herné prvky Into the Breach z pohľadu hráča	19
1.12	Ukážka z tutoriálu XCOM 2	20
1.13	Titulný obrázok Factorio	21
1.14	Snapshot továrne z Factorio	22
1.15	Náhľad na herné prvky strategickej vrstvy Dune	23
1.16	Herná plocha v zápase Hearthstone	25
1.17	Snapshot mesta Frostpunk v skorej fáze hry	26
1.18	Náhľad strategickej mapy v Total War saga:Troy	27
1.19	Náhľad okupovania nepriateľskej pevnosti v Total War RTS vrstve	28
3.1	Unity editor	44
4.1	Náhľad na hernú plochu	54
4.2	Detaily zvoleného regiónu	54

Úvod

V tejto kapitole predstavím tému a ciele našej práce, členenie práce a motivácie za výberom témy.

V tejto práci sme mali stanovené dva hlavné ciele, prvým cieľom bola teoretická časť, ktorá sa pozerá na žáner strategických hier, analyzuje ich povahu, čím sa strategické hry líšia od ostatných žánrov videohier. Za týmto účelom sa pozrieme na niekoľko jednotlivých, alebo skupín, strategických hier a pozrieme sa bližšie na ich vlastnosti.

Ďalej sa venujeme princípov tvorby strategických hier, kde sa pozeráme na zaužívané stratégie prístupu k vývoju strategických hier, ale aj na ich východiskové predpoklady, ktoré sa viažu na vývoj videohier všeobecne, a to sa viaže na vývoj všeobecného softvéru. Týmto si zároveň priblížime veľmi dôležitý koncept, “vývoj videohier, je vývoj softvéru”. Okrem toho sa však pozeráme aj bližšie na konkrétne metodológie používané pri vývoji hier, povahu vývoja hier, niektoré významné pozície vývojarského tímu pri vývoji videohier a niektoré technickejšie špecifikácie strategických videohier, ktoré sa, narozdiel od všeobecných vlastností strategických hier, užšie viažu na implementáciu konkrétneho softvéru.

Pri téme implementácie ostaneme a pozrieme sa na konkrétne softvérové prostriedky, spájané s vývojom strategických videohier a s vývojom videohier v Unity, na základné informácie o nich, o tom ako fungujú, respektíve na čo sa používajú.

Následne, prejdeme na druhý cieľ našej práce, ktorým je návrh dizajnu strategickkej hry, v súlade s prejdenou teóriou, a následná implementácia prototypu založeného na tomto dizajne.

Na konci tejto práce popisujeme dizajn tejto hry a taktiež štádia procesu implementácie prototypu, ktorý slúži ako pilotná aplikácia pre túto prácu.

Kapitola 1

Strategické hry ako žáner a ich vlastnosti

Z historického hľadiska, strategické hry vznikli za účelom simulovania vojnových bitiek, a boli primárne využívané pre účely tréovania stratégov a generálov. Medzi prvé takéto hry patrí kriegsspiel (doslovný význam - vojnová hra), vyvinutý pruskou armádou v 19. storočí za účelom naučiť dôstojníkov vojnové taktiky a stratégie. V 19. storočí je, samozrejme, ťažko hovoriť o videohre, avšak v dnešnej dobe kriegsspiel existuje voľno prístupný vo virtálnej podobe, s upravenými pravidlami.

Na adrese [1]www.general-staff.com/tag/kriegsspiel, napríklad nájdeme fórum komunity, ktorá sa v nedávnej minulosti, rok 2016, zaoberala adaptáciou kriegsspielu do existujúceho herného frameworku.

Na kriegsspiel-online.netlify.app zase nájdeme server pre online hranie, adaptácie kriegsspielu.

Čo je viac, na rovnakom koncepte ako kriegsspiel, simulujúcej skutočné vojnové konflikty, sa následne vyvíjalo mnoho ďalších strategických hier s vojnovou tematikou. My by sme však chceli presnejšiu definíciu strategických hier, ideálne až na úroveň samotných herných mechaník, ktoré je očakávané, že budú implementované v hre, ktorá sa prezentuje ako strategická hra. Kategorizácia hier však nie je presná veda, existuje mnoho rôznych kategorizácií videohier, ktoré môžu byť často krátko navzájom konfliktné. Týmto fenoménom sa vo svojej práci zaoberal aj Simon Dor [7], ktorý sa snažil popísať rôzne dôvody pre odlišné kategorizácie a čo tieto kategorizácie mohlo ovplyvniť.

Hlavné dôvody pre kategorizáciu hier do žánrov vieme rozlíšiť dva: V prvom rade ide o cieľný marketing, a vytvorenie štruktúry na ktorej sa dajú implementovať zmysluplné odporúčacie systémy a personalizované newslettery. Na základe

toho aké hry sa hráčovi páčili v minulosti, vieme aké hry sa mu pravdepodobne budú páčiť ďalej, a teda má zmysel ponúkať mu reklamu na tieto hry. Toto vieme rozhodovať buď na základe podobnosti medzi hrami (podobnosť kategorizácie rôznych hier), alebo na základe podobností užívateľov (inými slovami, ak máme hráča ktorému sa páči niekoľko hier z kategórie A a máme štatistiku, že 90% hráčov, ktorým sa páči niečo z kategórie A, sa taktiež páčia hry z kategórie B, tak má zmysel nášmu cieľovému hráčovi odporúčať hry z kategórie B).

Druhý účel je pre herných vývojárov a dizajnérov, ak herný tím dostane úlohu vytvoriť videohru z konkrétneho žánra, je pre nich veľmi dôležité vedieť aké herné mechaniky (herné mechaniky sú metódy v rámci videohry, navrhnuté na interakciu medzi hráčom a herným svetom [24]) by tá hra v takomto žánri mala obsahovať, aké by mali byť spôsoby interakcie medzi hráčom a hrou.

Stále, nám by sa zišlo mať nejakú definíciu strategických videohier s ktorou môžeme pracovať, skúsme sa pozrieť na nejaké všeobecné charakteristiky, na ktorých sa väčšina definícií zhoduje. [26]

Strategické hry, sú hry, v ktorých je hráčova schopnosť vytvárať rozhodnutia a plánovať dopredu, na základe dostupných informácií, kľúčová k výsledku hry. Postupné vykonávanie menších akcií, ktoré napomáhajú splneniu väčšieho plánu sú často oveľa významnejšie ako okamžitý výsledok vykonanej akcie. Tieto vlastnosti môžu byť stelesnené ako ovládanie malých postavičiek na veľkom bojisku alebo ako manipulácia obmedzených prostriedkov, hráč môže mať neobmedzený čas analyzovať situáciu pred každým rozhodnutím, alebo je čas obmedzený a hráč musí vytvárať rozhodnutia pod tlakom, často za meniacich sa okolností (napríklad následkom akcie iného hráča, alebo istého RNG elementu). Je pochopiteľné, že robiť rozhodnutia je jednoduchšie pokiaľ má hráč dostupné väčšie množstvo informácií, preto je zbieranie informácií dôležitou súčasťou gameplay loopu väčšiny hier. Zbieranie informácií však môže presahovať rozhranie samotnej hry, vo veľa hrách, je pre hráča veľkou výhodou, napríklad, rozumieť spôsobu akým hra funguje na pozadí, čo nie je vždy explicitne povedané hráčovi, no môže to viesť k zvoleniu efektívnejšej stratégie.

K tomuto nadväzuje ďalšia vlastnosť, ktorou je to, že hráč manipuluje nejakou množinou objektov alebo prostriedkov. Tieto môžu byť suroviny a peniaze, pre nejaký resource management, alebo je to vojenská základňa, s tréningovými budovami a jednotkami, alebo je to virtuálne impérium s oboma zložkami naraz. Toto je taktiež hlavný spôsob ako hráč interaguje so svetom konkrétnej videohry, čo je veľkým rozdielom, oproti napríklad role-playing hrám alebo akčných strielačkám, hráč nie je v pozícií konkrétnej osoby, ktorá niečo robí, namiesto toho má hráč priamu kontrolu nad väčším objektom (krajiny, armády, spoločnosti), a hráč len zadáva príkazy, ktoré

tieto objekty, alebo postavičky v nich, vykonávajú.

Z naratívneho hľadiska to môže dať zmysel, že hráč je prezentovaný ako kontrolujúca osoba (generál armády, prezident krajiny), ale v reálnom svete existuje v takejto situácii istý reťazec príkazov, ak generál chce, aby jeho armáda vykonala istý príkaz, musí ten príkaz vydať, sformulovať ho a dopraviť ho dotyčným jednotkám. Navyše samotné jednotky sú vlastné, mysliace bytosti, ktoré vedia konať z “vlastnej vôle”.

Tieto udalosti sú zo štandardnej strategickej hry takmer úplne odstránené, keď hráč zadá príkaz, jednotky ho okamžite dostanú a vykonajú. Ak hráč príkaz nevydá, nikto sa z vlastnej vôle nepohne. V štandardnej RTS, pokiaľ hráč nedá príkaz, aby jednotky išli do krytia, tieto jednotky budú stáť nehybne na mieste, aj keby sú pod priamou paľbou.

Samozrejme, sú tu výnimky, v *Dungeon Keeper* hráč prijíma bytosti, ktoré prejdú cez portál, toto sú hráčove jednotky. Tieto jednotky sa pohybujú cez dostupné priestory, keď sú zranené idú do svojho domova, keď sú hladné idú sa najesť, ak sa stretnú s jednotkami, ktoré nemajú radi, začnú sa biť, ak nemajú dosť jedla alebo nedostanú výplatu, odídu. Všetko z vlastného predprogramovaného správania, hráč má len limitované prostriedky, ako toto správanie ovplyvniť. Stále je tu však ten aspekt, že hráč ovláda celý “dungeon”, ale nejakú základňu, spravuje jej financie, rozhoduje čo postaví a ktoré jednotky budú prijaté alebo vyhostené.

1.1 Delenie Strategických hier podľa štýlu interakcie

Keď už máme dobré pochopenie toho, čo sú strategické hry, povedzme si o dvoch základných kategóriách ktoré rozoznávame v tomto žánri.

1.1.1 Real-time strategy

Real-time strategy (RTS) je, v jednoduchosti, strategická hra, v ktorej hráči vykonávajú akcie v reálnom čase, čo v podstate vytvára z času ďalší zdroj, ktorý treba spravovať. Každý hráč musí pod časovým tlakom robiť rozhodnutia, ideálne podľa vopred pripraveného plánu, zároveň musí vedieť pohotovo reagovať na udalosti ako sa dejú počas hry.



Obr. 1.1: Koláž rôznych rts hier

Typická RTS hra obsahuje mechaniky pre zbieranie surovín, budovanie základne, výskum technológií a nepriame ovládanie jednotiek. Nie každá RTS hra však potrebuje všetky z týchto mechaník, obzvlášť ak odídeme z tematiky vojnových hier a pozrieme sa na city-building hry, kde nemám žiadne jednotky, ovládanie jednotiek nie je nutné.

1.1.2 Turn-based strategy

V turn-based stratégiách (TBS) funguje systém ťahov, ktoré sa striedajú medzi hráčmi, pričom každý hráč vie konať len počas vlastného ťahu a po naplnení istej podmienky sa ťah končí a posunie sa ďalšiemu hráčovi. Dá sa to predstaviť ako šach, kde podmienkou

je že hráč musí vykonať jednu akciu počas svojho ťahu, a po naplnení tejto podmienky sa ťah posunie oponentovi, kedy už hráč nemôže konať. V TBS je čas vnímaný odlišne ako v RTS, hráč môže mať počas svojho ťahu neobmedzené množstvo času, môže mať určené množstvo času na každý ťah, alebo určené množstvo času zdieľané medzi viacerými ťahmi. Každý hráč má však možnosť plánovať a zbierať informácie aj počas ťahu iného hráča, čo pridáva skrytý význam k efektívnosti využívania času počas svojho, ako aj nepriateľovho ťahu.



Obr. 1.2: Batalj, TBS videohra

1.2 Stratégia a Taktika

Skôr ako budeme pokračovať na detailnejšie delenie žánru, zastavme sa na moment pri téme stratégie a taktiky. Veľa ľudí rozlišuje medzi strategickými a taktickými hrami, niekedy až tak, že rozlišujú medzi žánrami real-time strategy a real-time tactics, analogicky turn-based strategy a turn-based tactics.

Pojmy stratégie a taktiky však nie sú exkluzívne pre videohry, podobne ako samotný koncept strategických hier, tieto pojmy vznikli z aspektov reálneho sveta. Lawrence Freedman v svojej knihe *Strategy: A History*, tvrdí že pojem stratégie sa nedá presne vymedziť a definovať. Plán popisuje postupnosť krokov na dosiahnutie určitého cieľa, dobrý plán umožňuje konanie s istotou. Jednoduchý plán sa však nepovažuje za stratégiu, namiesto toho, stratégia prichádza na scénu v momente, keď aj najlepší plán obsahuje neistoty, nepredvídateľné udalosti, potenciál konfliktu.

[11] *There is no agreed-upon definition of strategy that defines the field and limits its*

boundries. One common contemporary definition describes it as being about maintaining a balance between ends, ways, and means; about identifying objectives; and about the resources and methods available for meeting such objectives. This balance requires not only finding out how to achieve desired ends but also adjusting ends so that realistic ways can be found to meet them by available means.

Takže stratégia je o koncepte nejakého plánu, ako dosiahnuť nami požadované ciele, a ako tento plán prispôsobovať okolnostiam. Ale mať stratégiu stále nie je to isté ako mať dosiahnutý cieľ, stratégiu treba vykonať, a jednotlivými krokmi sa približovať k cieľu. Prechody medzi týmito samotnými krokmi, sú práve taktika.

V kontexte vojnovnej hry, by toto mohlo byť vnímané nasledovne:

Stratégia je o vytváraní plánu ofenzívy, ako chceme dobiť herné územia, a v každom kroku počas tejto ofenzívy analyzovať naše slabé body, na ktoré nepriateľ môže zaútočiť. Taktiež rozhodovaniu, ako správne a efektívne umiestniť naše jednotky a iné zdroje na územia, aby dokázali súčasne vykonávať ofenzívu, no zároveň pokryť slabé miesta.

Súčasťou stratégie môže byť taktiež diplomacia, a teda s ktorým hráčom chceme mier, alebo spojenectvo, na ako dlho a ako sa tejto okolnosti prispôbiť, prípadne ako sa pripraviť na možnosť že druhý hráč nevyhovie nášmu plánu.

Následne, taktika bude o umiestňovaní daných jednotiek, podľa našej stratégie, budovanie obranných aj útočných štruktúr, bojovanie jednotlivých bitiek, podnikanie konkrétnych krokov za účelom získať nami požadovanú diplomaciu.

Tieto koncepty je vcelku pekne vidno v neskoršom kapitole, pri téme Grand Strategy hier, ktoré z pravidla obsahujú obe vrstvy gameplaya, strategickú a taktickú.

Aj mimo GS, väčšina strategických hier obsahuje aspoň nejaký aspekt taktiky a taktických cieľov, avšak v poslednej dobe sa začalo objavovať viacej hier zameraných čisto na taktický gameplay, ktoré úplne vynechávajú strategický komponent medzi jednotlivými súbojmi. Príkladom takejto hry môže byť LotR: The Battle for Middle-earth, hra v koncepte podobná StarCraftu, avšak hráčové akcie v jednej misií, neovplyvňujú akcie v inej, misie sú v určenom poradí, a hráč medzi jednotlivými misiami nerobí žiadne rozhodnutia, každú misiu začína v naskriptovaných okolnostiach, takže celá stratégia je súčasťou naratívneho kontextu, a hráč nad tým nemá kontrolu, celá hra je vlastne séria taktických simulácií v ktorých hráč musí uspieť.

