

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Vývoj herní aplikace v Unreal Engine

Michal Dvořák

© 2023 ČZU v Praze

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Michal Dvořák

Informatika

Název práce

Vývoj herní aplikace v Unreal Engine

Název anglicky

Game application development in Unreal Engine

Cíle práce

Cílem bakalářské práce je charakterizovat proces vývoje herních aplikací a tím pomoci začínajícím vývojářům objasnit jeho důležité součásti. Bude poukázáno na často opomíjený project management a bude kladen důraz na fáze v procesu herního vývoje jako je pre-production, production a post-production. Dílčím cílem práce je tvorba prototypu herní aplikace, který bude připraven k publikaci na platformě Steam a následně popis, jak samotná publikace probíhá.

Metodika

Metodika zpracování teoretické části práce vychází ze studia odborných zdrojů. Na základě jejich syntézy bude charakterizován vývoj herních aplikací.

Znalosti z teoretické části budou aplikovány v části praktické, kde bude navrženo konkrétní téma a metodika vývoje hry. Prototyp hry bude implementován v prostředí Unreal Engine 5 za použití programovacího jazyka C++, vizuálního skriptovacího jazyka Blueprints a za pomoci dalších technologií, které Unreal Engine a Epic Games nabízejí. Během vývoje prototypu bude využito 3D modelování, texturování a animace.

Výsledná aplikace bude otestována jak z funkčního hlediska, tak i z hlediska použitelnosti a uživatelské zkušenosti (UX). Na základě provedených testů bude navrženo, čím se vývoj bude dále zabývat.

Doporučený rozsah práce

35-40 stran

Klíčová slova

Unreal Engine, Steam, počítačová hra, C++, 3D grafika, Blender, vývoj

Doporučené zdroje informací

C++ language documentation: Learn to use C++ and the C++ standard library. [online]. [cit. 2022-05-02].

Dostupné z: <https://docs.microsoft.com/en-us/cpp/cpp/?view=msvc-170>

Get Started [online]. [cit. 2022-05-02]. Dostupné z: <https://www.blender.org/support/tutorials/>

HRISTOV, Georgi a Diyana KINANEVA. A Workflow for Developing Game Assets for Video Games [online].

2021 [cit. 2022-05-02]. Dostupné z: DOI: 10.1109/HORA52670.2021.9461355

Steamworks [online]. [cit. 2022-04-29]. Dostupné z: <https://partner.steamgames.com/>

STEFYN, Nadia. How video games are made: the game development process [online]. 2019 [cit.

2022-05-02]. Dostupné z: <https://www.cgspectrum.com/blog/game-development-process>

TATA A. S. K., Ishwarya, Naidu R. CHINA APPALA, Meghana K. a Reddy G. PRABHAKAR. MODERN

APPROACH TO DESIGN AND INTEGRATE CONCEPTUAL METHODS IN VIDEO GAMES WITH ARTIFICIAL

INTELLIGENCE. MATERIALS TODAY-PROCEEDINGS [online]. 2017, 9100-9106 [cit. 2022-05-02]. ISSN

2214-7853. Dostupné z: <https://doi.org/10.1016/j.matpr.2017.07.265>

Unreal Engine 5 Documentation [online]. [cit. 2022-04-29]. Dostupné z:

<https://docs.unrealengine.com/5.0/en-US/>

Předběžný termín obhajoby

2022/23 LS – PEF

Vedoucí práce

Ing. Tomáš Benda

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 31. 10. 2022

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 24. 11. 2022

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 14. 03. 2023

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Vývoj herní aplikace v Unreal Engine" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15.3.2023

Poděkování

Rád bych touto cestou poděkoval vedoucímu bakalářské práce Ing. Tomáši Bendovi za vstřícný a profesionální přístup, časté konzultace, cenné rady a připomínky. Dále bych chtěl poděkovat mé milované Karolínce za podporu a pomoc.

Vývoj herní aplikace v Unreal Engine

Abstrakt

Tématem této bakalářské práce je popis průběhu vývoje herní aplikace pro začínající indie vývojáře.

V úvodu teoretické části je představen herní engine Unreal Engine, jehož systémy jsou v průběhu práce také zmiňovány ve spojitosti s určitými tématy kapitol. Dále práce poukazuje na Project management, který má za úkol pomáhat vývojářům udržet ucelený projekt. V dalších kapitolách jsou popsány jednotlivé fáze vývoje herní aplikace od Pre-Production, Production, který rozepisuje témata level designu, animací, programování, verzování a tvorby herních assetů, po Post-Production. Závěr teoretické části obsahuje popis publikace hry na platformu Steam.

Praktická část využívá poznatky získané v teoretické části a popisuje postup zpracování dílčího cíle bakalářské práce. V první části je navržena hra, game design document, project management a je připraveno verzování projektu za pomoci Perforce serveru. Dále jsou do prototypu vloženy vymodelované základní 3D assety, kterým je následně vygenerována UV mapa, a které jsou otexturovány. Poslední část popisuje tvorbu prototypu herní aplikace v Unreal Engine 5. V C++ je implementován AStar (A*) algoritmus pro hledání nejkratší cesty vlaků a pro jejich následný pohyb po kolejích. Dále je vytvořeno základní UI a tři levely načítané pomocí level streaming. V závěru praktické části je popsáno testování prototypu herní aplikace a jsou shrnuty výsledky z dotazníků.

Obsah teoretické části práce (a s tím spojená i praktická část práce, která poznatky aplikuje) může sloužit začínajícím vývojářům jako manuál pro vývoj herní aplikace a pomoci jim u tvorby prvního projektu. Výsledná verze prototypu hry byla otestována metodami System Usability Scale a Cognitive think-aloud protocol. Výsledky testování potvrdily správné směřování vývoje hry, definovaly nedostatky prototypu hry a zároveň bylo zjištěno, jakým směrem se má budoucí vývoj ubírat.

Klíčová slova: Unreal Engine, Steam, počítačová hra, C++, 3D grafika, Blender, vývoj

Game application development in Unreal Engine

Abstract

The main topic of this thesis is a description of the development process of a game application for an aspiring indie developer.

The theoretical part introduces the game engine Unreal Engine whose systems are mentioned throughout the thesis. The thesis highlights Project Management which helps developers to maintain a coherent project. The next chapters describe the different phases of game application development from Pre-Production, Production (focuses on topics such as level design, animation, programming, version control and game asset creation) to Post-Production. The theoretical part includes a description of publishing a game on the Steam platform.

The practical part uses the knowledge gained in the theoretical part. In the first part, the game is designed, the game design document and project management are created and the project versioning is prepared using Perforce server. The last section describes the prototyping of a game application in Unreal Engine 5. The AStar (A*) algorithm is implemented in C++ to find the shortest path along the tracks. It also describes the creation of the game's basic user interface and the creation of three levels that are loaded by level streaming. Finally, the practical part describes testing of the prototype and summarizes the results from the questionnaires.

The theoretical part (and the related practical part of the thesis) can serve as a guide for beginning developers to develop a game application and help them with their first project. The final version of the game prototype was tested with the System Usability Scale and Cognitive think-aloud protocol methods. The results of the testing confirmed the correct direction of the game development, defined the weaknesses of the prototype game, and also identified the direction of future development.