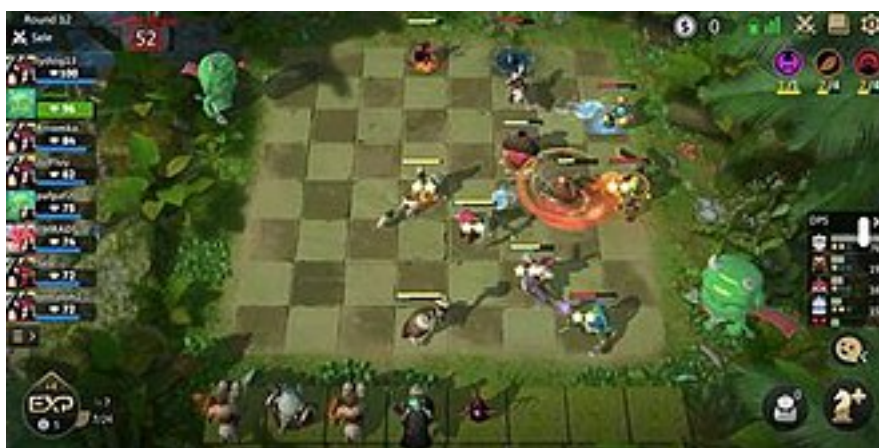
Pochopiteľne, čisto taktické hry nezaraďujem medzi strategické hry, čo je témou našej práce, napriek tomu sa však oplatí mať v týchto pojmoch jasno, kvôli taktickým prvkom nachádzajúcim sa v strategických hrách.

1.3 Delenie Strategických hier podľa kľúčových vlastností

Teória hier sa neustále vyvíja a producenti hier sa stále snažia prísť s novými mechanikami a špecifikáciami, mimo to aj technologický pokrok umožňuje vytvoreniu technicky náročnejších hier, kde môžu preraziť niektoré herné mechaniky, ktoré pred tým neboli prakticky použiteľné. Toto všetko vedie ku tomu, že veľa krát sa na trhu objaví celá skupina hier, založená na kľúčovej myšlienke alebo hernej mechanike. Podľa toho sa aj kategorizácia hier často upraví, pre efektívnosť marketingovej námahy. Každopádne, je prakticky nemožné jednoznačne vymenovať všetky kategórie a ich špecifikácie a koniec koncov, cieľom našej práce nie je zjednotiť kategorizáciu, my sa len snažíme popísať všeobecné charakteristiky celého žánru strategických hier a pozrieme sa na niektoré kategórie, aby sme si cez ne ukázali niektoré vlastnosti strategických hier, a ako prispievajú do identity strategických hier.

1.3.1 Auto-battlers

Kategória auto-battlerov, je žáner hier, kde si hráč kupuje a vylepšuje jednotky, rozmiestni ich na svoju hernú plochu a jeho jednotky následne bojujú proti inému, náhodne zvolenému, hráčovi, použitím spoločného herného AI pre bojujúce jednotky. Vyhráva hráč, ktorého jednotky sú posledné žijúce alebo hráč, ktorého jednotky majú najviac života/početnosti na konci časomier, prípadne iné upravené podmienky. [3]



Obr. 1.3: Dota auto-chess

Základná mechanika týchto hier spočíva v automatickom prebiehaní súbojov. Je tu upravený prístup k turn-based fungovaniu hry, a to v tom, že hra má dve rôzne fázy, plánovanie a bojovanie. Vo fáze bojovania majú hráči obmedzenú až nulovú možnosť ovplyvňovať hru, môžu však zbierať informácie pozorovaním súbojov alebo rozmiest-

nenia jednotiek iných hráčov. Vo fáze plánovania musia hráči uplatniť nazbierané informácie, aby upravili svoju hernú plochu a pripravili sa na ďalší súboj. Práve fáza plánovania vyzerá ako časť, kde sa v klasickej tbs hre hráči majú striedať a každý robiť vlastné ťahy, avšak vo väčšine auto-battlerov počas fázy plánovania hráči nevedia navzájom priamo interagovať, čo im umožňuje, aby mohli konať všetci naraz na vlastnej ploche. Týmto spôsobom sa ťah neposúva medzi hráčmi, ale fungujú len fázy, v ktorých majú hráči obmedzené niektoré činnosti. Zbieranie informácií je veľká časť týchto hier, pochopiť, ako fungujú systémy automatického súboja, ktoré jednotky dobre vychádzajú navzájom, ktoré jednotky je dobré postaviť proti nepriateľovým jednotkám, ako správne umiestniť svoje jednotky, sledovať a predvídať, ako sa vytvárajú páry, a z toho vydedukovať kto bude môj najbližší súper, sledovať ako funguje zbieranie jednotiek a využiť tieto vedomosti na poškodenie taktiky iných hráčov alebo posilnenie tej vlastnej.

1.3.2 Tower defense

Tower defense hry sú založené na jednom hlavnom princípe, hráč má pod kontrolou istú oblasť, na ktorú chodia nájazdy nepriateľov a hráčovou úlohou je obrániť túto oblasť, zvyčajne sú hráčovým hlavným nástrojom automaticky fungujúce obranné veže [21]. Na prvý pohľad to môže pôsobiť ako TBS, hra funguje v dvoch fázach, vo fáze prípravy máme čas postaviť obranné jednotky a môžeme si prečítať aké útočiacie jednotky prídu v najbližšej vlne, potom príde fáza obrany, kde sa nepriateľské jednotky snažia dostať cez vybudované obrany. Avšak, väčšinou hráč môže konať aj počas fázy, kedy prichádzajú nepriateľské jednotky, čo dodáva viac hĺbky do možností plánovania (na začiatku postaviť obrany najbližšie k miestu od kiaľ prichádzajú útočníci, suroviny získané z padnutých útočníkov môžeme počas útoku použiť na budovanie ďalších obranných štruktúr počas toho, ako postupujú ďalej do nášho územia).

Práca s informáciami je znovu veľmi dôležitá, nakoľko veľmi často v týchto hrách máme rôzne typy obranných štruktúr a rôzne typy útočníkov, podľa toho aký útočníci prichádzajú chceme postaviť vhodné druhy obranných veží, pričom tieto faktory často nie sú explicitne povedané, môže ísť o rýchlosť útočiacich jednotiek, ich počet, ich odolnosť, ako veľmi vzdialené od seba sú, niektoré jednotky môžu mať priamu oddolnosť alebo slabosť, voči špecifickým typom veží.

1.3.3 Real-World-Time

Veľmi špecifický štýl hier, ktoré sú určené na hranie počas dlhého obdobia (väčšinou týždne až mesiace, no niekedy aj neobmedzene dlho), kľúčovou vlastnosťou je, že počas tohto obdobia hra beží nepretržite, aj keď sú všetci hráči odpojení. Hráči si počas

1.3. DELENIE STRATEGICKÝCH HIER PODĽA KLÚČOVÝCH VLASTNOSTÍ11



Obr. 1.4: Kingdom Rush

tohto obdobia kultivujú svoj účet, často krát sa hráči schádzajú v spoločenstvách a spolu pracujú za nejakým cieľom. Tento cieľ môže byť viac abstraktný, ako pozícia na rebríčku, alebo môže byť viac konkrétny, ako dobytie určitej časti zeme(hernej oblasti), pričom v neskoršom prípade, kde je určený konkrétny cieľ, väčšinou po dosiahnutí cieľa daný server zaniká, keďže už má víťaza. Tieto servery sa znova spúšťajú v určitom intervale. V niektorých prevedeniach môžu byť spoločné ciele menej dôležité, obzvlášť hry, ktoré sa sústreďujú na viac oddechovú skúsenosť, kde sa proste hráč stará o svoj účet, súťaží s ostatnými hráčmi kto získa viac bodov, ale hra sa nesnaží budovať tlak na hráča. Iné hry v tejto kategórii, sa však zameriavajú viac na vojnovú stratégiu. Hráči si stavajú mestá a budujú armády, pomocou svojich armád sa snažia dobyť ďalšie mestá alebo zničiť ohrozujúceho nepriateľa.

V spolkoch je veľký dôraz na koordináciu, hráči môžu vyslať svoje armády na útok na nepriateľa, jednotky však cestujú svojou predurčenom rýchlosťou, často minúty až desiatky minút na políčko vzdialenosti, čo dáva obrancovi čas reagovať. Rôzni hráči, alebo aj rovnaký hráč útočiaci z viacerých miest, musí vedieť vhodne načasovať útoky, aby dorazili naraz, prípadne vo vhodnom poradí, aby prelomili obranu objektu. Sú tu mnohé taktiky, napríklad ponúknuť nepriateľovi falošné informácie, čo ho dovedie k obraňovaniu falošného cieľa, čím neobráni skutočný cieľ. Ak sa spolku pod útokom



Obr. 1.5: Divoké kmeny

podarí zistiť správny cieľ útoku, stále je na mieste rozhodnúť sa ako ho ubrániť, a či to vôbec tento cieľ ubrániť. Taktika a stratégia, so zameraním na dlhodobý výsledok, je to najdôležitejšie v týchto hrách.

1.3.4 4X

Tento názov pochádza z anglických slov Explore, Expand, Exploit, Exterminate, ktoré popisujú základné mechaniky a hráčov hlavný cieľ v týchto hrách. Ide zväčša o vojnové hry, kde účelom je preskúmať neznáme prostredie, vybudovať si stabilnú základňu operácií za pomoci nájdených surovín, a vyhladiť nepriateľov, získať totálnu kontrolu nad hernou plochou. Tieto hry takmer vždy budujú na real-time strategy princípoch, takže hráči konajú v reálnom čase a ich príkazy sú hneď zaznamenávané, čo vytvára priestor pre istú dominanciu cez mechanické schopnosti, je dôležité mať oči na každom mieste mapy a vedieť pohotovo reagovať na akcie útočníka, vzhľadom na niektoré doplnkové mechaniky, napríklad špeciálne schopnosti jednotiek, môže výsledok bitky medzi dvoma jednotkami rozhodnúť to, ktorý hráč dá skôr povel na použitie tejto schopnosti. Ale toto sú len malé taktické víťazstvá, nakoľko hráči ovládajú celé základne, a často krát rozmerné armády, väčšinou vyhráva hráč s lepším plánom hry alebo sa lepšie prispôsobí taktike nepriateľa.

Tieto hry zväčša využívajú nejaké mechaniky zatajovania informácií, pre každého hráča je viditeľná len časť mapy, na ktorých má jednotky alebo má obsadené strategické body, takže hráč väčšinou nevidí základňu, ani pohybujúce sa jednotky nepriateľa, je tu, teda taká hra v hre, kde sa hráč pokúša nazbierať čo najviac informácií o nepriateľovi, aby vedel predpovedať jeho ďalší krok a zároveň nedovoliť nepriateľovi získať dôležité



Obr. 1.6: Age of Wonders: Planetfall

informácie, ktoré jemu umožnia zistiť, na čo sa má pripraviť.

1.3.5 Grand strategy

Grand strategy game, alebo len GSG, je vojnová hra so zameraním na vojenskú stratégiu a manažovanie surovín a zdrojov na úrovniach konfliktu celých štátov, alebo impérií. Z technického hľadiska je táto kategória veľmi podobná na 4X, hlavný rozdiel je v škále a spôsobe ovládania. Zatiaľčo v bežnej 4X hre hráč ovláda jednu až niekoľko základní a počet ovládaných jednotiek sa pohybuje zväčša v dvojciferných hodnotách, čo umožňuje hráčovi rozumne ovládať armády na úrovni jednotlivých jednotiek, v GSG toto nie je možné, nakoľko hráč ovláda oveľa masívnejšie množstvá infraštruktúry aj jednotiek, často krát na úrovni, kde hráč iba vydáva rozkazy, štýlu “x jednotiek zaútočiť na miesto y”. Grand Strategy hry sa často sústreďujú na simulácie podobné skutočnému svetu, do miery, kde samotný názov tejto kategórie je založený na pojme “grand strategy” [23], čo je pojem používaný pre stratégiu krajín pre posúvanie a získavanie národných záujmov.

Mimo to, čo je prakticky civilné vyučovanie histórie, však Grand stratégie, ako napríklad už spomínaný Kriegsspiel, boli používané pre edukatívne účely špecializovaného vojenského personálu v oblasti vojrovej stratégie.



Obr. 1.7: Field of Glory: Empires

1.4 Známe strategické hry

Teraz si prejdeme zopár konkrétnych príkladov strategických hier, pozrieme sa na ich prevedenie a rozoberieme si, ako využívajú niektoré vlastnosti a mechaniky.

Vzorka bola vyberaná na základe dvoch hlavných kritérií, chcel som s relatívne malým množstvom hier pokryť čo najviac odlišných prvkov a kategórií, takže som sa snažil nevyberať navzájom podobné hry, zároveň som sa snažil vyberať hry s kvalitným preverením. Kvalita prevedenia je však, samozrejme, veľmi subjektívna záležitosť, takže pre potvrdenie môjho ohodnotenia kvality som sa pozeral na popularitu a komerčný úspech hier.

1.4.1 StarCraft

starcraft2.com/en-us/game

[8] *In StarCraft II, players juggle big-picture economy and micro-level unit control, pushing them to the very limits of their strategic and tactical capabilities.*

Populárna RTS hra s prvkami 4X, Starcraft má pomerne stabilnú eSportovú scénu a je trvale zarytá v srdciach mnohých hráčov. Starcraft obsahuje mnohé štandardné špecifikácie 4X hier, hráč zväčša začína s malou skupinkou jednotiek, ktoré sú zamerané na budovanie základne operácie. Na začiatku je dôležité nájsť náleziská surovín a začať budovať základňu, trénovať jednotky, preskúmať okolie, zistiť kde sú ďalšie výhodné suroviny, a kde sa nachádzajú nepriatelia. No a na koniec vojna do posledného muža,



Obr. 1.8: Titulný obrázok StarCraft 2

hráč, alebo hráči, posielajú svoje jednotky zaútočiť na nepriateľské základne, vytvárajú obrany vlastnej základne, aby odolali nepriateľským útokom, vylepšujú vlastné jednotky podľa potreby. Väčšina hier v tomto žánri, obsahuje ďalšie podporné mechaniky, ktoré sú väčšinou prebrané z iných žánrov, StarCraft obsahuje hrdinov, čo sú špeciálne jednotky, ktoré majú jedinečné schopnosti, detailnejšie systémy vylepšení, hrdinovia sa správajú ako hrdinovia z RPG žánru, avšak nemajú až tak veľkú dôležitosť, pretože je tu stále potreba manažovať celú základňu a hromadu iných jednotiek.

K manažovaniu základni a jednotiek vieme identifikovať dva základné typy manažovania, micro-management a macro-management.

Micro-management je manažovanie situácie na malej škále, napríklad ak hráč vyšle skauta na preskúmanie niektorej oblasti, hráč musí ovládať jednotku, kam a ako sa má hýbať, možno sa oplatí preskúmať každý roh, ak očakávame, že sa tam môžu skrývať nepriatelia alebo bonusy. Ak ovládame robotníkov, chceme aby neustále pracovali. Pokiaľ máme dosť zdrojov, chceme aby pracovali na nových budovách, ktoré nám umožnia nový výskum alebo trénovať nové jednotky. Pokiaľ dosť zdrojov nemáme, tak chceme aby zbierali zdroje z okolia základne. A znova, ak ovládame základňu, chceme sledovať vývojový strom budov a výskumov, ktoré naša základňa obsahuje, mnohé obsahujú špeciálne odvetvia výskumu, ktoré nám dajú isté výhody, alebo trénovanie špeciálnych typov jednotiek. Je asi ľahké vidieť, že je tu veľa rôznych vecí na spravovanie a, v zmysle fungovania RTS, prebiehajú všetky naraz.

Tu prichádza macro-management, čo je zložka hráčovských schopností spravovať všetky tieto menšie systémy súčasne, ako celok väčšieho systému. Tu stojí za zmienku, že štandardne býva škála hernej mapy pomerne rozsiahla a hráč pochopiteľne nemôže vidieť

do detailu každú časť mapy, preto sú praktické viaceré móde pozerania sa na svet, viac detailný náhľad na malú plochu mapy, toto je štandardne využívané pre micro-management, a menej detailná minimapa, ktorá zobrazuje väčšiu, ak nie rovno celú hernú plochu, avšak v minimálnom detaile.



Obr. 1.9: Obrázok herných prvkov SC2 z pohľadu hráča

Na obrázku 1.9 môžeme vidieť ukážku toho, ako hráč pristupuje k herným prvkom. Väčšina obrazovky je sústredenej na detailnú mapu, na dolnej strane obrazovky sú viaceré lišty, ktoré zobrazujú minimapu a panel s príkazmi pre použitie na minimape, napravo od nej je panel selekcie budov a jednotiek, a napravo od toho je zoznam príkazov pre aktuálne zvolené jednotky alebo výskumné/trénovacie možnosti pre zvolené budovy. Na vrchu obrazovky sú vypísané suroviny.

Táto minimapa je esenciálna pre macro-management, pretože ju hráč môže používať, aby rýchlo preskakoval medzi rôznymi časťami hernej mapy a micro-manažoval situáciu v jednotlivých menších častiach. Jedna fronta vecí na spravovanie je základňa, je potrebné mať dostatok robotníkov, ktorý, ak má hráč dosť zdrojov, môžu rozširovať základňu novými výskumnými a produkčnými budovami, obrannými štruktúrami, opravami poškodených budov alebo budovaním nových základní vo vzdialených oblastiach za víziou strategicky výhodnej pozície, alebo iných zdrojov. Naopak, ak hráč nemá dosť zdrojov na budovanie a rozširovanie, je dôležité aby robotníci zbierali dané zdroje a nesedeli nečinne.

Ďalej je nutné spravovať takmer každú budovu, jej vlastným stromom výskumu vylepšení základne alebo jednotiek, alebo výcviku jednotiek v produkčných budovách.

Nakoľko každý výskum aj tréning jednotiek trvá nejaké zdroje na počiatku procesu, ale potom trvá nejaký čas, kým sa ukončí a hráč vie využívať benefity daného procesu, je dôležité správne spravovať dostupné zdroje. Je možné využiť frontu príkazov, kde hráč budove naraz vydá viacej príkazov a budova ich bude vykonávať v zadanom poradí, čo výrazne znižuje réžiu vyžadovanú od hráča na spravovanie, ale toto vyžaduje, aby hráč vopred zaplatil všetky potrebné zdroje. Nakoľko príkazy vo fronte sa budú vykonávať postupne jeden za druhým, väčšina týchto zdrojov nebude okamžite využitá, toto preto môže byť neefektívne a je lepšie aby hráč frontu nevyužil, a dané zdroje radšej použil v inej budove ktorá môže produkovať súčasne s prvou budovou, a tým hráč dostane rýchlejšie úžitok za minúté zdroje. S rastúcim počtom budov však masívne narastá vyžadovaná réžia od hráča všetko korektne spravovať.

Popri tom musí hráč spravovať už zmienené vycvičené jednotky, ktoré môžu útočiť, obraňovať, alebo sa skrývať pred detekciou nepriateľa. V samotnej bitke jednotiek je taktiež vlastná úroveň micro-managementu, hráč môže určovať pozície jednotlivým jednotkám, používať vyskúmané schopnosti, posilňovať prebiehajúci konflikt inými jednotkami, je tu masívny repertuár taktík a zručností, ktoré hráč môže aplikovať do každej menšej bitky, všetky ale vyžadujú obrovskú réžiu a schopnosť pohotovo reagovať.

Hráč môže využívať celý sortiment bojových taktík a stratégií ako získať lepšiu pozíciu v každej bitke, a priblížiť sa tak k celkovému víťazstvu. Z grafického hľadiska, StarCraft pôsobí ako bitka armád, ale v skutočnosti je to bitka plánov, a vyhrá hráč, ktorý vie lepšie prispôbiť svoj plán, pohotovejšie reagovať na akcie protihráča a efektívnejšie manažovať svoje základne a jednotky.

1.4.2 Into the Breach

subsetgames.com/itb.html

[14] *In Into the Breach you must control powerful mechs from the future to hold off an alien threat. Each attempt to save the world presents a new randomly generated challenge in this turn-based strategy game.*

Na prvý pohľad klasické prevedenie turn-based hry, ktorej gameplay je založený na súboji jednotiek dvoch tímov, na 2D štvorcovej ploche. Je tu, ale zopár veľmi zaujímavých odlišností, ktoré už nie sú také klasické. Hlavná vlastnosť súbojov, okolo ktorej je celá hra navrhnutá a dáva tomu úplne iný rozmer, oproti podobným hrám, je ten fakt, že takmer každý útok ovplyvňuje pozíciu zasiahnutej/zasiahnutých jednotiek. Väčšinou to funguje štýlom, že jednotka na krátku vzdialenosť udrie svoj cieľ, ktorý, popri tom, že obdrží poškodenie z útoku, bude odtlačený o niekoľko políčok v smere útoku. Alebo diaľkový útok na isté políčko, ktorý posunie jednotky na susedných políčkach.



Obr. 1.10: Titulný obrázok Into the Breach

Popri tom táto hra obsahuje mnoho interakcií kolízií, napríklad ak dve jednotky narazia do seba (jedna jednotka je odtlačená smerom do inej) obe tieto jednotky obdržia bonusové poškodenie. Ak je jednotka odtlačená do pevného terénu (špeciálne políčka, ktoré reprezentujú solídny objekt) táto jednotka obdrží bonusové poškodenie. Ak je jednotka vtlačená do vody, a nevie plávať, utopí sa a je okamžite odstránená z hry.

Veľmi dôležitá podporná mechanika k tomuto je, že nepriateľské jednotky telegrafujú svoj nasledujúci ťah, čo znamená, že ak nepriateľská jednotka vykoná svoj ťah, hneď naznačí, čo plánuje spraviť najbližšie kolo. Napríklad, zaútočiť na políčko do ľava. Pokiaľ sú jednotky posunuté nejakou zo skôr spomenutých mechaník, tak vykonajú svoj ťah, pokiaľ možno, podľa pôvodného plánu. Takže ak v predchádzajúcom príklade nepriateľská jednotka plánuje zaútočiť doľava, čím by trafila priateľskú jednotku, ale my zaútočíme na túto nepriateľskú jednotku a posunieme ju o 1 políčko dole, keď bude vykonávať svoj ťah, táto jednotka zaútočí doľava z novej pozície, na ktorú bola posunutá, čím minie našu priateľskú jednotku. Navyiac, ak na tomto políčku stojí iná nepriateľská jednotka, tak zaútočí na túto. Ak táto druhá jednotka na tento útok umrie, tak už svoj útok nevykoná. Poradie v akom nepriateľské jednotky budú vykonávať svoje ťahy sú hráčovi taktiež vopred oznámené.

Toto dáva hráčovi početné možnosti narušiť nepriateľské plány, robí to z každého ťahu takmer puzzle, ktoré je o hľadaní najlepších krokov, ktoré môže hráč so svojimi jednotkami vykonať.

Ku tomuto pridajme fakt, že počas štandardnej misie v Into the Breach má hráč obmedzený počet jednotiek, s ktorými začne misiu, zatiaľ čo nepriateľské jednotky prichádzajú vo vlnách v určených intervaloch, takže hráč má obmedzený počet ťahov, aby sa vysporiadal s nepriateľskými jednotkami, ochránil svoje jednotky, a splnil rôzne vedľajšie objektívy (ochrániť alebo napadnúť vopred určené špeciálne políčka, a podobne).



Obr. 1.11: Herné prvky Into the Breach z pohľadu hráča

na obrázku 1.11 môžeme vidieť 2D mapu, na ktorej misia prebieha, priateľské a nepriateľské jednotky sú rozlíšené farbou indikátoru nad jednotkou, vidíme indikátory možných ťahov zvolenej jednotky (zeleno podfarbené políčka) a taktiež indikátor nepriateľských pohybov (červene škárované políčka sú ciele útoku nepriateľských jednotiek)

V tejto hre je veľmi dôležité, aby si hráč dôkladne premyslel, ako pristúpiť ku každej misií, aby vykonal ťahy, ktoré ho nevystavia zbytočnému nebezpečenstvu, ale zároveň sa musí pripraviť aj na budúce ťahy inak bude rýchlo zamorený prichádzajúcimi vlnami nepriateľov a stratí šancu uspieť.

1.4.3 XCOM

xcom.com/xcom-enemy-unknown

[12] *As the commander of XCOM, you control the global defense team by creating a fully operational base, researching alien technologies, planning combat missions, and controlling soldier movement in battle.*

Veľmi populárna turn based hra zameraná na bojovú taktiku. Hráč ovláda malú skupinku vojakov a jeho úlohou je bojovať proti nepriateľskej skupinke vojakov. Namiesto štandardného stiedania ťahov medzi hráčmi, má každá jednotka svoje vlastné počítadlo a podľa toho sa určuje poradie, ktorá jednotka je práve na ťahu.

Jednotiek zvykne v každej hre byť naraz rádovo okolo desať, jednotka môže počas svojho ťahu používať kombinácie nasledujúcich ťahov: pohybovať sa po štvorcovej hernej ploche, používať špeciálne schopnosti, alebo zaútočiť ak má jednotka výhľad a

dostrel na nepriateľskú jednotku. Na spomínanej hernej ploche sa na nachádzajú rôzne štruktúry, ktoré slúžia ako krytie pred nepriateľskou strelbou z istého smeru, vysoké steny poskytujú úplne krytie, takže cez ne nie je možné strieľať ani hádzať objekty, potom sú tu nízke steny, za ktoré sa jednotka vie prikrčiť a pridá jej to výraznú šancu vyhnúť sa prichádzajúcej strelbe, avšak ponad nízku stenu je možné hádzať objekty ako granáty, ktoré môžu ohroziť jednotky v krytí.



Obr. 1.12: Ukážka z tutoriálu XCOM 2

Na obrázku 1.12 môžeme vidieť ťah jednotky a pohyb po hernom poli, s indikáciou políčka kde sa jednotka vie kryť za bariérou.

Takže, je tu veľmi silný dôraz pre správne umiestňovanie jednotiek, na konci ťahu chceme, aby naša jednotka bola dobre chránená, zároveň sa však musíme dostať do pozície, aby sme mohli útočiť na nepriateľské jednotky, čo prináša potrebu balancovať potenciálne riziká a zisky, nakoľko oba elementy sú nevyhnutné pre úspešnú hru. Sú tu však aj RNG elementy, ktoré robia tieto kalkulácie menej predvídateľné, jednotky môžu počas strelby netrafiť, čím nespôsobia žiadne poškodenie, nepriateľské jednotky vedia zasiahnuť kritickou ranou, čím spôsobia extrémne poškodenie, a podobné mechaniky, ktoré sa v rôznych vydaniach mierne líšia.

1.4.4 Factorio

www.factorio.com

[25] *Factorio is a game in which you build and maintain factories. You will be mining*



Obr. 1.13: Titulný obrázok Factorio

resources, researching technologies, building infrastructure, automating production, and fighting enemies. Use your imagination to design your factory, combine simple elements into ingenious structures, apply management skills to keep it working, and protect it from the creatures who don't really like you.

Hra s názvom Factorio predvádza zaujímavý zvrät na kategóriu city-buildov. Hráč začína ako jednoduchá postavička uprostred pustatiny, s jedinou úlohou, zbierať suroviny a zarábať. Na každom kroku po tejto pustatine sú rôzne typy surovín, ktoré hráč môže ťažiť. Na začiatku hráč suroviny ťaží ručne, ale po chvíli dokáže z nazbieraných surovín vytvoriť automatické extraktory, ktoré ťažia svoju surovinu z daného miesta automaticky. Hráč tieto extraktory musí pravidelne vyberať, lebo vedia extrahovať suroviny len po obmedzenú kapacitu, pred tým než sa naplnia a prestanú ďalej extrahovať.

Extraktory však môžu získavať suroviny rýchlejšie, než ich hráč stíha mŕňať, čo rýchlo vyťaží jeho kapacitu surovín, ktoré môže nosiť v inventári, preto potrebuje postaviť sklad, ako miesto, kde môže hráč extrahované suroviny odkladať. Niektoré stavby potrebujú vklad istých surovín, aby mohli pracovať a produkovať ďalšie, vzácnejšie, suroviny.

Vyberať každý extraktor manuálne, môže byť pri dosť veľkom počte extraktorov neefektívne, preto môže hráč zainvestovať do dopravných pásov a systému, ktorý umožní extraktorom automaticky posielat' vyťažené suroviny cez dopravné pásy priamo do skladu. Následne z takéhoto skladu môže ísť rad dopravných pásov do štruktúr, ktoré potrebujú suroviny, ktoré prichádzajú do daného skladu a tým automatizovať celý pro-

ces výroby aj zložitejších a vzácnejších surovín.

V tomto bode už vidno ideu hry, ako hráč pokračuje v zbieraní surovín, stavia viac extraktorov, získava suroviny rýchlejšie, stavia pokročilejšie štruktúry, ktoré tieto suroviny ďalej menia a umožňujú ešte pokročilejšie stavby, ktoré vyžadujú ešte viac surovín na vstupe a produkujú ešte vzácnejšie suroviny. Pomocou dopravných pásov a rôznych systémov dokáže automatizovať tento proces a prepájať aj veľmi vzdialené systémy, ktoré majú prístup k potrebným surovinám. Takto môže postupom času vybudovať rozsiahle impérium ťažkého priemyslu, extrahovania surovín, následného zúšľachtovania surovín na vzácnejšie suroviny. Táto hra poskytuje prístup k obrovskému otvorenému svetu a masívnej hierarchie rôznych surovín a budov, ktoré tieto suroviny spracúvajú alebo produkujú.



Obr. 1.14: Snapshot továrne z Factorio

Na obrázku 1.14 môžeme vidieť ukážku masívnej infraštruktúry, po desiatkach hodín ťaženia a budovania.

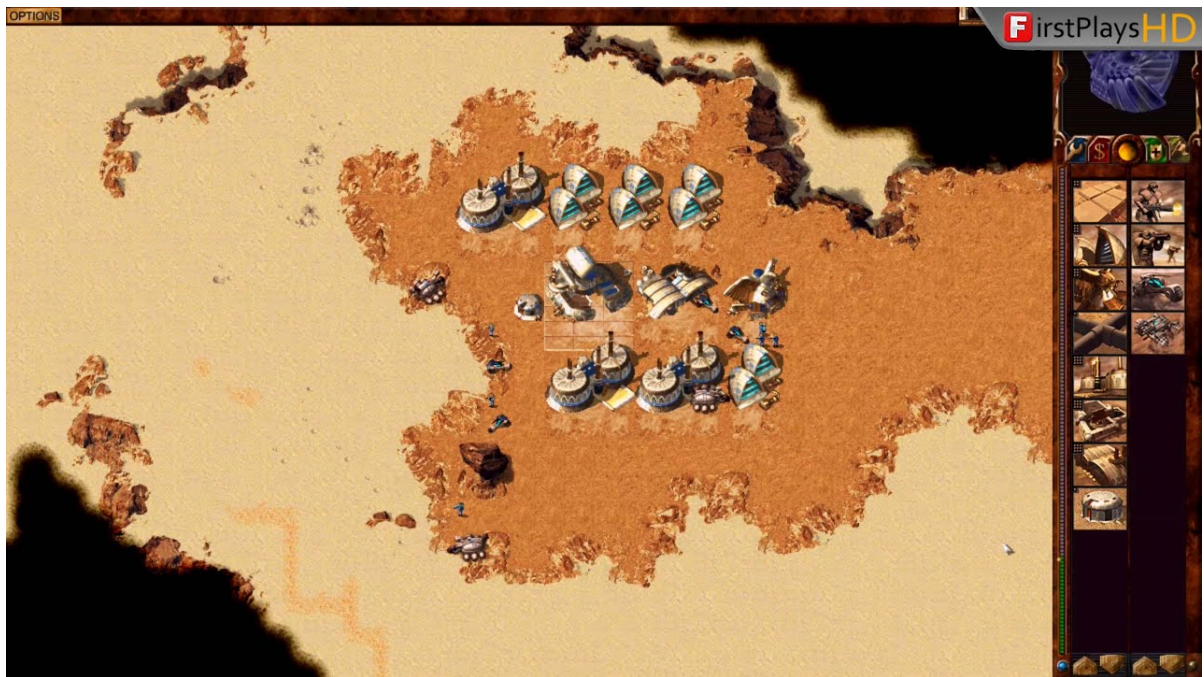
Je asi ľahko vidno, že v nejakom sandbox móde toto môže byť kludná a relaxačná hra, kde si hráč proste buduje továrne, ale za pomoci rôznych podporných mechaník, ale ekonomika, kde hráč potrebuje peniaze, aby mohol odomykať nové štruktúry alebo aj samotné stavenie budov, popri klasickej cene surovín, peniaze hráč získava len predajom štruktúr, alebo nazbieraných surovín. Ešte na to pridajme systém daní, kde hráč musí každé určité časové obdobie zaplatiť poplatok, ktorý sa zvyšuje každým obdobím.