Keywords: Unreal Engine, Steam, Computer Game, C++, 3D Graphics, Blender, Development

Obsah

1 Úvod.....	11
2 Cíl práce a metodika	12
2.1 Cíl práce	12
2.2 Metodika	12
3 Teoretická východiska	13
3.1 Unreal Engine.....	13
3.2 Project management	15
3.2.1 Tradiční přístup.....	15
3.2.2 Agilní přístup	16
3.3 Project management ve vývoji her.....	19
3.4 Pre-production fáze	20
3.4.1 Game Design Document	20
3.4.2 Prototyp.....	21
3.5 Production fáze.....	22
3.5.1 Level Design	23
3.5.2 Herní assety.....	24
3.5.2.1 Princip tvorby 3D herních assetů	27
3.5.3 Animace	35
3.5.4 Audio	37
3.5.5 Programování a skriptování	38
3.5.6 Version Control.....	39
3.6 User Experience	40
3.7 Testování použitelnosti	40
3.7.1 Kvalitativní / kvantitativní	41
3.7.2 Moderované / nemoderované.....	41
3.7.3 Osobní / vzdálené.....	41
3.7.4 System Usability Scale (SUS)	42
3.7.5 Cognitive think-aloud protocol (scénáře)	42
3.8 Post-production fáze.....	42
3.9 Komerční nasazení hry vytvořené v Unreal Engine	43
3.10 Vydání hry na platformu Steam	43
3.10.1 Registrace.....	43
3.10.2 Vyžadované informace	44
3.10.2.1 Platební údaje	44
3.10.2.2 Daňové informace	44
3.10.2.3 Steam Greenlight vs. Steam Direct	45

3.10.3	Nahrání aplikace na Steam.....	45
3.10.3.1	Depot.....	46
3.10.3.2	Build	46
3.10.3.3	Branch.....	46
3.10.3.4	Package	47
3.10.4	Store Page	47
3.10.5	Nastavení ceny	50
3.10.6	DRM.....	50
4	Vlastní práce	51
4.1	Návrh hry.....	51
4.2	Game Design Document	51
4.3	Project management	52
4.4	Assety	55
4.4.1	Modelování	55
4.4.2	Příprava modelu na UV mapping.....	56
4.4.3	UV mapování	57
4.4.4	Texturování	58
4.5	Prototyp	59
4.5.1	Vytvoření projektu	59
4.5.2	Verzování	60
4.5.3	Pathfinding.....	61
4.5.4	User Interface	64
4.5.5	Levely.....	65
4.5.6	Gameplay	66
5	Výsledky a diskuse	67
5.1	Testování	67
5.1.1	SUS dotazník.....	68
5.1.2	Vlastní dotazník	69
5.1.3	Vyhodnocení	70
6	Závěr.....	73
7	Seznam použitých zdrojů.....	75
8	Seznam obrázků, tabulek a grafů	82
8.1	Seznam obrázků	82
8.2	Seznam tabulek.....	83
8.3	Seznam grafů.....	84
9	Přílohy	85
9.1	Snímky z prototypu hry.....	85

1 Úvod

Videoherní průmysl se za poslední čtyři desítky let změnil k nepoznání. Videohry jsou v současnosti hlavním segmentem zábavního průmyslu. Nezřídka přinášejí desítky miliard dolarů ročně, a přímo tak soupeří s filmy, seriály nebo hudbou. Za rozšíření herního průmyslu může dostupnost hardwaru, který přinesl obrovský výpočetní výkon a je dostupný téměř všem, technologický pokrok, který dal rozlišení až 8K nebo Ray Tracing, růst sociálních médií a streamovacích služeb, přesun her z kamenných obchodů na online platformy a mnoho dalšího.

Vývoj her zahrnuje mnoho oborů a je vyžadována rozmanitá sada dovedností jako 3D modelování, programování, design, práce se zvukem nebo například psaní scénáře. Pod vývojem her si lze také představit postup softwarového inženýrství, tedy proces zahrnující fáze koncepce, preprodukce (pre-production), produkce (production) a postprodukce (post-production).

Rychlý vývoj videoherního průmyslu zapříčinil i růst indie vývojářů, nezávislých vývojářů pracujících v malé skupině nebo sami za sebe, snažících se vytvořit svou originální jedinečnou hru a předat tímto prostředkem své myšlenky a nápady ostatním lidem. Začít s vývojem her může téměř každý, kdo má chuť a zapálení. V nabídce je mnoho nástrojů a aplikací zdarma (Unreal Engine, Unity, Blender), existují tisíce různých videonávodů, článků a dokumentací. Ovšem velké množství informací je pro začínající vývojáře i nevýhodou a hledání důležitých pojmů a toho, na co se zaměřit, je složitější než se může na první pohled zdát.

Bakalářská práce pomáhá těmto začínajícím vývojářům, a to především v oblasti vývoje indie 3D hry, usměrňuje jejich práci a cíle, vysvětluje základní pojmy ve vývoji her aby čtenářům bylo jasné, na co se primárně soustředit.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem bakalářské práce je charakterizovat proces vývoje herních aplikací a tím pomoci začínajícím vývojářům objasnit jeho důležité součásti. Bude poukázáno na často opomíjený project management a bude kladen důraz na fáze v procesu herního vývoje jako je pre-production, production a post-production.

Dílním cílem práce je tvorba prototypu herní aplikace, který bude připraven k publikaci na platformě Steam a následně popis, jak samotná publikace probíhá.

2.2 Metodika

Metodika zpracování teoretické části práce vychází ze studia odborných zdrojů. Na základě jejich syntézy je charakterizován vývoj herních aplikací. Teoretická část je především zaměřena na vývoj 3D typu hry a ve vybraných kapitolách zmiňuje, jakým způsobem nebo jakými nástroji k danému tématu přistupuje Unreal Engine.

Znalosti z teoretické části byly aplikovány v části praktické, kde bylo navrženo konkrétní téma a metodika vývoje hry. Prototyp hry je implementován v prostředí Unreal Engine 5 za použití programovacího jazyka C++, vizuálního skriptovacího jazyka Blueprints a za pomoci dalších technologií, které Unreal Engine a Epic Games nabízejí. Během vývoje prototypu bylo využito 3D modelování, texturování a animace.

Výsledná aplikace byla otestována jak z funkčního hlediska, tak i z hlediska použitelnosti a uživatelské zkušenosti (UX) metodami System Usability Scale (SUS) a Cognitive think-aloud protocol. Na základě výsledků z provedených testů je navrženo, čím se vývoj bude dále zabývat.

3 Teoretická východiska

V teoretické části je popsán project management a hlavní fáze vývoje herní aplikace. Závěr teoretické části je věnován publikaci hry na platformě Steam.

3.1 Unreal Engine

Unreal Engine je jedním z nejpoužívanějších nástrojů pro tvorbu her, architektonické a automobilové vizualizace, tvorbu filmového a televizního obsahu, živých vysílání, školení, simulací a dalších aplikací v reálném čase. Unreal Engine je vyvíjen firmou Epic Games již od roku 1998. V tomto roce vyšla na trh první hra s názvem Unreal. Jádro enginu je napsáno v jazyce C++ a podporuje platformy Microsoft Windows, Linux, Max OS, Mac OS X a PC. I přestože byl vyvinut primárně pro akční hry z pohledu první osoby (first-person), využil se úspěšně i v jiných žánrech jako například RPG (Role playing game – hra na hrdiny), MMO RPG (Massively Multiplayer Online Game – online hra pro velký počet hráčů) a další (Evan, 2016).

Za rozšíření Unreal Enginu mimo herní business může jeho podpora a neustálý vývoj Virtual Production, ve které hraje velmi důležitou roli. Je jedním z nejpokročilejších real time 3D nástrojů sloužící k vytváření XR (Extended Reality - rozšířená realita) a stává se novým standardem právě ve vytváření filmů nebo televizních pořadů (Ayman, 2022).

Nejnovější verzí enginu je Unreal Engine 5 (UE5). UE5 přináší mnoho nových funkcí a nástrojů umožňující herním vývojářům urychlit a zefektivnit vývoj. Mezi nejčastěji zmiňované patří například:

Lumen

Lumen je řešení globálního osvětlení a odrazů, které okamžitě reaguje na změny scény a světla. Lumen implementuje softwarový Ray Tracing (RT). Umožňuje tak dynamickou globální iluminaci a odrazy, a to i pro grafické karty bez podpory RT. Odpadá tak nutnost zapékání lightmap po každé změně statických světel. Pro karty podporující RT nabízí Lumen i možnost hardwarového Ray Tracingu. Lumen artistům a designérům dává možnost vytvářet dynamičtější scény s větším realismem (Epic Games, 2023).