Zrazu sa kludná sandboxová hra zmení na hru o pretekoch vo vývoji, efektivita je absolútne kľúčová, hráč nechce postaviť drahú továreň, ktorá nebude môcť pracovať, lebo nemá požadovaný príjem pre produkciu. Hráč musí taktiež sledovať prípadnú ekonomiku, aby vedel produkovať čo najvýhodnejšie suroviny na predaj, aby si mohol dovoliť uhradiť všetky poplatky aj cenu rozvoja a rastu.

1.4.5 Dune

[15] *Dune consists of two interwoven gameplay layers: an adventure layer reminiscent of dungeon crawler games, and a top-down strategy layer more reminiscent of 4X games.*

Hoci je Dune mix medzi strategickou a adventúrnou hrou, dobrodružná časť je využívaná hlavne pre naratívne účely príbehu, gameplay je silno založený na strategických prvkoch. Hráč má od mimoplanetárnej mocnosti úlohu spravovať ťažný priemysel na planéte, kde sa hra odohráva. Avšak, táto planéta je obsadená iným impériom, ktoré má taktiež záujem o ťaženie na tejto planéte a majú pod kontrolou silnú armádu. Ďalšou úlohou hráča je zničiť celé konkurenčné impérium. Navyše sa tu nachádza aj ďalšia strana lokálnych obyvateľov, ktorých namiesto ťažby zaujíma ekológia planéty, hráč musí pomocou lokálnych kmeňov udržiavať ekológiu planéty, aby ostala obývateľná.



Obr. 1.15: Náhľad na herné prvky strategickkej vrstvy Dune

Takže, hráč musí spravovať svoju ťažobnú a vojenskú silu a konkurencie schopnosť, zároveň musí udržiavať ekológiu planéty, ktorá často prichádza na úkor efektivity ťažby, takže je potrebné mať plán, ako balancovať priority a pridelovať prostriedky adekvátne

na jednotlivé účely.

1.4.6 Heartstone

playhearthstone.com/en-us

[9] *Welcome to Hearthstone, the strategy card game that's easy to learn but impossible to put down!*

Kartová hra založená na koncepte duela dvoch hráčov, reprezentovanými svojím avатарom, ktorý má dostupný vopred vybraný balíček herných kariet, ktoré vie počas duelu hrať. Cieľom duelu, alebo zápasu, je zničiť nepriateľského avatara.

Keďže táto hra nie je v štýle jednorázového vydania, namiesto toho sa neustále updatuje a vyvíja priebežne, v priebehu rokov sa tu nazbieralo obrovské množstvo rôznych mechaník, ale základný gameloop sa dá popísať nasledovne:

Na začiatku máme hernú plochu, na krajoch sú proti sebe uložení avatari hráčov, každý avatar má svoj život a dosiahnutím hodnoty 0 životov tento hráč prehráva. Každý hráč má okrem avatara svoj balíček 30 vopred vybraných kariet, každý hráč si na začiatku svojho kola potiahne jednu kartu, pričom na začiatku prvého kola začína s vyšším počtom kariet. Každý hráč vie na ruke držať obmedzený počet kariet, ak si potiahne novú kartu v čase, keď má plnú kapacitu kariet na ruke, táto karta zhorí a je odstránená z tohto duelu.

Medzi avatarmi je herná plocha, kde sa odohráva väčšina bežného zápasu. Hráč môže počas svojho ťahu hrať karty z ruky, každá karta potrebuje skonzumovať nejaký počet mana kryštálov, aby mohla byť zahraná, hráč má obmedzene veľa mana kryštálov, ale na začiatku každého kola získa jeden nový kryštál až po maximálny počet, navyše na začiatku kola sa obnovia všetky jeho prázdne mana kryštály. Karty môžu mať 2 hlavné efekty.

Zahraná karta môže na hernej ploche vytvoriť miniona, čo je menšia postava, ktorá má život a silu útoku, poprípade ešte ďalšie efekty. Každé kolo má každý minion jeden útok, ktorý vie hráč použiť, aby s daným minionom zaútočil na dostupné nepriateľské postavy.

Druhá možnosť je, že karty spôsobia nejaký okamžitý efekt, typickým príkladom takýchto kár sú kúzla, ktoré môžu spôsobiť poškodenie, potiahnuť karty z balíčka do ruky, alebo aj vyvolať miniona, alebo kombinácia niekoľkých efektov. Samozrejme, v skutočnosti je tento gameloop o trochu komplikovanejšími, následkom neustáleho vytvárania nových herných mechaník a nových druhov kariet, ale pre predstavu je tento



Obr. 1.16: Herná plocha v zápase Hearthstone

popis postačujúci.

Je tu rozpor, či kartové hry patria medzi strategické hry, medzi taktické hry, alebo sú mimo oba žánre a tvoria svoj vlastný, menší žáner. A je pravda, že kartové hry sú viac zamerané na taktiku a dosahovanie krátkodobých cieľov, ako získať čo najlepší ťah práve teraz, ale je tu aj aspekt vytvorenia si herného plánu, ktorý zakomponuje celý balíček 30 kariet, kde každá karta je jeden schod, ktorý napĺňa špecifickú úlohu a prispieva k vyššiemu cieľu, potom hráč musí aplikovať správnu taktiku, aby prešiel jednotlivé ťahy, čo najefektívnejšie sa priblížil k danému cieľu, ktorý mu v ideálnom prípade zaručí výhru. Takýto pohľad na budovanie balíčku sa rozhodne dá nazvať stratégiou, a preto som sa rozhodol spomenúť tento štýl hier ako strategické hry.

1.4.7 Frostpunk



Obr. 1.17: Snapshot mesta Frostpunk v skorej fáze hry

www.frostpunkgame.com

[13] *Frostpunk is a city-builder survival where every decision comes with a cost. Frostpunk challenges one's tactical skills, while questioning his morality.*

Frostpunk je vcelku netradičný city-builder, ktorý sa sústreďuje na vytváranie rozhodnutí v štýle *make your own journey*. Väčšina rozhodnutí je postavených do kontroverznej morálnej naratívy, avšak v zmysle city-building stránky tejto hry, je tu nezanedbateľný dôraz na manažovanie surovín a plánovaného vývoja mesta, hráč musí zabezpečiť adekvátne rozdelenie pracovnej sily na zbieranie esenciálnych surovín, spolu s rastúcou populáciou musí hráč zabezpečovať rast mesta a udržiavať dostupnosť potrebných surovín, popri tom však musí udržiavať aj vnútro-politickú situáciu, ktorá je prepojená s morálnym kodexom tejto hry, pokiaľ hráč spraví priveľa rozhodnutí ktoré sa jeho poddaným priečia, môže rýchlo ukončiť svoje obdobie vládnutia.

1.4.8 Total War

Hoci je väčšina Total War hier marketovaných ako RTS hry, v skutočnosti tieto hry fungujú cez dve roviny gameplayu.



Obr. 1.18: Náhľad strategickej mapy v Total War saga:Troy

Prvá rovina, ktorú hráč stretne, je turn-based stratégia, hráč operuje na strategickej mape, ktorá je rozdelená na menšie územia, každé územie môže byť ovládané najviac jedným hráčom súčasne. V tejto fáze môže hráč manažovať svoje impérium, stará sa o rôzne zdroje, ovládané armády, peniaze, môže vylepšovať bojové jednotky, rozmiestňovať armády po mape, aby dokázal obrániť všetky svoje dôležité územia a zároveň naplňať svoje útočné plány. V rôznych implementáciách Total War môžu byť vnútropolitické aspekty, hráč sa musí starať o spokojnosť svojich obyvateľov, inak môžu začať rebelovať, čo môže mať seriózne následky.

Keď hráč rozmiestni svoje jednotky a príde do situácie vojnového konfliktu, čo môže nastať, ak hráč pošle svoje jednotky na nepriateľské územie s armádou, alebo nepriateľské jednotky vstúpia na hráčové územie a hráč tu má jednotky, ktoré môžu brániť toto územie, potom hráč vstúpi do simulácie bitky, oveľa detailnejší náhľad na oblasť, kde sa bitka odohráva, hráč dokáže ovládať jednotlivé skupinky jednotiek. Takáto bitka sa odohráva v reálnom čase, takže tieto konflikty prebiehajú v rovine RTS. Hráč si tu môže navrhnuť útočný alebo obranný plán, zvoliť formáciu jednotiek, pripraviť prípadnú mašinériu, mnoho ďalších taktických rozhodnutí.



[t]

Obr. 1.19: Náhľad okupovania nepriateľskej pevnosti v Total War RTS vrstve

Názov hry	Real-Time	Turn-based	Grand Strategy	4X	Tower def.	Auto-battler	iné
StarCraft	✓	×	×	✓	×	×	×
Into the Breach	×	✓	×	×	×	×	×
XCOM	×	✓	×	×	×	×	×
Factorio	✓	×	×	×	×	×	✓
Dune	×	✓	×	×	×	×	✓
Hearthstone	×	✓	×	×	×	×	✓
Frostpunk	✓	×	×	×	×	×	✓
Total War	✓	✓	✓	✓	×	×	×

Tabuľka 1.1: Tabuľka vybraných hier a ich umiestnenie v žánri

Kapitola 2

Princípy tvorby strategických hier

2.1 Princípy vývoja softvéru používané pri tvorbe strategických hier

S populárnymi metódami tvorby videohier, s intuitívnym rozhraním a relatívne nenáročným použitím, ako je Unity (ale aj kopec ďalších), je možné začať si vytvárať vlastnú videohru pre prakticky kohokoľvek, bez ohľadu na predošlé vzdelanie. Toto, podľa mňa, vytvára veľmi mylný obraz o procese vývoja hier. Vidohry rozhodne môžu byť umením, avšak proces vytvárania hier, je stále proces vývoja softvéru, ktorý sa drasticky líši od bežného vývoja umenia.

V skutku, toto nie je nevyhnutne len môj názor, nakoľko aj *Erik Bethke* cíti potrebu tento fakt zdôrazniť vo svojej knihe *Game Development and Production* [2], slovami “game development is software development”.

2.1.1 Životný cyklus vývoja softvéru

S touto myšlienkou na vývoj hier teda vieme aplikovať východiská princípov vývoja softvéru, základný model životného cyklu vývoja softvéru (Software development life cycle - SDLC) navrhuje zahrnutie niekoľkých fáz [16], z nich môžeme odvodiť kľúčové činnosti, ktoré sprevádzajú tento cyklus:

- plánovanie a definovanie požiadaviek
- navrhnutie dizajnu
- implementácia softvéru
- testovanie a spustenie produktu
- údržba a podpora

Počas plánovania sa vyhodnocujú podmienky projektu, finančné nároky, vytvorenie časového harmonogramu a organizácie tímu/tímov pracujúcich na projekte. Definovanie požiadaviek je považované za súčasť plánovania, mali by definovať ako má finálny produkt fungovať, čo by mal byť výsledok jej práce, a čo táto funkcia bude požadovať. Súčasťou požiadaviek sú tiež zdroje potrebné pre jeho vybudovanie.

Dizajn modeluje spôsob akým má aplikácia fungovať. Súčasťou dizajnovania môže byť prototypovanie, prototyp je zjednodušená verzia softvéru, ktorá je časovo nenáročná na vybudovanie. Prototyp demonštruje základnú myšlienku toho, ako má výsledný softvér vyzerať a fungovať. Prototyp sa využíva hlavne na rýchle získanie spätnej väzby.

Implementácia je samotný proces programovania softvéru.

Programovanie je však málokedy bezchybné, preto je dôležité medzi implementáciou a publikovaním/spustením predaja, zahrnúť fázu testovania, kde sa overí či softvér naozaj funguje tak, ako je požadované, nájdu sa prípadné chyby a opravujú sa. Následne sa produkt sprístupní pre cieľných užívateľov.

Metodológie vývoja softvéru

Teraz sa pozrieme na metodológie SDLC, ktoré popisujú spôsob akým sa dá pristupovať k vývoju softvéru. V dnešnej dobe, väčšina vývojárov používa jednu z dvoch tried metodológií: Tradičný development a Agile development [18]

Tradičný development

Tradičné metodológie, ako vodopádový prístup, sú založené na sekvenčnom vykonávaní fáz SDLC, kvôli tomuto, tradičné metodológie vývoja závisia na stanovení stabilných požiadaviek na začiatku procesu, ktoré sa nebudú meniť počas vývoja.

Pre tradičné metodológie vývoja sú charakteristické štyri fázy[18].

Vo fáze definovania požiadaviek je potrebné definovať požiadavky a vytvoriť časový odhad na dĺžku trvania ostatných jednotlivých fáz. Tento časový odhad by mal zahŕňať predpoklad akýchkoľvek problémov, ktoré sa počas práce môžu objaviť a spomaliť vývoj.

Nasleduje fáza dizajnovania a plánovania architektúry, kde sa vyprodukuje technická infraštruktúra v podobe diagramov a modelov. Tieto by mali navyše pomôcť odhaliť potenciálne problémy, ktoré sa môžu počas vývoja vyskytnúť, a poskytnú pracovnú roadmapu, na ktorú sa môžu vývojári odkazovať, v prípade potreby, počas implementácie.

Ďalšia fáza, je zameraná na produkciu kódu, vývoj sa často rozdelí na menšie časti rozdelené medzi tími.

Fáza testovania sa do istej miery prelína s fázou vývoja kódu, pre zaistenie, že sú chyby odhalené čo najskôr a zníži sa tak riziko, že oneskoria projekt.

Metódy tradičného prístupu sú založené na pevne definovaných požiadavkách a vyvíjajúcej sa dokumentácií, ktorá sa píše súčasne s vyvíjaným kódom. Zmeny požiadaviek alebo dizajnu počas fázy vývoja sú veľmi problematické, ich implementácia stojí veľa času a zahodeného vývoja.

Agile development

Agile development je založený na myšlienke incrementálneho (iteratívneho) vývoja, kde sú vývojové fázy prechádzané opakovane počas vývojového cyklu. V agile, namiesto toho, aby sa k vývoju pristupovalo ako k jednému veľkému cyklu, sa k vývoju pristupuje po menších častiach, nazvaných “inkrementy” alebo “iterácie”. Počas každej iterácie vývoj prejde všetky tradičné fázy vývoja, následne sa zbiera feedback z prebehnutého testovania alebo od samotných zákazníkov a na základe tohto feedbacku sa vytvárajú požiadavky a dizajn nasledujúcej iterácie.

Princípy agile prístupu k vývoju softvéra je založený na agile manifeste, dokument od tvorcov agile prístupu, popisujúci základné hodnoty a princípy, ktorým verili, že by mali byť používané pri vývoji softvéra [10] agilemanifesto.org/.

Agile taktiež využíva koncept prototypovania, prototyp je reprezentácia niektorého aspektu, alebo celého konceptu, dizajnu softvéru [5].

Prototyp nie je plne funkčný softvér, pri vývoji prototypu sa robia vážne kompromisy na funkčnosti, aby bol veľmi rýchlo implementovateľný. Hlavným zmyslom prototypu, je rýchlo získať feedback.

2.1.2 Životný cyklus vývoja hier

Ako som spomínal skôr, princípy vývoja videohier vychádzajú z princípov vývoja softvéru, avšak videohry sú špecifický typ softvéru, ktorý obsahuje umenie, audio a gameplay [2].

Podobne, ako pri klasickom SDLC, vieme uplatniť tradičnú aj agile metodológiu k vývoju videohier, väčšina modelov životného cyklu vývoja hier (GDLC) však vychádza z princípov iteratívneho prístupu [20].

Jeannie Novak popisuje GDLC ako proces s 3-stupňovými iteráciami dizajnovania, prototypovania a následného vyhodnotenia [19]. Postup veľmi podobný agile metodoló-

gií, avšak po každej iterácii sa vytvorí prototyp, ktorý sa otestuje, zozbiera sa feedback a následne sa podľa výsledkov upraví dizajn do nasledujúcej iterácie.

Radaman a Widyani popisujú niekoľko ďalších metód prístupu k GDLC od rôznych spoločností, ktoré sa zaoberajú vývojom hier [20]. Je veľa rôznych spôsobov, ako pristúpiť k vývoju hier. Napriek tomu, že väčšina zdrojov odporúča iteratívny prístup, je možné použiť aj vodopádový prístup, pokiaľ vieme naplniť predpoklady stabilných požiadaviek a nemenného dizajnu.

Radaman a Widyani rozoberajú vo svojej práci aj spôsoby, akým zvoliť metodológiu pre konkrétny problém, to je ale téma, ktorá presahuje rozsah našej práce.

Fázy vývoja videohier

V procese vývoja videohry je niekoľko významných míľnikov, v závislosti od aktuálneho stavu produktu. Podobne ako pri rôznych metódach vývoja, ani tieto míľniky nie sú univerzálne pre všetky projekty, nie každá metóda sa musí zameriavať na každý z týchto míľnikov.

Prototyp

Pri tvorbe videohier sa často využíva prototypovanie, je to trochu špecifická “fáza” vývoja, pretože prototyp môže byť vytvorený z akejkoľvek časti návrhu hry, jednotlivé herné mechaniky alebo na demonštráciu funkčnosti technológie, plánovanej na použitie vo finálnom produkte [19].

Hlavným účelom prototypu je redukovať riziko, že bude vyplývaný čas na implementáciu niečoho nefunkčného, alebo nežiadúceho [17]. Pretože prototypovanie je v princípe rýchly proces (oveľa rýchlejší ako implementovanie plného softvéru), umožňuje nám odhaliť chyby v dizajne a stanoviť nový smer, bez plýtvania viacej času, ako je absolútne nevyhnutné.

Koncept

Konceptová fáza vývoju začína formovaním myšlienky o hre a končí sa rozhodnutím o začatí plánovania projektu (predprodukcii) [19]. Cieľom tejto fázy je stanoviť si ciele, ktoré má hra naplniť, identifikovať cieľovú skupinu na trhu, posúdiť zdroje spoločnosti pre tento projekt, identifikovať koncept, s ktorým budú súhlasiť členovia vývojového tímu a bude mať potenciálne miesto na trhu. Písaný dokument s týmito ideami sa nazýva konceptový dokument.

Predprodukcia

Fáza plánovania, v ktorej sa uskutočnia niektoré z najdôležitejších dizajnových rozhodnutí. Kľúčové herné mechaniky sa často stanovujú v tejto fáze, validácia týchto mechaník môže prebehnúť napríklad prototypovaním. Stanovujú sa tu návrhy na rozsah hry, umelecký štýl grafickej stránky hry, technické požiadavky aj časový rozvrh produkcie [17].

Na konci tejto fázy by mali byť zhotovené Game Design Document (GDD), prípadne aj Technical Design Document (TDD).

GDD slúži ako príručka k procesu vývoja videohry, popisuje gameplay, príbehovú líniu, rozhranie a pravidlá hry. [19]

TDD je založený na GDD a zaoberá sa technickými špecifikáciami hry. Popisuje špecifikácie herného engine, na ktorom je hra vybudovaná, prípadne ich porovnáva k iným populárnym enginom. Popisuje víziu prechodu od konceptu k softvéru, ktorý člen vývojvého tímu bude mať aké úlohy, časový odhad na dokončenie jednotlivých úloh, aké nástroje budú použité na vytvorenie hry a aké softvéry musia byť na tento účel nakúpené [19].

Produkcia

Fáza, v ktorej sa vytvára produkčný kód, umelecké asety, všetky iné potrebné druhy obsahu, v súlade so schváleným dizajnom z predchádzajúcich fáz. Výsledkom tejto fázy je hotová aplikácia, pripravená na vstup na trh.

Nakoľko táto fáza môže byť pomerne rozsiahla, vieme tu rozlišovať niekoľko menších krokov, pre lepšiu orientáciu v procese.

Pre-Alpha

V závislosti od rozsahu alebo spôsobu manažovania projektu, môže byť stanovených niekoľko menších míľnikov, ktoré dajú vývojárskemu tímu smer a istý cieľ, ktorý je ľahšie dosiahnuteľný ako Alpha, alebo neskoršie štádiá projektu [17].

Alpha

V štádiu Alpha je naša aplikácia hrateľná od začiatku po koniec. Grafika nemusí byť finálna, v obsahu môžu byť skoky alebo medzery, ale základný koncept hry je ukončený [19].

Táto fáza sa sústreďuje na ladenie výslednej hry, rozsiahle playtesty sa často odohrávajú práve na Alpha verzií hry, ktorých účelom je odchytať nedoladené časti príbehy, chyby v kóde, tieto nájdené chyby môžu byť zaradené do bug databázy a následne budú

programátori pracovať na oprave týchto chýb.

Beta

V tomto štádiu sú už všetky asety integrované do výslednej hry, úlohou tejto fázy je izolovať a eliminovať všetky programové chyby, problémy s výkonom, grafické asety, ktoré potrebujú úpravy alebo problémy s kompatibilitou na podporovaných platformách [19].

Za týmto účelom sú spustené ešte rozsiahlejšie Beta testy, ktoré už nie sú zamerané na gameplay, herné mechaniky ani príbeh, nakoľko tieto časti sú už v Beta verzií považované za finálne. Miesto toho, sú tieto testy zamerané špeciálne na hľadanie vyššie uvedených problémov.

Gold

Keď boli adresované všetky známe problémy s hrou, jej výkonom a kompatibilitou, výsledná verzia sa nazýva zlatý kandidát - kandidát, ktorý môže byť schválený vydavateľom a následne odoslaný na distribúciu. Počas tejto fázy sa už nevykonávajú žiadne zmeny na výslednom produkte, s výnimkou najzávažnejších chýb, ktoré neboli odhalené skôr [19].

Status "zlatý", pochádza z praktiky vytvárania Gold Master (GM) verzie, ktorá bola poslaná na hromadnú duplikáciu fyzických produktov.

Dnes je však oveľa populárnejšia možnosť digitálneho prístupu, preto býva viac a viac časté, že sa vytváranie fyzických kópií hry úplne vynechá [17].

Postprodukcia

Po vydaní hry sa práca stále nekončí, v postprodukčnej fáze si produkčný tím všíma odozvu na ich hru, potenciálne z celého sveta. Míľniková štruktúra vývoja môže byť vymenená za vydávanie menších updatov (patch), ktorých cieľom je nahradiť alebo vylepšiť menšie časti pôvodnej hry. Hoci zmeny rozsiahlejších častí hry sú zriedkavé, v dnešnej dobe väčšina hier po vydaní sleduje, ako svet obavuje nové, neodchytené, chyby kódu alebo špecifické chyby na niektorých typoch zariadení a pomocou týchto patchov sa ich snažia rýchlo adresovať [19].

Alternatívne, môže tím pokračovať v míľnikovej štruktúre vývoja s novými míľnikmi založenými okolo DLC (Downloadable Content), alebo iných väčších rozšírení pôvodnej hry [17].

Tím vývoja hry

So všetkou teóriou ako pristupovať k tvorbe hier, ešte stále potrebujeme jednotlivých ľudí, ktorí budú vytvárať jednotlivé úlohy pre vytvorenie produktu. Títo ľudia budú dokopy vytvárať vývojársky tím. Ľudia v tomto tíme môžu zastávať určité pozície, v

závislosti od ich náplne práce. Niektorí ľudia môžu zastávať aj viac pozície v jednom tíme naraz, a rôzne tími môžu mať rôzne pozície upravené podľa vlastných potrieb, prípadne nemusia niektoré úlohy využívať vôbec.

Produkcia

Producent, alebo producenti, sú zodpovední za dohliadanie na to, aby bol projekt dokončený v rámci časových a finančných očakávaní a požiadaviek na kvalitu, ako aj na to, aby každý člen tímu robil prácu ktorú má [19].

V prípade viacerých producentov na projekte, je zvyčajne jeden z nich vymenovaný za exekutívneho producenta, jeho úlohy následne zahŕňajú manažovanie produkcie, prototypovania a podpory projektu.

Ostatní producenti následne spravujú dodržovanie časových očakávaní, kontraktové požiadavky, vyplácanie členov tímu, správa rozpočtu a plánovanie.

Dizajn

Úlohou herných dizajnérov je prispievať k dizajnu funkčných systémov (herný svet, rozhrania interakcie) a riešiť problémy v týchto oblastiach dizajnu. Dizajnéri sa sústreďujú na gameplay, dizajn levelov, príbehovú líniu alebo rozhrania. Niektorí dizajnéri majú skúsenosti a znalosti aj z iných oblastí, umenie, programovanie a môžu sa podieľať aj na týchto úlohách [19].

Ak sa na projekte podieľa viac ako 1 dizajnér, môžu si rozdeliť oblasti pôsobenia v rámci toho projektu. Týmto spôsobom môžeme rozlišovať viaceré dizajnérske pozície.

Kreatívny riaditeľ

Kreatívny riaditeľ zabezpečuje, že grafický štýl a koncept výslednej hry sa zhodujú s pôvodnou víziou.

Režisér dizajnu

Režisér dizajnu rieši konflikty v tíme, vedie tím pri dizajnovaní prototypov a zabezpečuje produkciu funkčnej a kompletnej dokumentácie.

Vedúci dizajnér

Vedúci dizajnér dohliada na dizajnový tím a podieľa sa na každodenných dizajnových procesoch, zodpovednosť vedúceho dizajnéra môže taktiež zahŕňať vývoj gamep-

layu a dokumentácie a dizajn levelov.

Naratívny dizajnér

Naratívny dizajnéri zodpovedajú za príbeh hry, naratívne východiská, ktoré vedú hráča, alebo hráčov, herným svetom. Zodpovedajú aj za vedľajšie príbehy, nadväznosť a súdržnosť príbehu mimo samotnú hru, v prípade hier, ktoré predstavujú pokračovanie už existujúceho príbehu (napríklad z predchádzajúceho vydania rovnakého titulu, ale pri herných adaptáciách z iných médií).

Grafika

Herní grafici majú za úlohu vytvárať koncepty umenia, a asety ktorú budú použité v hre. Rozlišujeme 4 základné typy úloh, ktoré rozdelujú typy grafikov, kreslenie, modelovanie, textúrovanie a animovanie. Okrem zamerania podľa typu úloh sú aj rôzne využitia na ktoré sa môžu grafici sústrediť, ako postavy, interiérové priestory, exteriérové priestory, vizuálne efekty, cinematické zábery, alebo mnoho ďalších [19].

Vedúci grafik

Vo väčšine projektov je zvolený jeden grafik, ktorý je považovaný za vedúceho, jeho úlohou je dohliadať na dianie v rámci grafického oddelenia projektu a zúčastňuje sa na denných procesoch tvorby grafiky. Vedúci grafik komunikuje s kreatívnym riaditeľom a zabezpečuje zladenie medzi konceptov hry a grafikou vytvoreného produktu. Vedúci grafik je väčšinou zvolený pre špecifické zameranie, napríklad vedúci animátor, alebo vedúci kopecný grafik a podobne.

Koncepčný grafik

Koncepčný grafik vytvára kresby a náčrty herného prostredia, rekvizít alebo postáv. Tieto koncepty sú často súčasťou koncepčného dokumentu a používajú sa na reprezentáciu myšlienky projektu, v čase keď ešte skutočné herné prostredia alebo modely postáv neexistujú.

Modelový grafik

Úlohy modelových grafikov zahŕňajú tvorbu 3D assetov z 2D náčrtov, často je potrebné vytvoriť celé modely od základu, alebo používanie techník na namapovanie skutočných rekvizít do virtuálneho prostredia. Modeloví grafici vytvárajú všetky grafické asety ktoré budú použité vo výslednom produkte.

Textúrový grafik

V procese textúrovania je potrebné aplikovať 2D textúry na 3D modely. Tieto textúry

niekedy treba vytvoriť od základu, alebo pomocou rôznych techník, napríklad fotografiu z reálneho sveta nahráť do virtuálneho prostredia a následne ho upraviť.

Animátor

Animovanie zahŕňa aplikovanie pohybu na objekty a postavy v hernom svete. Okrem toho, animátory sú často súčasťou tvorby cinematických scén.

Programovanie

Úlohy programátorov môžu zahŕňať množstvo procesov, od vytvárania herného enginu, alebo nástrojov na tvorbu herného sveta na použitie pre ostatných členov tímu, až po spravovanie databáz, integrovanie grafiky a audia.

Technický riaditeľ

Technický riaditeľ vytvára technický dizajn projektu, vyberá nástroje a hardware ktoré budú používané a dohliada na chod implementácie počas celého projektu.

Vedúci programátor

Vedúci programátor dohliada na tím programátor a komunikuje s technickým riaditeľom, väčšinou je súčasťou každodennom procese programovania.

Sieťový programátor

Sieťový programátor zabezpečuje funkčnosť a bezpečnosť multiplayeru, alebo módu viacerých hráčov.

Programátor grafiky

Programátor grafiky (väčšinou grafik aj programátor súčasne) poskytuje programátorské riešenia na grafické problémy hry.

Programátor nástrojov

Programátor nástrojov spolupracuje s tímami grafikov a dizajnérov, a inkorporuje ich prácu do kódu hry. Niekedy je potrebné vytvoriť editory pomocou ktorých sa dajú dizajnovať levely, alebo podobné nástroje, ktoré môžu byť použité členmi mimo programátorskej sekcie.

Programátor umelej inteligencie

Programátor umelej inteligencie (AI) sa sústreďuje na implementáciu správania, ktoré pôsobí "inteligentne". Teto správania sú zväčša aplikované na nehrateľné-postavy (NPC) alebo na "virtuálnych hráčov" (s ktorými hráč nevie interagovať priamo, ako oponent v šachu).

Programátor audia

Programátor audia implementuje zvuky a hudbu do hry, navyše má na zodpovednosti efektívne pristupovanie k zvukovej karte a zabezpečiť správne načítavanie rôznych audio formátov.

Audio

Vývoj videohry zahŕňa vytváranie potrebného audia, toto môže zahŕňať zvukové efekty, hudbu alebo reč [19].

Zvukový režisér

Zvukový režisér spravuje zvukové oddelenie, komunikuje s programátorom audia, aby zabezpečil, že všetky zvukové asety sú správne inkorporované do výslednej hry. Okrem toho spravuje výber zamestnaného zvukového personálu a licencie pre použitie už existujúcej hudby v hre.

Zvukový dizajnér

Zvukový dizajnér je zodpovedný za tvorbu zvukových efektov a okolitých zvukov v hre. Tvorba takýchto audio súborov niekedy vyžaduje, aby zvukový dizajnér nazbieral zvukové nahrávky zo skutočného sveta, podľa potreby ich upravil a poskytol zvukovému programátorovi, inokedy však stačí takéto súbory získať z už existujúcich audio knižníc.

Hlasový umelec

Hlasový umelec poskytuje zvukové nahrávky hovoreného slova pre naratívu alebo konverzáciu.

Testovanie a záruka kvality

Herné testovanie zahŕňa hranie videohry skôr ako je verejne dostupná za účelom zistenia či je "hrateľná" - bez závažných programovacích chýb, dôsledná a zábavná.

Narozdiel od testovania, záruka kvality (QA) zahŕňa založenie štandardov a procesov pre vývoj hry. Funkcie QA zahŕňajú monitorovanie procesov, vyhodnocovanie produktu a auditovanie. Zmyslom týchto štandardov a procesov, je zabezpečenie že vývoj produktu napĺňa stanovené štandardy dokumentácie, dizajnu, programovania a kódu [19].

Manažér testovania

2.1. PRINCÍPY VÝVOJA SOFTVÉRU POUŽÍVANÉ PRI TVORBE STRATEGICKÝCH HIER3

Manažér testovania spravuje fungovanie procesov testovania aj QA, môže taktiež spravovať rozpočet na testovanie, alebo komunikovať s vyšším manažmentom pre zabezpečenie, že testovaný produkt bude dokončený bez vážnych zdržaní.