Nanite

Tento systém umožňuje vykreslovat scény s miliardami trojúhelníků v reálném čase. Designéři mohou využít assety (kapitola 3.5.2) filmové kvality bez razantního úbytku snímkové frekvence. Při využití Nanite assetu odpadá nutnost ručně model optimalizovat

pomocí standardních technik jako je zapékání normal map (kapitola 3.5.2.1) a vytváření LODů (Level of Detail). Nanite optimalizuje model automaticky (Epic Games, 2023).

MetaHuman

MetaHuman je bezplatný cloudový nástroj, který umožňuje vytvářet digitální postavy v intuitivním editoru. Každé postavě lze přizpůsobit účes, obličej, tělesné proporce, výšku apod. (Epic Games, 2023).

Mesh to MetaHuman

Mesh To Metahuman rozšiřuje základní nástroj MetaHuman. Umožňuje převést vlastní model obličeje (např. photoscan) na MetaHuman. Takto převedená postava má proporce obličeje předlohy a je pro ni zároveň připraven kompletní Rig pro animace (Epic Games, 2023).

Vývoj hry je složitý proces. V současné době si ovšem lze mezi herními enginy vybírat (Unreal Engine, Unity, Godot, Cry Engine), což pomáhá k ušetření měsíců práce a optimalizace, především vývojářům začátečnickům.

I přesto, že je Unreal Engine připravený pro vývoj AAA her (například *Star Wars Jedi: Fallen Order* nebo *Hogwarts Legacy*), a je tak velmi obsáhlý, zůstává mu zároveň přívětivost pro začínající vývojáře. Kvůli výše zmíněným novinkám (Nanite, Lumen atd.), které zahrnuje verze Unreal Engine 5, se stal Unreal Engine natolik populární, že již lze na internetu najít tisíce výukových videí a výukových programů. Unreal Engine dále usnadňuje začátečnickům práci množstvím předpřipravených tříd a skriptů, nabízí šablony her (například hra z první osoby nebo hra ze třetí osoby) nebo nabízí mnoho assetů zdarma, zahrnující animace, prostředí, materiály, textury či efekty. Dalším důležitým bodem, proč je vhodným herním enginem pro začínající vývojáře, je jeho vizuální skriptovací jazyk Blueprints (kapitola 3.5.5), díky němuž může vytvořit plnohodnotnou hru i člověk bez předchozích zkušeností s programováním a bez napsání řádky kódu (ITuts Editor, 2022).

Na základě výše zmíněných důvodů, proč je Unreal Engine vhodný i pro začínající vývojáře, autor v následujících kapitolách teoretické části bakalářské práce zmiňuje, jaké systémy Unreal Enginu se pro danou oblast kapitoly využívají, aby čtenář získal obecný přehled. V praktické části autor Unreal Engine použil pro vývoj prototypu hry.

3.2 Project management

Neboli projektové řízení, představuje soubor činností/úkolů, které je potřeba naplánovat a provést, aby se dosáhlo za předpokládaných nákladů, v požadovaném termínu, během určitého času a v určité kvalitě jasně definovaných cílů. Tímto cílem může být cokoliv napříč všemi odvětvími, od webové stránky, po plánování stavby. Stručně lze project management charakterizovat jako účinné a efektivní dosahování změn a cílů. Zabývají se jím velké společnosti i jednotlivci (ManagementMania, 2016).

3.2.1 Tradiční přístup

Tradiční řízení projektu je takový přístup, který je prováděn v lineární posloupnosti. K realizaci projektu se přistupuje krok za krokem a nepočítá se s případnými změnami. Je nezbytné definovat rozsah a požadavky projektu na samotném začátku (Sennett, 2022).

Každý projekt se řídí stejným životním cyklem, který zahrnuje pět fází:

1. **Zahájení projektu** – Diskuze o nápadu, rozpočtu, směru, časovém plánu nebo problému, kvůli kterému se chce projekt uskutečnit (Jaimovich, 2022).
2. **Plánování** – Stanoví se podrobný harmonogram, konkrétní cíle a požadavky projektu. Nakonec se vytváří plán řízení projektu, který obsahuje předchozí zmíněné prvky (Jaimovich, 2022).
3. **Realizace** – To, co je naplánováno, se realizuje. Vychází se z informačního základu z předchozích fází *zahájení* a *plánování*. Přínosné je stanovení milníků. Nejdůležitější je komunikace (například informovat všechny zúčastněné strany o dosaženém pokroku) (Sládeček, 2017; Jaimovich, 2022).
4. **Kontrola** – Sleduje se pokrok v práci a dodržování termínů. Slouží taktéž ke kontrole, zdali jsou cíle plněny, a zdali nedošlo k odchýlení ze stanoveného rozsahu projektu (Jaimovich, 2022).
5. **Uzavření projektu** – Oficiální uzavření projektu. Během uzavírání projektu je možné nalézt nedostatky a to, na čem se dále bude pracovat. Dále je zhodnocena celková práce, která na projektu byla vynaložena (Jaimovich, 2022).

Záleží na konkrétní firmě či jednotlivci, jak se k project managementu postaví a jak ho bude vykonávat, proto nemusí být pořadí fází striktně dodržováno. Dodržování těchto fází ovšem může pomoci k odhalení slabých nebo silných stránek projektu (Sládeček, 2017).

Tradiční přístup zahrnuje metody Vodopádu, Metodu kritické cesty nebo CCPM (Critical chain project management).

3.2.2 Agilní přístup

Agilní přístup je souhrn metodik, které jsou vytvořeny za účelem zvýšení efektivity, produktivity, rychlosti vývoje a pro zlepšení vzájemné spolupráce a komunikace. Souhrn těchto metodik je postaven na takzvaném iterativním vývoji. Práce na projektu je rozdělena na menší úkoly, tzv. *iterace – sprinty* (opakující se bloky, kde je možné zakomponovat změnu). Na konci každé iterace je produkt ve formě prototypu dodán zákazníkovi a na základě jeho zpětné vazby a požadavků jsou prováděny změny. S každou iterací se přidávají nové funkce a projekt postupně roste. Agilní přístup je tak vhodný pro projekty, kde dochází k vývoji produktu – softwaru. Po splnění všech požadavků zákazníka vývoj končí (AltexSoft, 2023; Radigan, 2023a).

Mezi nejčastěji zmiňované metodiky agilního přístupu patří Scrum a Kanban.

Scrum

Metodika Scrum je založena na souboru přesně definovaných postupů a rolí, které musí být zapojeny do procesu vývoje. Scrum se uplatňuje především v projektech vývoje softwaru a videoher. Důvodem je flexibilní přístup, který minimalizuje možnost chyb. Tým tak může držet krok se změnami na trhu a přizpůsobit produkt tomu, co požaduje cílová skupina (Wrike, 2023).

Výchozím bodem procesu je seznam cílů a požadavků, které tvoří plán projektu. Práce týmu je rozdělena do předem stanovených krátkých a periodických časových bloků. Tyto bloky se nazývají *sprinty* a obvykle se pohybují v rozmezí 2 až 4 týdnů. Každý sprint je entitou sám o sobě, to znamená, že na konci sprint poskytuje výsledek, variantu finálního produktu (Agile Alliance, 2023).

Scrum tým

Scrum tým spolupracuje na dodávkách požadovaných přírůstků produktu. Tým se skládá ze tří rolí:

- **Vývojáři / Tým** – Ve Scrum týmu je vývojářem jak každý člen týmu, který odvádí práci, tak i ti členové týmu, kteří jsou mimo vývoj softwaru. Ručí za splnění úkolů během sprintů (Agile Alliance, 2023; Strihová, 2020).
- **Product Owner (vlastník produktu)** - Drží vizi produktu a určuje priority *produktového backlogu* (Strihová, 2020).
- **Scrum Master** – Pomáhá týmu co nejlépe využívat Scrum k budování produktu a zároveň dohlíží na dodržování pravidel. Má na starosti koučování *Product Ownera* i vývojářů a zajišťuje, aby celý tým fungoval efektivně. V ideálním případě je *Scrum Master* zapotřebí pouze v počátcích projektu (Digité, 2023).