Vedúci tester

Vedúci tester dohliada na tím testerov a často sa podieľa na každodenných testovacích procesoch. Medzi úlohy vedúceho testera patrí hľadanie chýb a nekonzistencií v 3D geometriách, v modeloch, textúrach, estetike alebo hernej logike. Vedúci tester taktiež spravuje dáta playability testerov a zabezpečuje, že hľadajú chyby v správnych častiach hry. Navyše môže komunikovať s vyšším manažmentom projektu, kde sa rozhoduje o tom či sa na nájdené chyby oplatí investovať čas a prostriedky pre opravu.

Compatibility a Format tester

Compatibility(pre počítačové hry) a Format(pre konzoly) tester testujú či má hra funkčnú medzi-platformovú kompatibilitu. Ak je hra podporovaná všetkými konzolami, overujú či na každom type konzoly beží rovnako dobre, z hľadiska výkonu. Ďalej kontrolujú či hra funguje správne s rôznym hardvérom, testujú sa odlišné vstupné zariadenia. Pre počítačové hry sa testuje či hra beží adekvátne na zariadení, ktoré tesne spĺňa minimálne technické požiadavky.

Production, QA a Regression tester

Production, QA a Regression tester navrhujú možné zlepšenia, prídavky alebo označujú časti na odstránenie. V súlade s týmto, môžu taktiež prirovnávať testovaný produkt k podobným produktom na trhu, ktoré predstavujú konkurenciu. Regression testery sa obzvlášť zameriavajú na overovanie, či odstránené chyby boli naozaj úplne a korektne odstránené odstránené.

Playability, Usability a Beta tester

Tester z tejto skupiny sú väčšinou súčasťou cieľovej trhovej skupiny, na ktorú je hra zameraná a testujú z pravidla Beta verziu hry. Beta testovanie môže spočívať vo forme stres-testovania alebo progression-testovania, ktoré zahŕňa dôkladné prejdienie hry od začiatku po koniec za účelom určenia či je táto hra hrateľná a použiteľná.

2.2 Prvky dizajnu strategických hier

Je pomerne náročné zovšeobecňovať dizajn strategických hier, ale je fajn mať isté vodítka počas fázy tvorenia konceptu. Povedali sme niečo o vlastnostiach strategických hier, ale “dôležitosť rozhodnutí” sa nedá priamočiaro implementovať. Ideálne by bolo popísať tieto prvky pomocou herných mechaník, pretože tie sa dajú priamo preložiť na implementáciu, toto však nie je realistické, pretože herné mechaniky sú veľmi špecifické pre konkrétnu implementáciu.

Andrew Rollings a Ernest Adams vo svojej knihe *Andrew Rollings and Ernest Adams on game design* navrhujú, že každá strategická hra zdieľa aspekt manažovania [22]. Pretože to je jeden z najľahšie identifikovateľných rozdielov od ostatných žánrov, v strategických hrách, ako sme spomínali v kapitole 1, dôležitosť hráča nepochádza z postavy, ktorú hráč ovláda, namiesto toho hráč ovláda väčšie konštrukty, na úrovni skupín či impérií, alebo niečo na podobnej škále pre hry ktoré nie sú zamerané na združenia ľudských bytostí. Rollings a Adams v tej istej knihe identifikujú 3 hlavné faktory dizajnu strategických hier: téma, prezentácia a perspektíva.

Téma

Témy strategických hier sú často odvodené z niektorej/niektorých aktivít: dobývanie, objavovanie a obchodovanie. Toto je užitočné, nakoľko tieto témy sa pomerne jednoducho prevádzajú na implementáciu.

Dobývanie je nevyhnutnou súčasťou vojnových hier, alebo 4X hier v časti “exterminate”. Dobývanie vyžaduje jedného alebo viacerých súperov, ktorý predstavujú cieľ dobývania a vedľa sa bránia s nejakou úrovňou výzvy, potrebuje prevedenie bitiek, ktoré otestuje hráčové taktické schopnosti, bitky zväčša prebiehajú formou ovládania množstva menších jednotiek, tieto vyžadujú nejakú formu manažovania jednotiek, toto manažovanie však nesmie byť natoľko zložitá, až začne uberať zo strategického sústredenia. Ak sa na manažovanie jednotiek pozrieme ako na spektrum, na úplnom konci toho spektra sú RPG alebo akčné hry, kde hráč ovláda iba jednu jednotku, ktorá je mocná a potenciálne enormne komplexná. Hry sústredené na ovládanie menšieho počtu komplexnejších jednotiek má tendenciu sústrediť sa viac na taktickú stránku hry, na úkor tej strategickej [22]. Ďalšia, hoci nie nevyhnutne povinná, no predsa populárna, je inklúzia ekonomiky a budovania základne a armády, ako istej prípravy pred začatím samotného dobývania. Tieto prvky navyše vyžadujú nejakú formu hernej ekonomiky, získavania prostriedkov pomocou ktorých hráč môže budovať svoju základňu a armádu.

Objavovanie je veľmi často využité ako podporná mechanika, v StarCrafte nie je hlavným cieľom objavovať, ale stále potrebujeme zistiť kde sa náš nepriateľ nachádza, kde sa nachádzajú suroviny, a podobne. Objavovanie vyžaduje rozmerný herný svet a mechaniky, ktoré podporujú a povzbudzujú objavovanie. Nemá zmysel preskúmať svet ktorý nič neskrýva, takže forma zatajovania informácií pred hráčom, populárna voľba je “fog of war”, pri ktorej hráč vidí len to, čo sa nachádza v okolí jeho jednotiek. Samotný spôsob objavovania môže byť sprostredkovaný hernou mechanikou, či už hráč posielá svoje jednotky, alebo iným spôsobom získava informácie. Samozrejme objavovanie samotné nemusí stačiť, niekedy je na mieste systém odmien a rizík, ktoré sprevádzajú objavovanie a podnecujú stratégiu a rozhodovanie sa medzi možnosťami.

Obchodovanie môže slúžiť ako hlavná mechanika pri niektorých hrách, avšak ako sme už spomínali pri dobývaní, že ak chceme mať hernú ekonomiku a budeovanie armády, potrebujeme aspoň jednoduchú formu obchodovania (nákup samotných jednotiek alebo vylepšenia základne) [22]. Tieto mechaniky môžu byť prepracované podľa potreby, aby sa viac priblížili skutočnosti, alebo zvýšili náročnosť, môže byť implementovaná funkcia distribúcie prostriedkov, kde nestačí prostriedky len získať a začať ich používať, ale od miesta kde sa získali k miestu kde sa chcú použiť, musí existovať spojenie. Táto náročnosť však musí byť vhodne zvolená a optimalizovaná, napríklad pri rýchlej vojnovéj RTS ako StarCraft, hráč nemá dost času na správu distribúcie prostriedkov. Naopak pri pomalšej TBS, ako Civilization, má hráč potenciálne neobmedzene veľa času aby si vedel analyzovať a naplánovať distribúciu surovín aj počas toho ako spravuje ostatné akcie.

Rôzne hry môžu implementovať rôzny z tém, a samozrejme, kreatívny dizajnéri môžu vymyslieť aj úplne nové témy pre svoju hru. Pomer medzi týmito témami má veľký vplyv na dojem z hrania, pre predstavivosť si môžeme predstaviť rozdiel medzi typickou hrou štýlu monopoly, ktorá nemá prakticky žiadne objavovanie a je sústredená na obchodovanie, a porovnať si to s klasickou vojnovou rts, na štýl StarCraftu, ktorá je takmer plne sústredená na bojové mechaniky a koncept dobývania, objavovanie a obchodovanie sú použité len ako podporné témy, ktoré hráčovi umožnia nájsť ciele pre dobývanie, alebo nákup jednotiek a zbraní [22].

Prezentácia

Strategické hry často fungujú na veľmi zložitých mechanikách, kvôli tomu je dizajn rozhrania, ktorým hráč interaguje s hrou, veľmi dôležitý, dobrý dizajn tohoto rozhrania môže ztrivializovať náročnosť pochopenia týchto mechaník, naopak zlý dizajn to môže ešte viac skomplikovať [22].

Perspektíva

Po dlhú dobu sa v strategických hrách z technických príčin využívala exkluzívne perspektíva 2D pohľadu z hora, alebo “vtáčieho pohľadu”. Takáto perspektíva je ale aj dnes, s menšími technologickými obmedzeniami, veľmi praktický, nakoľko umožňuje pre relatívne intuitívne ovládanie všetkých prvkov, ktoré má hráč pod kontrolou. Koniec koncov, aj v skutočnom svete je generál skrytý od bojiska, a pracuje na 2D mape [22].

Kapitola 3

Prostriedky vývoja strategických hier

V tejto kapitole popisujeme softvéry, ktoré sa často spájajú s vývojom strategických hier v Unity.

3.1 Unity

<https://unity.com/>

Unity je cross-platformový game engine vyvíjaný spoločnosťou Unity Technologies v Dánsku. Unity pozostáva z vlastného renderovacieho enginu, nVidia PhysX enginu a Mono. Mono je open-sourcová implementácia MicroSoftových .NET knižníc. Unity navyše ponúka integrovaný editor, pre intuitívny vývoj herného sveta, tvorbu animácií a mnoho ďalších možností. Ponúka základné rozhranie pre skriptovacie jazyky, primárne C#. [6]

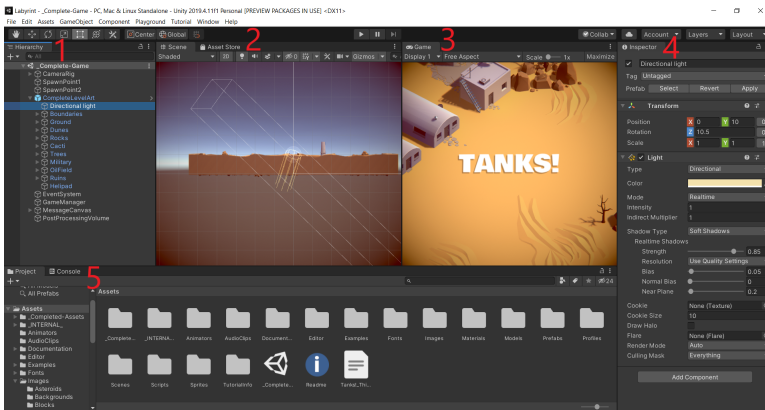
3.1.1 Dokumentácia

Unity poskytuje rozsiahlu oficiálnu dokumentáciu, ktorá výrazne zjednodušuje proces učenia sa práce s Unity. Túto dokumentáciu možno nájsť na adrese <https://docs.unity3d.com/Manual/UnityManual.html>, oficiálna dokumentácia je pravidelne aktualizovaná pre verzie Unity s dlhodobou podporou (LTS), ale stále sú dostupné aj staršie verzie dokumentácie, dokonca, keď sa používateľ snaží prepnúť verziu, ostatné verzie sú roztriedené podľa toho či obsahujú aktuálne otvorenú stránku, čo môže byť užitočné pri hľadaní spätnej kompatibility konkrétnych funkcií so staršími verziami.

3.1.2 Editor

Unity editor má vysokú úroveň prispôbitelnosti pre maximálne pohodlie pri používaní. Editor obsahuje niekoľko rôznych okien s rôznym obsahom, ktorých umiestnenie

si užívateľ môže intuitívne usporiadať, do takmer ľubovoľnej pozície, len pomocou “ťahania” myškou, zmeniť veľkosti jednotlivých okien ťahaním okraju daného okna alebo zlúčiť viaceré okná do jedného, čím budú obe (alebo viaceré) okná reprezentované ako záložky na hornej strane okna, pričom jedna záložka je zvolená a tá reprezentuje prvok, ktorý je v tomto okne viditeľný.



Obr. 3.1: Unity editor

Na obrázku 3.1 môžeme vidieť možné rozmiestnenie prvkov editora, v ktorom sú tieto prvky očíslované.

Unity editor funguje na princípe asetov, pričom asety sú položky, ktoré môžu byť použité v projekte. Tieto asety sú uložené na disku v priečinku projektu, tento priečinok je možné prehliadať priamo v editore (číslo 5 na obrázku 3.1).

Asety je možné pridávať z priečinkov priamo (číslo 2 na obrázku 3.1), kliknutím a potiahnutím myškou. Scéna je v podstate kontajner, ktorý obsahuje prvky. Tieto prvky v scéne sa volajú herné objekty a sú základnou jednotkou renderovania a správania herného sveta. Všetko, čo sa nachádza v scéne, je herný objekt. Povaha konkrétneho herného objektu je určená komponentami, ktoré sú pripojené k danému hernému objektu. Komponenty určujú správanie herného objektu, komponenty pokrývajú širokú množinu funkcií, od grafických komponentov, ktoré spôsobujú, že sa herný objekt, ku ktorému sú pripojené, renderujú a zobrazujú na kamere, tieto môžu mať navyše kolízie, ktorý umožňuje iným objektom a/alebo hráčovi interagovať s týmto objektom, cez objekty s funkcionálnou logikou správania, ktoré reagujú na vzniknuté podnety (ktoré môže vyvolávať iný herný objekt, alebo hráč) predurčeným správaním, až po neviditeľné objekty, ktoré sprostredkujú interakcie medzi inými objektami na scéne.

Všetky herné objekty na scéne sú automaticky uložené do hierarchie danej scény (číslo 1 na obrázku 3.1). Užívateľ si môže herné objekty v hierarchii usporiadať do

skupín, podľa logiky použitia daných objektov. Usporiadanie v hierarchií využíva základné vlastnosti dedenia prvkov od rodičovského objektu na jeho potomkov, napríklad zmena pozície (čiže úprava komponentu transform), sa prejaví na pozícií (komponente transform) jeho potomkov [27].

Používateľ si môže prezerat' objekty na scéne a v hierarchií, vybraný herný objekt sa zobrazí v inšpektore (číslo 4 na obrázku 3.1), kde sa zobrazia všetky užitočné detaily o danom hernom objekte. Samotný herný objekt obsahuje veľmi málo informácií, obsahuje meno, tag (podľa ktorých sa dajú objekty kategorizovať bez ohľadu na usporiadanie v hierarchií a dá sa im priradiť správanie podľa tejto kategorizácie), určiť či sa má zobrazovať v hernom okne (o tomto viac v nasledujúcom odseku), a potom sa zobrazia všetky komponenty pripojené k danému hernému objektu. Škála možných komponentov už bola spomínaná skôr, každý herný objekt však obsahuje komponent transform, ktorý určuje pozíciu, veľkosť a orientáciu príslušného objektu na scéne. V inšpektore môže používateľ manuálne nastavovať a meniť špecifikácie jednotlivých komponentov alebo pridávať nové komponenty. Jeden typ komponentu, ktoré môže používateľ priradiť hernému objektu, sú vlastné skripty, ktoré je možné písať a uložiť ako aset do priečinkového súboru. Tieto skripty môžu objektu dodať logiku správania, často prepojenú na iné komponenty, prípadne komponenty na iných herných objektoch (napríklad dva objekty s komponentom collider môžu detekovať stretnutie, na základe čoho spustia správanie v skripte) alebo môžu len udržiavať dáta (prípadne s nimi manipulovať a meniť ich, podľa podnetov), ku ktorým môžu iné objekty pristupovať. V rámci týchto skriptov je navyše možné pristupovať nie len k herným objektom, ale všetkým jeho komponentom, takže všetko, čo používateľ môže nastaviť manuálne v inšpektore, vie taktiež meniť počas behu hry pomocou skriptov.

Všetky tieto nástroje reprezentujú informácie na pozadí hry, užitočné pre herného vývojára, ale dôležitá stránka hry je, ako ju vidí hráč. Tu prichádza herné okno (číslo 3 na obrázku 3.1), kde používateľ vidí, čo je renderované zo zvolenej kamery v hernej scéne. Inými slovami, toto okno reprezentuje, čo uvidí hráč. V hernom okne však nemusí byť renderované vždy, každý herný objekt má informáciu či je aktuálne aktívny alebo nie. Neaktívne objekty sa nerenderujú na kamere, k nim priradené skripty nereagujú na podnety a ich kolízia je ignorovaná ostatnými objektami.