Události

Události podporují týmovou spolupráci a zajišťují, aby mezi členy Scrum týmu probíhala v průběhu životního cyklu vývoje produktu neustálá komunikace.

- **Denní schůzka** – Jedná se o krátkou každodenní schůzku. Na těchto schůzkách tým hodnotí vykonanou práci v předchozím dni. Dále tým plánuje, co bude dělat v následujících 24 hodinách. V této době členové týmu diskutují o problémech, které by mohly znemožnit dokončení projektu (Digité, 2023).
- **Sprint** – Určuje časový rámec, ve kterém musí být práce dokončena – často 30 dní. Nové sprinty začínají ihned po skončení předchozího (Strihová, 2020).
- **Plánování sprintu** – Probíhá na začátku sprintu. Cílem je definovat co a jak se bude v rámci sprintu dělat. Stanoví se cíle. Vychází se z *produktového backlogu* (Strihová, 2020).
- **Přezkoumání sprintu** – Ukázka toho, jaká práce byla dokončena s ohledem na *produktový backlog* pro budoucí vývoj. Dokončený sprint se přezkoumává. Měl by být také jasný a hmatatelný pokrok, který se dále prezentuje klientovi (Digité, 2023).
- **Retrospektiva sprintu** – Tým přezkoumá splněné cíle ukončeného sprintu. Vyhodnotí si pokroky i nezdary, aby se napříště tyto chyby neopakovaly. Tato fáze slouží, z hlediska procesu vývoje, k implementaci zlepšení (Digité, 2023).

Artefakty

- **Produktový Backlog** – Uspořádaný seznam toho, jaké jsou cíle, co je potřeba ke zlepšení produktu a co produkt potřebuje k uspokojení potenciálních zákazníků. Přípravuje ho *Product Owner*. Snahou je odpovědět na otázku „Co by se mělo udělat?“ (Digité, 2023).
- **Sprint Backlog** – Je soubor úkolů vybraných z *produktového backlogu*, které se vyberou k provedení sprintu. Sprint Backlog je předpověď vývojového týmu o tom, jaká funkce bude dodána v příštím sprintu a kolik práce bude potřeba k jejímu dodání. Sprint Backlog zobrazuje veškerou práci, kterou vývojový tým označil za nezbytnou k dosažení cíle sprintu. Sprint Backlog bývá zobrazen na tzv. *Scrum board* (Digité, 2023; Strihová, 2020).
- **Přírůstek** – Souhrn dokončených položek *produktového backlogu* během aktuálního sprintu a všech předchozích sprintů. Přírůstek produktu odráží skutečnost, jakého pokroku již bylo dosaženo a co tak bude k dispozici koncovému uživateli (Digité, 2023).

Kanban

Kanban (informační tabule) je metodou agilního přístupu, která byla vytvořena na konci 40 let 20.století japonským inženýrem Taiichi Ohno. Kanban se na rozdíl od Scrumu a od obecně agilního přístupu, které se zaměřují na sprinty a iterace, zabývá stavy práce. Zaměřuje se na rozdělení práce, a to konkrétně na rozdělení na menší úkoly a především na jejich vizualizaci. Vizualizace je vytvořena za pomoci Kanban board (nástěnka) (Kanban Tool, 2023).

Kanban board je nástroj určený k vizualizaci práce, přehlednosti, omezení rozpracované práce a k maximalizaci efektivity. Jedná se o místo, kde členové týmu shromažďují své nápady nebo požadavky od zákazníků. Kanban board může být reprezentována fyzickou nebo digitální nástěnkou. Nástěnka v základu obsahuje tři sloupce (To Do, In Progress, Done), ve kterých jsou úkoly zobrazené v podobě karet (Cards) (Kanbanize, 2023a; Radigan, 2023b).

- **Karty:** Vizualní znázornění úkolů. Karta obsahuje informace o konkrétním úkolu a o jeho stavu, jako je například deadline, příjemce, popis atd. (Kanbanize, 2023b).
- **Sloupec:** Každý sloupec představuje konkrétní činnost, která dohromady tvoří "pracovní postup" (workflow). Jednotlivé karty procházejí pracovním

postupem až do jeho dokončení. Sloupce mohou být jednoduché jako například "K provedení", "Probíhá", "Dokončeno" ale i mnohem složitější (Kanbanize, 2023d).

- **Work-In-Progress:** Omezení maximálního počtu úkolů v jednotlivých fázích pracovního postupu. Omezení rozpracovanosti úkolů umožňuje týmu rychlejší dokončení pracovních položek. Zároveň se tým soustředí pouze na aktuální úkoly (Kanbanize, 2023c).
- **Swimlanes:** Jedná se o horizontální „pruhy“, které je možné použít k oddělení různých činností, týmů, tříd, služeb a dalších (Kanbanize, 2023d).
- **Commitment point:** Je stav, který signalizuje připravenost pracovní položky k realizaci. Když je položka umístěna na *Commitment point*, znamená to, že se tým rozhodl na ní pracovat (Kanbanize, 2023d).

Kanban je tzv. pull systém, který řídí tok práce tím, že nahrazuje to, co je dokončeno jinou položkou. Na Kanban nástěnce se práce vždy pohybuje zleva doprava. Úkoly ze sloupce vlevo jsou přesouvány pouze tehdy, když jsou dokončeny všechny stávající úkoly napravo od něj, anebo v případě naléhavého úkolu. Tím se zamezuje tomu, aby docházelo k nárůstu rozpracovaných (Work-In-Progress) úkolů (Teamhood, 2022).

3.3 Project management ve vývoji her

Projekt management ve vývoji her zastává důležitou roli. Herní projekty mohou být velmi chaotické a celý vývoj bývá dlouhý proces – v závislosti na velikosti vývojářského týmu, typu hry apod. Velká herní studia mohou zahrnovat stovky designérů, programátorů, scénáristů a testerů. Proto je využíván agilní přístup (Nuclino, 2023b).

Project management nevyužívají pouze velká studia, ale i malá. I samotný indie vývojář (indie – nezávislý vývojář nebo hra) může naplno využít tyto nástroje, díky nimž se projekt rozdělí na menší části a vývoj se zefektivní. Neexistuje univerzální nástroj, který by stanovil všem stejný postup. Herní AAA studia a indie vývojáři her mají zcela odlišné pojetí project managementu, pracovní postupy a dostupné nástroje. Není tedy nutné se držet předem stanovených pravidel. Postup si každý přizpůsobí svým potřebám (Corrales, 2021).

Pro project management existuje nespočet softwaru a aplikací. Zde je výběr některých z nich:

- **Trello** – Známy, poměrně jednoduchý Kanban systém.
- **Jira** – Výkonný a komplexní agilní nástroj s možností využít Scrum i Kanban metodu.
- **HacknPlan** – Nástroj orientovaný na vývojáře her využívající Kanban metodu. Obsahuje integrovaný Game Design Document.
- **Nuclino** – Pro řízení herních projektů.
- **Clickup** – Nástroj zaměřený na efektivní správu úkolů.

Jako příklad z praxe lze uvést českou vývojářskou společnost Bohemia Interactive Studio. Bohemia Interactive využívá k project managementu službu GitLab a od společnosti Atlassian využívá nástroje Jira a Confluence.

3.4 Pre-production fáze

Pre-production je fáze plánování, u které začíná každý projekt. Pro uskutečnění nápadu je většinou nezbytné zahájit shromažďování informací o budoucím vývoji. Zaměřuje se na vytvoření základních konceptů a sepsání prvotních návrhů a dokumentů, které popisují budoucí hru. Během pre-production fáze jsou běžně tvořeny prototypy a koncepty prostředí, postav, ovládacích prvků hry, UI (User Interface – uživatelské rozhraní) a mnoho dalšího co bude během vývoje potřeba (Room 8 Studio, 2020).

Nadia Stefyn (2022) zmiňuje otázky na které pre-production fáze hledá odpovědi:

- O čem hra je?
- Jaká je cílová skupina?
- Je pro ni na trhu místo? Jaká je konkurence?
- Na které platformě bude zveřejněna?
- Bude se jednat o hru zdarma?
- Jak dlouho bude trvat vývoj?
- Jaké lidi a zdroje bude vyžadovat?
- Jaký je odhadovaný rozpočet?

3.4.1 Game Design Document

Po shromáždění potřebných informací a požadavků na projekt je vytvořen tzv. Game Design Document (GDD), který obvykle slouží jako podrobný průvodce popisující, jak hra

bude fungovat, její koncept, žánr, děj, typ hry, herní mechaniky, level design a prostředí. GDD přináší jednotnou vizi o projektu pro celý vývojový tým (French, 2022; Playcent Games, 2022).

Stejně jako u project managementu, je pouze na vývojáři nebo studiu, jakým způsobem GDD pojme. Dříve se využíval převážně textový soubor o mnoho stránkách, kde se hra dopodrobna popisovala. Tento způsob je zdoluhavý a výsledek rigidní. Proto se tyto dokumenty potýkají s obtížnou údržbou. Stále je ale používán a není špatnou volbou GDD pojmout touto formou (French, 2022; Raid, 2020).

Spolu s agilním přístupem k project managementu je nyní využívám i agilní přístup k dokumentaci – tedy takový přístup, který podporuje kreativitu, je iterativní a kolaborativní. Při agilním přístupu lze tedy využít i stejné, již zmíněné nástroje. V aplikaci Trello lze přehledně vše napsat do karet a obratem tak i jednoduše vytvářet úkoly, na čem přesně pracovat. Další variantou je použití nástrojů, které umožní všechny nápady a plány vizualizovat. Takovým nástrojem je například Nuclino. Tento přístup je spíše využíván v menších týmech. Velká herní studia se přiklání k jiným řešením (Nuclino, 2023a).

Každý GDD je jedinečný. Existují však některé prvky, které bývají v GDD opakovaně obsaženy (Kliuch, 2022):

- **Obecné informace** – žánr, cílová skupina, podobné hry apod.
- **Název hry** – Hra by měla mít chytlavý název, který odráží to, o čem hra je. Výběr takového názvu velmi pomáhá v marketingu a celkově i v počátcích, kdy se získává pozornost hráčů.
- **Popis hry** – Jedná se o shrnutí hry. Nezachází se do detailů, jde pouze o to, aby bylo po prvním přečtení jasné, jaký žánr hra bude mít a jakého bude typu.
- **Gameplay a mechaniky** – Jak se hra bude hrát a jak funguje. Popisuje se, jaká jsou pravidla, podmínky vítězství a porážek, ovládání, jak se hráč může chovat, jak hra reaguje na různé akce hráče, jak funguje progression system atd.
- **Technický popis** – Popíše se platformy, technologie, herní engine, software, aplikace, jaké se budou používat.
- **Art styl** – Jak hra bude vypadat. Využívají se obrázky, rendery a concept arty.

3.4.2 Prototyp

Je běžné, že se v pre-production fázi začíná s vývojem prototypu hry. Prototypování ovšem může přesahovat i do production fáze. Prototyp hry nemá za úkol vypadat jako

výsledný produkt a nemá tak hru, ve většině případů, prezentovat. Jsou v něm zkoušeny základní mechaniky, zkoumá se funkčnost, UX (kapitola 3.6), použitelnost, hratelnost nebo level design (kapitola 3.5.1). Prototypováním se začíná, aby bylo otestováno, zda sepsané nápady budou funkční a je vhodné v nich dále pokračovat. „Na papíře“ vypadá vše hezky a jako neodmyslitelná součást hry, ale ne vždy to tak je. Do cesty mohou vstoupit překážky jako například technické potíže, nízká dostupnost technologií, nedostatečné zkušenosti vývojáře, nebo mechanika/funkce sepsaná v pre-production fázi nevypadá ve hře dobře. Vývojem prototypu se nezdá přichází i na nové mechaniky a nápady, které se do hry nakonec implementují (Kevuru Games, 2021; MANY WATERS LTD, 2022).

3.5 Production fáze

V production fázi dochází k realizaci nápadů a cílů, které byly sepsány v pre-production fázi. Vynechání pre-production fáze může vést k prodloužení vývoje a k celkové nepřehlednosti projektu. Bez předem stanovených cílů není projekt ucelený a velmi jednoduše se rozpadne.

Production je nejnáročnější a nejdelší fáze, která trvá nejčastěji v rozmezí od 1 až 4 let. Hra se začíná skutečně formovat. Příběh hry postupně dostává finální podobu. Jsou vytvářeny assety (kapitola 3.5.2), jsou stanovena herní pravidla, jsou tvořeny levely a světy, pracuje se na ozvučení, je psán kód a mnoho dalšího (MasterClass, 2021; Stefyn, 2022).

Production fáze lze rozdělit na tyto části: (Stefyn, 2022; MANY WATERS LTD, 2022)

- **First playable** – Začínají se využívat hotové assety. Jedná se o počáteční testování hry, které poskytuje lepší představu o hratelnosti a vzhledu hry. Od finální verze je tato část stále daleko.
- **Vertical slice** – Plně hratelná ukázka, která slouží pro představení hry investorům, v crowdfunding kampani nebo čistě jen potenciálním hráčům.
- **Pre-alpha** – Dokončuje se vize hry a dochází k důležitým rozhodnutím, co ve hře ponechat nebo co vyškrtnout. S dalšími radikálními změnami hratelnosti není po této fázi počítáno.
- **Alpha** – Do hry jsou přidány všechny důležité funkce. Hru je možné hrát uceleně od začátku až do konce. Herní mechanismy jsou funkční. Stále je ale nutné přidat některé prvky a hledat a opravovat chyby.

- **Beta** – Tým se soustředí především na optimalizaci a na technické a estetické nedodělky, nikoliv na přidávání nových prvků.
- **Final release** – Také označován jako “Gold Master”. Vše je dokončeno a hra je připravena k vydání.

3.5.1 Level Design

Level, neboli úroveň, je prostor, ve kterém se hra odehrává. Nastavuje hráči určité hranice, kde se může pohybovat a interagovat s okolím.

Level design je herní vývojářská disciplína, která zahrnuje vytváření levelů, misí nebo etap videoher. To se provádí pomocí editorů – takový editor běžně nabízí samotný herní engine. Editory úrovní taktéž mohou být zahrnuty do již vydaných her, aby hráči mohli vytvářet své vlastní levely, a vnesly tak do hry nový obsah. Level design je technický i umělecký proces (Techopedia, 2017).

Na rozdíl od environment designu, který se zaměřuje na vizuální stránku levelu, se level design zabývá hráčskými schopnostmi, herními mechanikami, překážkami a vším dalším, co vytváří pozitivní UX. Soustředí se tak na funkčnost a možnosti levelu, nikoliv jen na vizuální stránku. Level designér si musí předem představit herní zážitek a to tím, že se vžije do hráče, do jeho myšlení a pokouší se přijít na všechny možnosti, jak by se hráč v daném levelu mohl chovat. Potřebuje přemýšlet o logice, událostech a akcích, o podmínkách, které musí být splněny, aby hra reagovala na pokyny hráče, a o výzvách, kterým bude hráč čelit (Demirçin, 2022).

Level design nemá žádná pevně stanovená pravidla a samotný proces designování se liší studio od studia. Proto bude níže popsané workflow v co nejjobecnějším měřítku, aby odpovídalo většině projektů, se kterými se začínající vývojář běžně setkává.