Okrem označených okien z obrázka 3.1, unity editor obsahuje ešte pár ďalších funkcií. Asset store je verejná knižnica Unity asetov, ktorá má dostupné vlastné okno v editore, v starších verziách bol celý unity store priamo v editore, dnes slúži z technických dôvodov len na presmerovanie na prehliadačovú stránku asset store. Táto knižnica obsahuje množstvá predpripravených asetov, alebo celých balíkov asetov obsahujúcich

grafické objekty, hotové skripty aj s popisom ich funkcie. Tieto asety môžu byť priamo od Unity Technology, ale táto knižnica je dostupná pre používateľov nie len na sťahovanie, ale aj na nahrávanie, takže celá komunita môže ponúkať svoje vybudované asety, alebo celé balíky, v asset store. Niektoré asety sú dostupné zadarmo, alebo za variabilný jednorázový poplatok.

A posledná dôležitá súčasť editora, ktorú si spomenieme, je konzola, ktorá funguje ako klasická výstupná konzola. Príkaz `Debug.log()` vie vypísať vstupnú správu na konzolu, neodchytené výnimky sa vypíšu na konzolu s príslušnými detailami, chyby pri kompilácií sa vypíšu na konzolu.

3.2 CSharp

<https://docs.microsoft.com/en-us/dotnet/csharp/>

C# je multi-paradigmaticý objektovo-orientovaný programovací jazyk, s automatickým garbage collectorom, čo enormne uľahčuje prácu s pamäťou.

C# bol taktiež vyhlásený za medzinárodný štandard spoločnosťou Ecma, v ich štandardoch[4] nájdeme medzi cieľami dizajnu jazyka C#, mimo iné, aj:

účel použitia pri vývoji softvérových komponentov vhodných na nasadenie v distribuovaných prostrediach.

Portabilita

Unity Engine používa C# ako hlavný skriptovací jazyk, pomocou Mono.

3.3 VisualStudio

<https://visualstudio.microsoft.com/>

Microsoft Visual Studio je integrované vývojárske prostredie (IDE) od spoločnosti Microsoft, ktoré obsahuje editor kódu, integrovaný debugger, code profiler, podporuje nástroje na refaktorizáciu kódu a mnohé ďalšie nástroje.

Unity využíva Visual Studio ako predvolený softvér pre písanie a úpravu skriptových asetov.

3.4 Git

<https://git-scm.com/>

Git je voľne dostupný open-source version-control systém, určený na sledovanie a kontrolovanie zmien v rámci softvéra, s podporou “branchovania” (sledovania viacerých

verzií rovnakého softvéru súčasne) a neskoršieho “mergovania” (zjednotenia do jednej konzistentnej verzie) týchto branchov.

GitHub poskytuje webové rozhranie a hosting systému git, taktiež poskytuje bezpečné spojenie s možnosťou synchronizácie lokálneho repozitору Git, takže sa dá používať ako distribučné médium.

Pri našej pilotnej aplikácii používame GitHub ako hostovací servis Gitu.

Kapitola 4

Implementácia pilotnej hry a vyhodnotenie práce

V tejto kapitole popíšeme proces implementácie našej pilotnej hry, jednotlivé vývojové štádia cez ktoré sme prešli, a popis výsledného produktu.

4.1 Vývojové štádia a dizajn hry

Pre praktickú časť sme sa rozhodli vytvoriť edukatívno-strategickú hru, ktorej hlavný kontext je, ovládať krajinu v skutočnom historickom kontexte, a viesť ju v určenom časovom rozmedzí. Mimo rozsah účelov tejto práce, bude naša aplikácia ďalej rozvíjaná v projekte na edukatívny herný softvér zaoberajúci sa Slovenskou republikou za komunistického režimu po konci 2 svetovej vojny, čo nám teda určuje krajinu, historický kontext aj časové rozmedzie. Hlavným účelom tejto hry, je edukatívna zložka a teda oboznámiť hráča s týmto historickým obdobím, oboznámiť ho so spôsobom akým politika, ekonomické plánovanie a sociálny systém tejto doby fungovali, a zabezpečiť toto učenie interaktívnym a zaujímavým spôsobom. Potrebujeme však mať nejakú rovnováhu medzi zaujímavou hrou, a edukatívnou zložkou. Herný koncept tejto hry je spravovanie krajiny, zabezpečovanie obchodu a potreby populácie, a v neposlednom rade, tvorenie rozhodnutí v morálnom a politickom kontexte. Musíme zabezpečiť, aby tieto rozhodnutia neboli bezvýznamné, ak sa udalosti vždy odohrajú podľa historickej osnovy, tieto rozhodnutia môžu z pohľadu hráča stratiť význam, a tým stratíme emočné spojenie, ktoré hráč môže mať voči vytváraniu týchto rozhodnutí. Namiesto toho, chceme aby bol každý hráč schopný ovplyvniť udalosti v tejto virtuálnej histórii. Postupnosť udalostí, to kedy sa dejú, alebo či sa vôbec udejú niektoré udalosti, bude všetko záležať od hráčových rozhodnutí. Až do takej roviny kde hráč môže vytvoriť úplne alternatívne časové línie, vzhľadom na skutočnú históriu.

Toto, pochopiteľne, prichádza do rozporu s edukatívnou hodnotou hry, nemôžeme

predsa hráča naučiť falošnú históriu. Preto potrebujeme vhodné protiopatrenia. V prvom rade, táto línia, vytvorená hráčom, bude neustále porovnávaná so skutočnou históriou, dokonca, niektoré udalosti môžu byť spustené vo fixnom čase zhodujúcim sa s históriou, ktorá ponúkne hráčovi orientačný záchytný bod, voči ktorému môže porovnávať ostatné udalosti v čase. V druhom rade, napriek tomu že hráč vie ovplyvniť usporiadanie udalostí na časovej osi, nemôže zásadne zmeniť štruktúru samotných udalostí, čo ho naučí aspoň základ toho *čo*, alebo *ako*, sa stalo, hoci nie *kedy* sa to stalo. Výsledný edukatívny softvér má byť komplementárny voči inému vzdelávaciemu médiu, ako je školné vzdelanie, jeho cieľom nie je celé ho nahradiť, preto sme prišli k záveru, že je potenciálne akceptovateľné, pokiaľ náš softvér naučí len niektoré aspekty cieľového materiálu. Dôležité tu však bude zaistiť, že náš softvér nebude pôsobiť rušivo na ostatné média, ktoré sa pokúšajú učiť aj aspekty v ktorých náš softvér nie je dostatočný, toto by narušilo jeho komplementárny charakter.

V hre budú vystupovať dva politické smery, ktorými hráč môže ovplyvniť svoj spôsob vedenia krajiny, prostriedkom rozhodnutí a správou vzťahov. Tieto dve strany budú, samozrejme, vo vzájomnom rozpore, a týmto chceme reprezentovať istý morálny konflikt, s ktorým sa hráč bude musieť vysporiadať. Okrem týchto dvoch externých strán, je tu však tretí faktor, ktorý bude reprezentovaný miestnymi obyvateľmi a ich potrebami, spravovať tieto je hráčovou hlavnou úlohou, ale popri tom bude nevyhnutné zvoliť si prístup k problematike východ-západ, a hľadať riešenie pre lokálne potreby obyvateľov v rámci možností ktoré tento prístup umožňuje.

Ďalšie hľadisko, ktoré je pri implementácii dôležité zohľadniť je to, že táto hra v jej finálnej podobe nemá tvoriť samostatný produkt, namiesto toho bude tvoriť iba strategickú vrstvu finálnej hry, ktorá bude spojená s ďalšími, podobne rozsiahlymi, časťami.

Na začiatok je teda potrebná základná herná plocha, ktorá bude obalovať všetky dáta a ostatné štruktúry, a tým umožní istú hierarchiu v rámci vnútorných štruktúr. Je veľmi vhodné mať hernú plochu rozdelenú na menšie časti, regióny, pričom všetky regióny budú obalené v jednom objekte ktorý bude držať informácie ktoré sa týkajú celej krajiny, zatiaľ čo jednotlivé regióny budú držať informácie týkajúce sa len daného regiónu.

V rámci každého regiónu chceme zaviesť úroveň manažovania niekoľkých prvkov. Každý región bude predstavovať produkčnú jednotku, čo znamená že sa v ňom môžu produkovať isté suroviny. To ktoré suroviny sú produkované, bude dané produkčnými štruktúrami, ktoré sú v danom regióne postavené. Zároveň bude mať každý región vlastný počet obyvateľov, ktorí sú kľúčový k produkovaniu surovín. Aby sa produkčná

budova mohla zaviesť do prevádzky, vyžaduje dostatok pracovníkov. Hráč teda musí zabezpečiť rovnováhu medzi množstvom pracovníkov a produkčných štruktúr. Zároveň nechceme implementovať štandardné spôsoby “získavania” novej populácie stavebnými projektami, ktoré môžeme pozorovať v podobných hrách. Namiesto toho, bude množstvo populácie v každom regióne finálne (popríklad sa v štádiách hry môže upravovať, ale to bude tiež fixne viazané na príbehové udalosti a teda mimo klasické spravovanie hráča), a táto populácia sa bude deliť na 2 časti, robotníci a roľníci. Roľníci sú predvolený stav populácie, ktorí pasívne pracujú na poliach a generujú jedlo (ktoré si popíšeme čoskoro), hráč má ale možnosť zmeniť časť populácie roľníkov na robotníkov, čím zníži produkciu jedla, ale umožní mu to produkčné prevádzky, a teda generáciu surovín.

Produkcia jedla bude fungovať globálne, takže každý región nemusí produkovať sám na seba, roľníci zo všetkých regiónov produkujú jedlo pre celú krajinu. Jedlo sa naviac neskladuje, čo je vyrobené navyše sa pokazí a zanikne, takže dôležitá je len rovnováha produkcie/spotreby nie je žiadna zásoba, ktorá umožní dočasnú nerovnováhu tohoto pomeru. Pokiaľ spotreba presahuje produkciu, krajina prejde do “hladujúceho” stavu, a zníži sa globálne množstvo produkcie surovín v krajine.

Veľkou témou tohto historického obdobia bola automatizácia, a toto bude práve koncept, ktorým môžeme vyrovnať rovnováhu medzi konverziou populácie na robotníkov a stále zachovať dostatočnú produkciu jedla. Ako hra bude pokračovať, hráč bude mať prístup k modernejším technológiám ktoré zefektívňujú produkciu jedla, a teda stačí menej roľníkov na uspokojenie spotreby, čo hráčovi umožňuje získať viac robotníkov.

Na nasledujúcom kroku sme sa zasekli na relatívne dlhú dobu, pretože sme podcenili ako náročné bude nájsť vhodnú rovnováhu medzi dobrou hernou mechanikou, a historickou presnosťou. Potrebovali sme navrhnúť mechaniku v rámci hernej ekonomiky, ktorá by imitovala systém 5-ročnicového hospodárskeho plánovania v socialistických krajinách. Prešli sme viacerými iteráciami rôznych verzií tejto mechaniky, ale vždy sme mali pocit že by to mohlo byť lepšie. Nakoniec sme však, myslím si, našli pomerne vhodný spôsob, ktorý zároveň zachováva historickú presnosť. Hráč si určí plán produkcie, ktoré musí naplniť za najbližších 5 ťahov, v týchto plánoch necháme hráčovi prakticky neobmedzenú voľnosť, preto je namieste pričleniť mechaniku trestania plánov ktoré sú nastavené príliš vysoko (aby zaručene pokryli akúkoľvek potrebu obyvateľstva), alebo príliš nízko (aby sa hráč vyhol zlyhaniu naplnenia týchto plánov), ktoré by mali hráča odradiť od takýchto taktík.

Potrebujeme systém požiadaviek z rôznych častí hry, ktoré sú hráčovi dostupné keď nastavuje tieto plány. Tu môžu byť reprezentované potreby obyvateľstva, požiadavky v rámci medzinárodného obchodovania, a podobne. Následne potrebujeme systém vyhod-

notenia, ktorý trestá ak nastavenie plánu nenapĺňa požiadavky, alebo ak množstvo skutočnej produkcie nenapĺňa požiadavky, bez ohľadu na samotný plán. Následne trestá ak plán nebol naplnený, alebo ak nejaká produkcia presiahla plán, zatiaľčo niektorá iná produkcia nebola naplnená.

Asi najväčší problém v dosahovaní daného cieľa, je fakt, že požiadavky sa menia, aj počas obdobia kedy sa napĺňa nastavený plán, ktorý sa už zmeniť nedá. Výsledným efektom je to, že je takmer nemožné nastaviť plán, ktorý bude úspešný počas celej doby kde sa ním hráč musí riadiť. Závažnosť trestov, aj ostatné aspekty hry, by mali byť postavené na predpoklade, že hráč v tomto efektívne zlyhá, ide len o to snažiť sa zlyhať čo najmenej. A to je historicky veľmi presná reprezentácia toho ako 5 ročné plány v skutočnosti fungovali.

Z hľadiska nášho edukatívneho cieľa chceme hráčovi predstaviť viac historických faktov, než je možné pomocou herných prvkov. Pre tento účel využijeme koncept “gamepedia”, čo bude vlastná vrstva hry, do ktorej sa hráč vie prepnúť, na štýl ako keď otvoríte herné menu. Gamepedia je zbierka rôznych relevantných náučných textov, herných návodov, nápovied a tipov, na štýl encyklopédie, ale v rámci videohry. V našom prípade ju budeme mať rozdelenú na 3 hlavné časti. Prvá je zameraná na návody a opisy herných mechaník, na túto sekciu sa hráč môže obrátiť keď nerozumie, ako niečo v hre funguje. Druhá časť, je zameraná na hráčov vlastný príbeh, ako som už spomínal, hra sa bude odohrávať v určitom čase v reálnej histórii, ale nechceme len prerozprávať čo sa stalo v histórii, hráč bude svojimi skutkami vytvárať vlastnú časovú líniu udalostí. Tretia časť bude prirovnávať hráčov jedinečný príbeh k skutočnej histórii a popisovať udalosti v historickej správnosti.

Počas celého vývoja sa vyskytlo nespočetne veľa problémov, ktoré začali narúšať chod celého vývoja. Hneď prvý, pomerne závažný, problém nastal keď, krátko po začiatku, začali niektoré časti kódu vyzeráť dosť komplikovane, nachádzalo sa tu mnoho dôležitých premenných a stovky riadkov kódu, pomedzi to však boli aj akosi “izolované” časti kódu ktoré úspešne plnili svoju funkciu a nemali veľa relevancie k ostatným častiam, ako aj novo-vyvíjaným častiam kódu. Nepomohol ani spôsob ukladania samotného kódu v Unity, rozdelený do jednotlivých skriptov, každý vo vlastnom priečinku a obsahujúc najviac jednu verejnú triedu, čím ešte viac znížilo prehľadnosť celkového kódu. Vznikla potreba veľmi rýchlo určiť pravidelné “vlny” refaktorizácie, vyššie spomínané izolované časti kódu ktoré plnili svoju úlohu boli odobraté do vlastného priečinku, pokiaľ sa takto dali logicky oddeliť od zvyšku. Jednotlivé verejné triedy mali jednoduché, jasne pomenované metódy, podľa princípov clean code, aby sa maximalizovala prehľadnosť a minimalizoval sa čas potrebný na pochopenie určitej časti kódu, aj v prípade dlhšieho časového odstupu. Aj keď bolo všetko ľahko prehľadné a viditeľné,

často som sa stretával s problémom “analysis paralysis”, keď som mal implementovať novú funkčnosť do existujúceho kódu, akokoľvek jednoduchá mohla byť, často som sa zasekol pri rozmyšľaní nad “najlepším spôsobom” akým by sa požadovaná funkčnosť mohla vložiť bez porušenia existujúcej funkčnosti. Toto bola veľakrát, aj pri jednoduchých veciach, nie úplne jednoznačná otázka. Viedlo to teda ku tomu, že aj napriek jednoduchej požiadavke, trvala implementácia netriviálne veľa času. Toto síce nie je úplne chybou kódu samotného, ale príde mi to dosť dôležité pri vývoji ľubovoľného softvéru. Najefektívnejšie riešenie, je prestať hľadať “najlepšie” riešenie, a miesto toho sa pokúsiť o čo najjednoduchšie riešenie. Toto znovu nie je jednoznačná otázka, ale je podstatne ľahšie nájsť nejaké jednoduché riešenie, ktoré sa v následných vlnách refaktorizácie upraví požadovanú efektivitu a prehľadnosť podľa potreby.

4.2 Popis prototypu

Na základe dizajnu sme teda začali implementovať prototyp našej strategickej hry. V súlade s dizajnom sme implementovali hernú plochu, ktorá obsahuje 5 regiónov, ako možno vidieť na obrázku 4.1.

Globálne suroviny, ktoré sme implementovali sú:

Energia - získava sa stavbou elektrární, postavené produkčné budovy ju vyžadujú.

Jedlo - implementované podľa dizajnu.

Železo - ťaží sa vo výrobnjej budove “baňa”, používa sa na stavbu budov a produkciu traktorov

Traktory - produkujú sa vo výrobnjej budove zo železa, používajú sa na výrobu roľníckej funkcie v regióne.

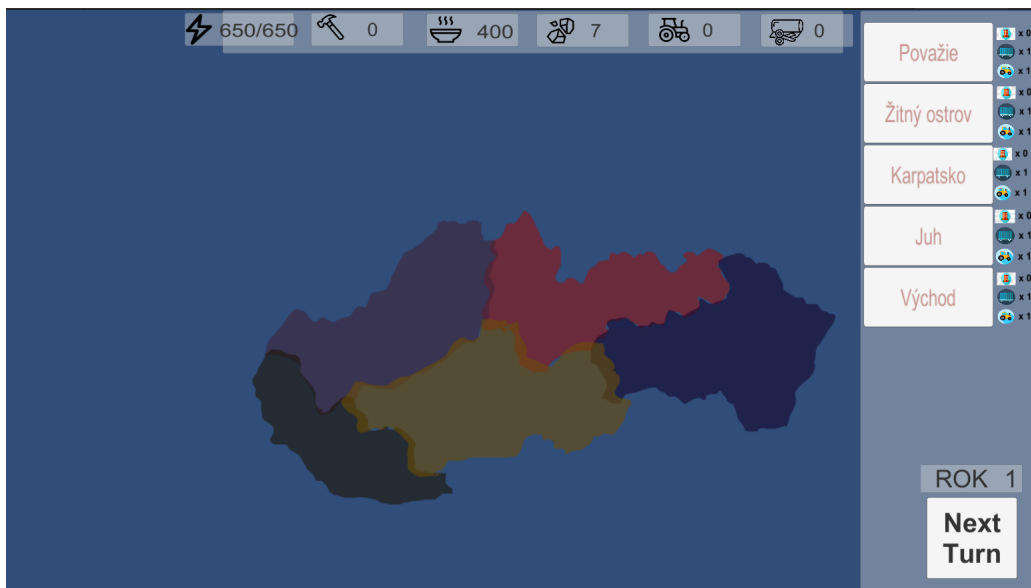
Textil - účel len pre potreby obyvateľstva.

Každý región má vlastnú populáciu, delenú medzi roľníkov a robotníkov, a ku tomu prislúchajúcu produkciu jedla a iných surovín.

Okrem toho však má každý región 2 ďalšie štruktúry, JRD a infraštruktúru.

JRD udáva efektivitu produkcie jedla, násobením produkcie jedla v danom regióne, násobok začína na 1 a zvyšuje sa o 0.5 každú úroveň. Dá sa vylepšiť za cenu traktorov. Infraštruktúra poskytuje násobok všetkým produkciám v danom regióne. Dá sa vylepšiť za cenu železa.

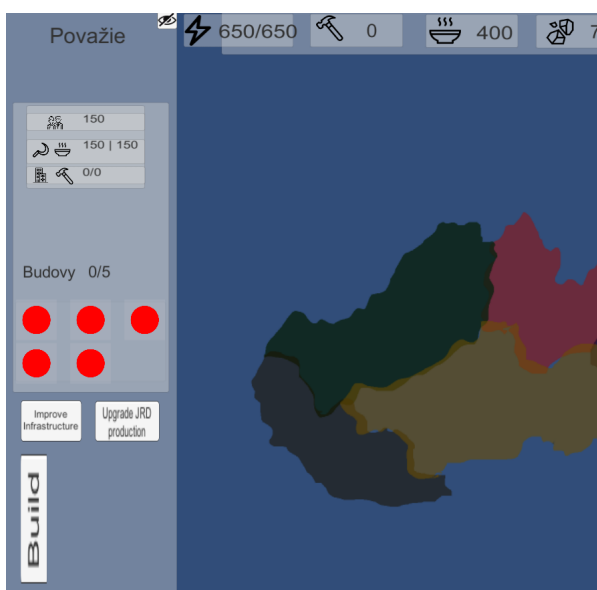
Hráč vie interagovať s regiónom pomocou kliknutia na daný región na mape, alebo na pravej strane zo zoznamu regiónov. Kliknutie na región vykoná selekciu, pre zvolený



Obr. 4.1: Náhľad na hernú plochu

región sa na ľavej strane objaví lišta ktorá popisuje príslušné informácie, zoznam postavených produkčných budov a úroveň štruktúr, ako aj možnosti budovania produkčných štruktúr, pokiaľ sú niektoré dostupné. Toto je zobrazené na obrázku 4.2.

Ak hráč postaví novú štruktúru, tá nezačne hneď produkovať, zvyšok ťahu len nečinne čaká. Na konci ťahu sa overí či je v danom regióne dosť robotníkov, a ak áno začne sa produkcia (ak nie, táto budova bude nečinne čakať do zrušenia, alebo do získania nových voľných robotníkov).



Obr. 4.2: Detaily zvoleného regiónu

Po každom 5-tom ťahu sa objaví plánovacie okno, ktoré hráča informuje o jeho pro-

dukcií za posledných 5 ťahov a porovná ich s plánom ktorý si stanovil pred 5 ťahmi (pokiaľ sa tak stalo). Po potvrdení sa zobrazí ďalšie okno, kde si hráč nastaví svoj plán na najbližších 5 ťahov. Po potvrdení sa vráti do pohľadu na krajinu. Plánovanie má v tomto štádiu veľmi limitovanú funkciu, hráč má plnú vôľu a systémy trestania nenaplnenia plánov nie sú súčasťou prototypu.

4.3 Porovnanie s podobnými prácami

Priemysel videohier narástol do rozsiahlych rozmerov, pochopiteľným následkom tohoto faktu je, že na túto tematiku existuje mnoho rôznych prác. Väčšina z týchto prác je vo forme tutoriálov, väčšinou platených, ktorých hlavným účelom je naviesť čitateľa k metóde, alebo niekoľkým metódam, ako vytvoriť vlastnú hru, ak je to tutoriál zameraný na implementáciu, alebo naviesť čitateľa k dosiahnutiu ako tak správneho dizajnu hry, pokiaľ je tutoriál zameraný na dizajn hry. Moja práca nemá s týmito druhmi prác skoro žiadne spoločné ciele. Ďalej mám pre porovnanie, tému podobnú k tej mojej, publikovanú prácu z VŠE v Prahe, ktorá sa venuje návrhu a vývoji strategickej hry v Unity. Účelom tejto práce je, podľa autora,

“Cíle byly stanovené dva. Prvním cílem je teoretická část, která popisuje životní cyklus vývoje her a jeho jednotlivé etapy. následováno porovnáním životního cyklu vývoje softwaru s životním cyklem her. Dalším cílem práce je praktická část, která zahrnuje následnou tvorbu koncepce a prototypu hry.”[28]

Toto znie principiálne podobne ako moja vlastná téma, avšak z môjho pohľadu sa autor viacej zameriava na popis komerčného vývoja hier, toto je vidno pri jeho popise jednotlivých rôl vývojarského tímu, ako aj fakt že má celú podkapitolu venovanú financovaniu, a generovaniu finančných prostriedkov pri vývoji videohry. Ja by som sa v mojej práci chcel týmto aspektom vývoja úplne vyhnúť, predovšetkým marketingu a financiám okolo videohier. Vo všeobecnosti sa chcem na vývoj videohry pozeráť nie ako na tvorbu komerčného produktu, ale skoro ako na tvorbu umeleckého diela, len z hľadiska dizajnu, a samotného programovania. Samozrejme tieto hľadiská nie sú postačujúce na vyvorenie skutočnej hry, preto musím pracovať s istými predpokladmi (napríklad grafické asety, budem predpokladať že existujú grafické asety s vhodnými vlastnosťami, bez toho aby som popisoval dané vlastnosti). V implementačnej časti mi významne pomôže spolupráca s konzultantom, ktorý mi zabezpečí mnohé z týchto predpokladov. Jeden bod ktorý by sa dal vytknúť je, že ak sa zaoberám hrou z hľadiska dizajnu, bude ma zaujímať ako navrhnuť hru ktorá bude vhodná pre široký výber potenciálnych hráčov, ktorý si budú vedieť hru užiť, a teda budú chcieť hrať viac. Toto by sa dalo označiť ako marketingová stratégia, a je ľahké za tým vidieť koreláciu finančných príjmov, zatiaľ čo to je pravda, chcel by som sa vyhnúť tomu explicitne

načrtnúť túto paralelu, a zameriavať sa len na dizajn “dobrej” hry, a nie “komerčne úspešnej” hry.

Záver

V tejto práci sme mali stanovené dva hlavné ciele, prvým bola teoretická časť, kde sme sa pozreli na žáner strategických hier a analyzovali ich vlastnosti a špecifikácia, ktoré ich odlišujú od iných hier. Najskôr sme si ich rozdelili podľa prístupu k vykonávaniu akcií, kde rozoznávame turn-based alebo real-time stratégie. Potom sme sa pozreli na kategorizáciu strategických hier podľa kľúčových vlastností a niektorých herných mechaník, okolo ktorých sú hry v danej kategórii postavené.

Spomenuli sme si rozdiel medzi stratégiou, dlhodobým plánovaním s rozsiahlimi následkami, a medzi taktikou, krátkodobé dosahovanie okamžitého cieľa, pozreli sme sa v akom sú vzájomnom vzťahu a povedali sme si, že rozlišujeme strategické hry, ktoré zväčša obsahujú stratégiu aj taktiku, a taktické hry, ktoré obsahujú len taktické prvky bez zamerania na strategické plánovanie.

Následne sme sa pozreli na vybranú selekciu konkrétnych strategických hier a preskúmali niektoré ich vlastnosti a mechaniky.

Ďalej sme si priblížili postup vývoja strategických hier, kde sme sa snažili poukázať na podobnosť medzi vývojom bežného softvéru a vývojom videohier, a to tým, že sme načrtli podobnosť medzi životnými cyklami vývoja softvéru a videohier. Okrem toho sme si však ukázali aj prvky dizajnu špecifické pre strategické hry, ktoré sa pomerne priamočiara viažu na implementáciu mechaník. Analýza týchto prvkov bola inšpirovaná knihou *Andrew Rollings and Ernest Adams on Game Design*, ktorej autori identifikovali niekoľko hlavných prvkov dizajnu strategických hier, ako téma dizajnovanej hry, ktorá určuje spôsob akcií ktorými hráč interaguje s herným svetom, alebo dôležitosť herného rozhrania. Populárne témy zahŕňajú dobývanie, alebo objavovanie. Tieto prvky boli analyzované a zdokumentované v praktickom a užitočnom štýle.

Následne sme sa pozreli na softvérové nástroje používané pri vývoji strategických hier, ktoré sú najviac zamerané na programovaciu stránku projektu, takže nástroje podporujúce vývoj a údržbu funkčného kódu a úpravu herného sveta z už existujúcich častí finálneho projektu.

Na konci sme sa venovali nášmu druhému cieľu, ktorým bol návrh dizajnu strategickéhohry a následná implementácia prototypu založeného na tomto dizajne. V tejto časti sme popísali dizajn navrhutej aplikácie a popis implementovaného prototypu,

ako aj vývojových štádií sprevádzajúcich jeho implementáciu.

Literatúra

- [1] New battles on old battlefields, 2010. <https://www.general-staff.com/tag/kriegsspiel/>.
- [2] Erik Bethke. *Game development and production*. Wordware Publishing, Inc., 2003.
- [3] Silvester Buček and Martina Kobetičová. Establishing new genres in digital games: The auto battler case study. *Acta Ludologica*, 3(1):46–66, 2020.
- [4] Done By and Osama Alkhoun. Language specification. 2011.
- [5] Bradley Camburn, Vimal Viswanathan, Julie Linsey, David Anderson, Daniel Jensen, Richard Crawford, Kevin Otto, and Kristin Wood. Design prototyping methods: state of the art in strategies, techniques, and guidelines. *Design Science*, 3, 2017.
- [6] Jeff Craighead, Jennifer Burke, and Robin Murphy. Using the unity game engine to develop sarge: a case study. In *Proceedings of the 2008 Simulation Workshop at the International Conference on Intelligent Robots and Systems (IROS 2008)*, volume 4552, 2008.
- [7] Simon Dor. Strategy in games or strategy games: dictionary and encyclopaedic definitions for game studies. *Game Studies*, 18(1), 2018.
- [8] Blizzard Entertainment. Starcraft 2, 2010. starcraft2.com/en-us/game.
- [9] Blizzard Entertainment. Hearthstone, 2014. playhearthstone.com/en-us.
- [10] Martin Fowler, Jim Highsmith, et al. The agile manifesto. *Software development*, 9(8):28–35, 2001.
- [11] Lawrence Freedman. *Strategy: A history*. Oxford University Press, 2015.
- [12] 2K Games. Xcom: Enemy unknown, 2012. xcom.com/xcom-enemy-unknown/.
- [13] Merge Games. Frostpunk, 2018. www.frostpunkgame.com/.
- [14] Subset Games. Into the breach, 2018. subsetgames.com/itb.html.

- [15] Virgin Games. Dune, 1992. playclassic.games/games/adventure-dos-games-online/play-dune-online/.
- [16] GORAN JEVTIC. What is sdlc? phases of software development, models, best practices. 2019.
- [17] Adam Kramarzewski and Ennio De Nucci. *Practical Game Design*. Packt Publishing, 2018.
- [18] Yu Beng Leau, Wooi Khong Loo, Wai Yip Tham, and Soo Fun Tan. Software development life cycle agile vs traditional approaches. In *International Conference on Information and Network Technology*, volume 37, pages 162–167, 2012.
- [19] Jeannie Novak. *Game development essentials: an introduction*. Cengage Learning, 2011.
- [20] Rido Ramadan and Yani Widayani. Game development life cycle guidelines. In *2013 International Conference on Advanced Computer Science and Information Systems (ICACISIS)*, pages 95–100. IEEE, 2013.
- [21] Damon Reece. Best tower defense games of all time. *Gameranx: Top Rated Games, Reviews and News*, 2015.
- [22] Andrew Rollings and Ernest Adams. *Andrew Rollings and Ernest Adams on game design*. New Riders, 2003.
- [23] Razvan Sibii. Grand strategy. *Encyclopedia of United States National Security*, pages 299–300, 2006.
- [24] Miguel Sicart. Defining game mechanics. *Game Studies*, 8(2):1–14, 2008.
- [25] Wube Software. Factorio, 2012. www.factorio.com/.
- [26] Bezchotnikova Svitlana and Bezchotnikova Alina. Game mechanics as videogame genre identifier. *Global Media Journal*, 16(30):1–6, 2018.
- [27] Unity Technology. Unity documentation. <https://docs.unity3d.com/Manual/Hierarchy.html>.
- [28] Jan Švejda. Návrh a vývoj hry v enginu unity - strategie. 2021.

Príloha A: obsah elektronickej prílohy

V elektronickej prílohe priloženej k práci sa nachádza pilotná aplikácia. Verejný GitHub repository tejto aplikácie je dostupný na nasledujúcej adrese <https://github.com/horvath203/PilotApplication>.

Tento repository bude priložený aj ako príloha pri odovzdanej práci. V repository sa nachádza textový návod na inštaláciu a spustenie.