Ali Youssef (2022) level design shrnuje do těchto fází:

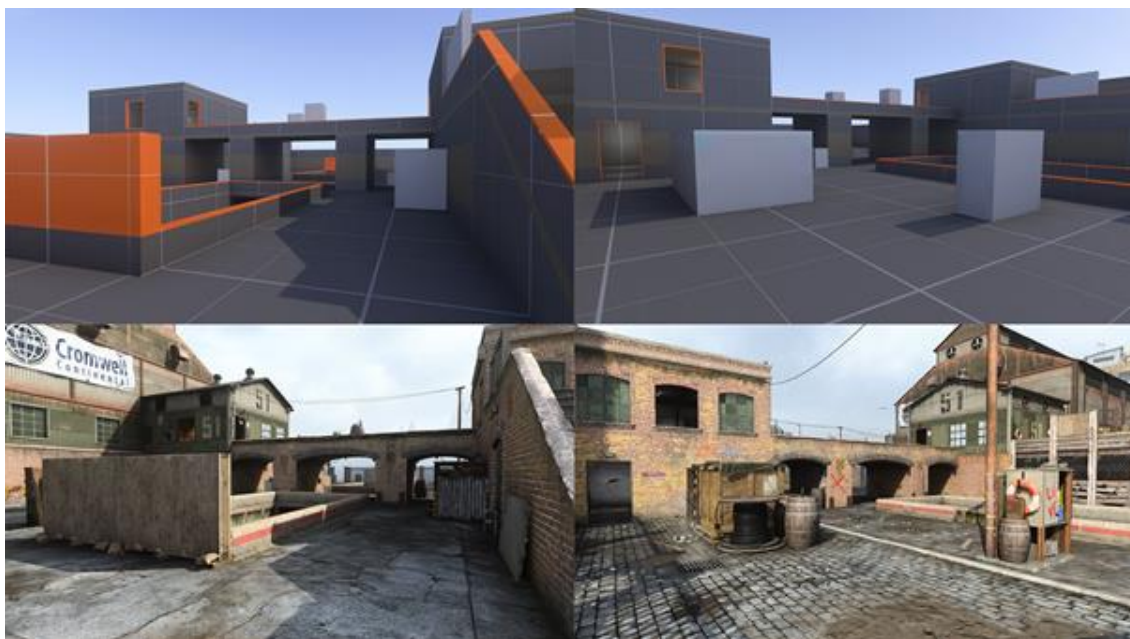
- **Nápad:** Obecná myšlenka, která slouží jako základní pilíř levelu.
- **Dokumentace:** Vytvoří se řádná dokumentace. U větších projektů či týmů se jedná konkrétně o tzv. Level Design Document (LDD).

LDD odpovídá například na otázky: Jaké jsou cíle? Jaké jsou výzvy, které hráč musí splnit? Existují speciální zóny nebo předměty, které hráč může vzít? Existují v levelu bossové?

LDD dále obsahuje vývojový diagram levelu a top down mapu. LDD se nesoustředí na relevantní věci (kde se co přesně nachází apod.), soustředí se pouze na důležité části (8Bit, 2023).

- **Blocking-out:** Je proces budování levelu pomocí jednoduchých geometrických tvarů. Je zaměřen na rozložení a celkovou funkčnost mapy.
- **Zakomponování ostatních assetů a funkcí:** Přidávání zvuku, zaměření více na vizuální stránku.
- **Playtesty:** Slouží ke shromažďování zpětné vazby a k následným úpravám levelu.
- **Finalizace:** Přidávání drobností, assetů, funkcí, které level příliš neovlivňují, ale mohou za výsledný lepší dojem z hraní (efekty, odrazy, nasvícení).

Obrázek 1: Blocking-out level (nahore) a hotový level (dole)



Zdroj: The Level Design Book (2023)

3.5.2 Herní assety

Herní assety nejsou nic jiného než předměty, které jsou ve hře vidět nebo slyšet. Například v akčních hrách to mohou být postavy, zbraně, stromy, budovy, speciální efekty, zvukové efekty (Unity Technologies, 2023).

Již při vývoji prototypu je zapotřebí využít herní assety, aby se dokázala vytvořit základní představa, jak hra bude vypadat. Existuje mnoho způsobů, jak je vytvořit. Finální řešení se odvíjí od složení týmu.

Existují tři základní možnosti, jak takové assety získat. První variantou je zakoupení assetů online, popřípadě stažení assetů nabízených zdarma. Použití těchto assetů většinou spadá pod určitou formu licence. Často je možné se setkat s licencí známou jako Creative Commons (CC) License „*Licence Creative Commons posilují pozici autora při rozhodování, za jakých podmínek bude dílo veřejně zpřístupněno. Licence Creative Commons fungují na jednoduchém principu: autor jejich prostřednictvím nabízí neurčitému počtu potenciálních uživatelů licenční smlouvu, na základě které jim poskytuje některá svá práva k dílu a jiná si vyhrazuje. Creative Commons nejsou popřením klasického pojetí autorského práva, jsou jeho nadstavbou a jako takové vycházejí z občanského zákoníku (§ 2358 – 2389 Zákona č. 89/2012 Sb., občanský zákoník).*“ (Creative Commons Česká republika, 2023b).

Prvky určující rozsah pravomocí: (Creative Commons Česká republika, 2023b)

- **Share** – Tento prvek je společný pro všechny typy CC licencí. Vyjadřuje, že licencované dílo je možné šířit, tzn. kopírovat, distribuovat a sdílet s veřejností.
- **Remix** – Opravňuje uživatele k pozměňování a úpravám díla. Umožňuje také celé licencované dílo nebo jeho část zpracovat s jiným dílem. Příkladem úprav může být úprava digitálních fotek, zhudebnění nebo remixování hudebních skladeb.

Podmínky, které je nutné respektovat při nakládání s dílem: (Creative Commons Česká republika, 2023b)

- **Attribution (BY)** – Je nutno uvést údaje o autorovi a dílu, a to způsobem, jaký autor stanovil. Při nespecifikování těchto údajů je nutné minimálně uvést jeho jméno (nebo pseudonym), název díla a odkaz na původní licenci Creative Commons. Pokud se šíří upravené dílo, je třeba také uvést způsob, jakým se dílo upravilo.
- **Share-Alike (SA)** – Pokud se licencované dílo jakkoliv upraví, je povinností výsledek práce vystavit pod stejnou licenci. Pokud se například upraví stažený 3D model, který je vystaven pod licenci CC BY-SA, je nutno pro tento 3D model použít stejný typ licence CC BY-SA.

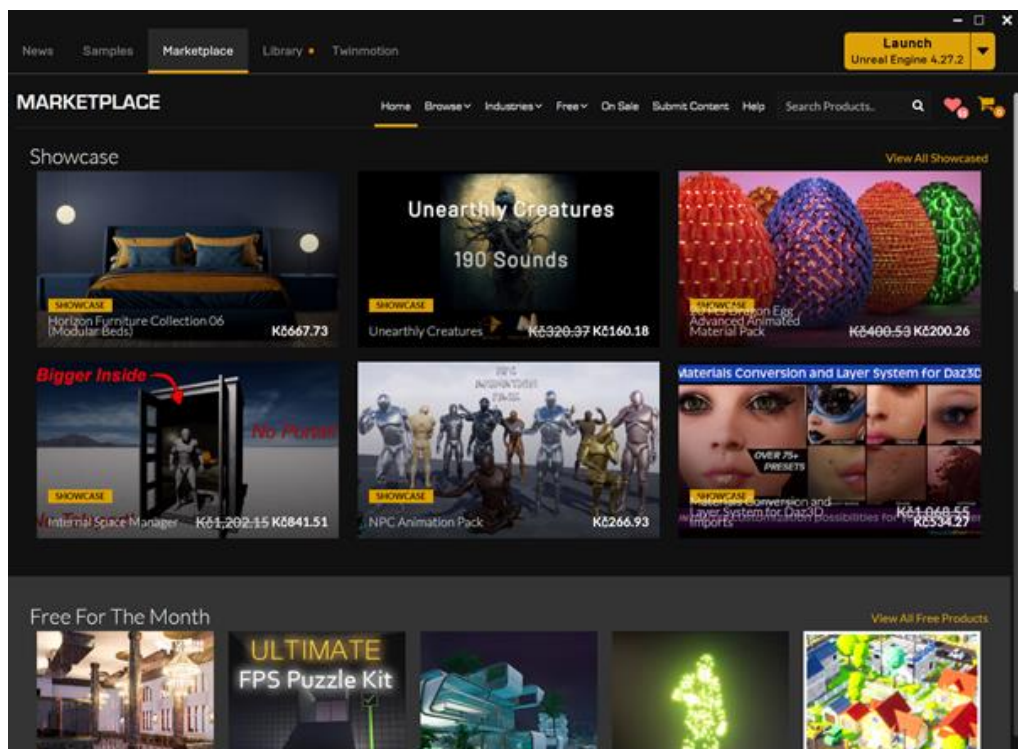
- **Non-Commercial (NC)** – Umožňuje nakládat s dílem pouze pro nekomerční účely – šíření díla nesmí generovat žádný finanční zisk.
- **No Derivative Works (ND)** – Zakazuje jakkoliv dílo upravovat. Jedná se o opak licenčního prvku “Remix“. Proto se v žádné licenci CC neobjevují tyto dva prvky společně.

Konkrétním příkladem kombinace CC licencí může být licence CC BY-NC, u které tvůrce assetu vyžaduje určité uvedení autorova jména a dílo, ve kterém je asset použit, nesmí vytvářet zisk. Stejný typ licence funguje i na ostatní díla (Creative Commons Česká republika, 2023b).

S tímto způsobem souvisí i možnost dát tvorbu assetů outsourcovat, tedy dát assety jako zakázku jednotlivci či firmě z daného oboru. Je ale nutné jako v předchozím případě brát zřetel na autorská práva a ověřit kvalitu autora.

Hojně využívaná jsou „nákupní centra“ herních enginů (Unreal Engine Marketplace). Jedná se o lehčí a jistější variantu. Valná většina produktů je vytvořena přímo pro daný engine, import do enginu je jednoduchý a je jistota, že se modely a obecně vše zakoupené dá v enginu využít bez jakéhokoliv omezení.

Obrázek 2: Unreal Engine Marketplace



Zdroj: Autor

Bohužel se i v těchto nákupních centrech vyskytují produkty, které nedosahují požadovaných kvalit, a proto je stále nutné před nákupem zkontrolovat recenze produktu, případně prověřit ostatní varianty nebo alternativy (Juegoadmin, 2021).

Časově náročnější variantou k získání assetů, oproti nakupování online, je vytváření assetů svépomocí. Nad assety je kontrola a dokáže se tak přímo ovlivnit jejich kvalita.

3.5.2.1 Princip tvorby 3D herních assetů

Indie vývojáři se nejčastěji zabývají vytvářením vlastních 3D assetů. Při tvorbě herních assetů existují určitá workflow, která se liší podle typu hry (2D/3D), druhu assetu (vegetace, 3D scany, běžné assety) a podle softwaru, který se používá.

U vegetace je pro indie vývojáře vhodnějším způsobem assety zakoupit, či využít specializovaný middleware (software, který zpracovává specializované aspekty jako grafiku, AI apod.), který se přímo zabývá jejím generováním. Tvorba vegetace, především té reálné, je velmi zdoluhavý proces a nezkušenému vývojáři by zbytečně prodlužovala vývoj. Nejznámějším generátorem je SpeedTree nebo jeho alternativa zdarma TreeIt.

Nejběžnější workflow při tvorbě kompletního 3D assetu je:

1. Získání referencí

3D modely jsou většinou inspirovány skutečnými objekty. Je proto užitečné získat si podklady, ze kterých se bude vycházet. Využívá se různých obrázků, ilustrací, fotografií nebo concept artů. I když by byl model zcela fiktivní, pomáhá si myšlenky sepsat a načrtnout (3D-Ace, 2021).

2. Vytvoření highpoly modelu

Jakmile je dostatečně vytvořena představa o tom, jak model bude vypadat, začíná samotné modelování v 3D softwaru. Mezi nejznámější patří Blender, Maya, 3DS Max nebo Cinema4D.

Highpoly (dále HP) model je model s vysokým počtem polygonů obsahující všechny detaily. Nedbá se zde na počet polygonů. HP model je vymodelován pomocí běžných *hard surface* technik za pomoci modifikátorů subdivision surface (vytváří složité vyhlazené povrchy s více polygony z jednoduššího méně segmentovaného modelu), zaoblování, boolean operace apod. Pro vytvoření HP modelu rovněž slouží sculpting (*sochařství*), využíváný především pro modelování organických objektů nebo i pro modelování technických detailů. Není výjimkou využívání 3D scanů a jejich následné úpravy. Hlavním cílem je, aby model vypadal co nejlépe a splňoval stanovená kritéria. Je vhodné předem

myslet na to, že mnoho detailů lze vytvořit při texturování. Například vyrytý text v kovu, rýhy nebo vzor dřeva. Náročnost a čas při modelování se tak zkrátí (Garofalo, 2022).

Výsledný model je vyexportován v preferovaném formátu, nejběžněji v *.fbx*. Před exportem jsou na všech částech modelu „aplikovány“ modifikátory (zaoblení, subdivision surface apod.). Je vhodné mít k dispozici kopii modelu bez aplikovaných modifikátorů. Kopii je možné použít při tvorbě *lowpoly* modelu a v určitých situacích dokáže urychlit práci (Seydamet, 2021).

3. Vytvoření *lowpoly* modelu

Lowpoly (dále LP) model je takový model, který je optimalizovaný pro použití v herních enginech. Cílem je dosáhnout co nejmenšího počtu polygonů a tím zmenšit výpočetní náročnost. Na LP model se zapékají detaily z HP modelu, a to konkrétně do *normal* textury. Dalšími texturami, které se mohou zapékat, jsou *AO* textura (Ambient Occlusion) nebo *High* textura (viz odd. 6. *Texturování*). Proto nemusí LP model obsahovat kompletní geometrii jako HP model. Jako příklad lze uvést zaoblenou hranu modelu. HP model obsahuje geometrii zaoblení. LP model tuto geometrii vynechá a zůstane mu jedna „ostrá“ hrana. Iluze zaoblení hrany je dosaženo *texture bakingem* (viz odd. 5. *Texture Baking*) (Seydamet, 2021).

Obrázek 3: Porovnání *High poly* a *Low poly*



Zdroj: Lei (2017)

Vytváření takového modelu je individuální záležitostí. Varianty mohou být následující: HP modelu postupně rozpustit hrany a tím zjednodušit geometrii, aplikovat modifikátory s menším nastavením segmentace či vytvořit zcela nový model.

Základní pravidla při tvorbě LP modelu: (On Mars 3D, 2021)

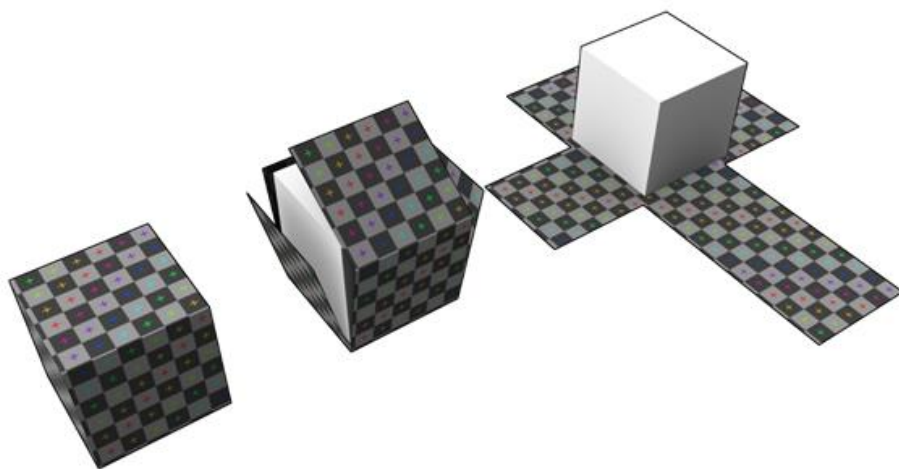
- Využít HP model co nejvíce je to možné. LP model se vytvoří mazáním hran, zaoblení, nebo aplikací modifikátorů s jinými parametry. Zajistí se tak zároveň i lepší výsledek u texture bakingu.
- Stejně jako u HP modelování panuje mýtus, že se musí striktně dodržovat čtyřhranné polygony. U LP modelu jsou trojúhelníkové polygony v naprostém pořádku a mnohdy jsou i vítány v rámci optimalizace.
- Pokud má zaoblení velký poloměr, je doporučeno zaoblení ponechat (případně mu snížit segmentaci). Ostrá hrana může v takovém případě (viz bod níže) způsobovat během texture bakingu artefakty.
- Čím více se základní tvar LP modelu přibližuje HP modelu, tím lepší je výsledek u texture bakingu. Může se využít tzv. Shrinkwrap modifikátor.
- Čím dál bude model od hráče, čím méně ho hráč uvidí a čím méně se na něj hráč soustředí, tím více geometrie lze z modelu odstranit.

4. UV mapping / unwrapping

UV mapping je proces, při kterém se přenáší 3D model, respektive síť polygonů 3D modelu (povrch 3D modelu), do 2D prostoru. Tento prostor má horizontální a vertikální osy označené písmeny U a V. Osy X, Y a Z jsou používány ve 3D prostoru. UV mapy se používají pro texturování (Pilgway, 2023).

Proces vytváření UV mapy se nazývá UV unwrapping (rozbalování), který se provádí pomocí 3D modelovacího nebo texturovacího programu. Během UV unwrappingu jsou vytvářeny na 3D modelu švy (Seams). Švy ve 3D fungují podobně jako je tomu u oblečení, kde šev představuje místo, kde je látka sešita dohromady. Šev při UV unwrappingu je část polygonové sítě 3D modelu (konkrétně hrany), která se rozdělí a model je pak možné reprezentovat ve 2D prostřednictvím UV mapy. Švy jsou označovány ručně nebo automaticky pomocí funkcí nabízené v 3D softwaru. Bez UV mapy by nebylo možné správně a efektivně texturovat. Kvalita UV mapy vede k usnadnění budoucího texture bakingu, k texturování nebo v enginech pro *Light Baking* (zapékání globální iluminace a světel do textury modelu) (Denham, 2023b).

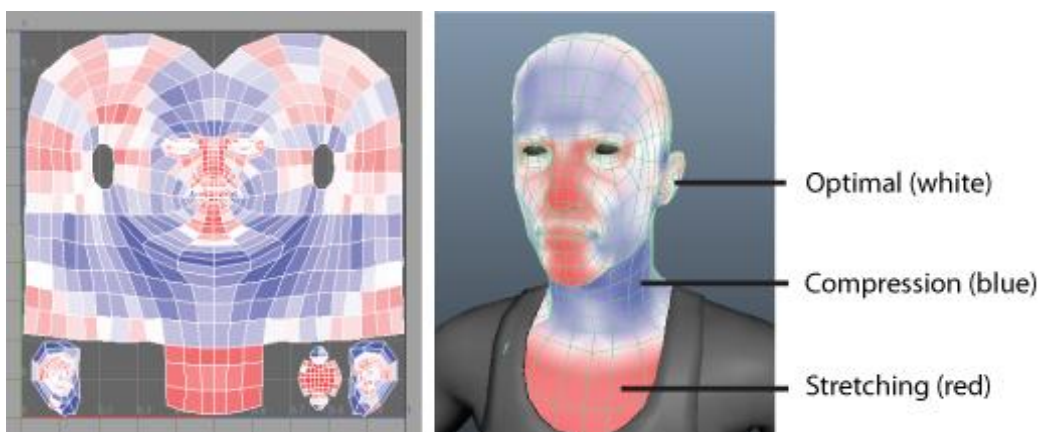
Obrázek 4: Rozložení modelu do 2D prostoru



Zdroj: Wikipedia (2022)

Během unwrappingu dochází k různým kompromisům, kdy je snaha dosáhnout co nejmenšího počtu švů, aby zároveň nedošlo k deformaci povrchu. Deformací je myšleno, jak moc se rozložené polygony na UV mapě liší od skutečného rozložení u 3D modelu. Takováto deformace se objevuje nejčastěji na zaoblených nebo organických modelech.

Obrázek 5: Deformace UV mapy



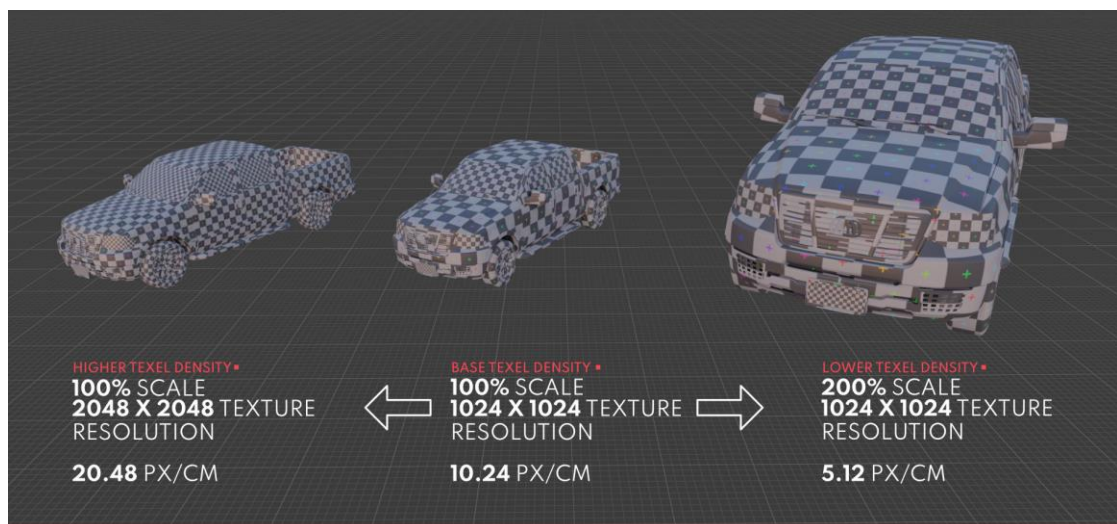
Zdroj: Autodesk (2023)

Pravidla při tvoření švů: (Denham, 2023b; NVIDIA, 2023)

- Švy vytvořit podél ostrých okrajů, nejčastěji ty, které mají méně než 90 stupňů.
- Švy umístit na místa, kde nebudou vidět (kde se nachází vlasy, oblečení, kam hráč neuvidí).
- Švy by měly kopírovat přirozenou strukturu objektu.
- Kontrolovat, zda se nepřekrývají Islands (UV island je spojená skupina polygonů oddělena od jiných skupin polygonů v UV mapě). Překrývání není vhodné u texture bakingu, u vytváření lightmap nebo při využití specifického shaderu (program běžící na grafické kartě). Ovšem v některých případech se překrývající Islands použijí (např. pro tiling textury).

Během unwrappingu modelu se řeší tzv. *Texel Density* (TEXEL - TEXture pixEL). Texel density (*TD*) je závislá na velikosti 3D objektu a rozlišení textury. Obvykle se měří jako pixely/metry nebo pixely/cm. Například 512px/m znamená, že na každý metr ve hře bude mít textura 512 pixelů. Pokud by byla zeď o velikosti 1x1m, textura by měla rozlišení 512×512. Pokud by zeď měla rozměry 2x2m, pak by textura musela mít rozlišení 1024×1024, aby měl každý metr čtvereční rozlišení 512×512 (Dries, 2022).

Obrázek 6: *Texel Density*



Zdroj: Dries (2022)

Texel Density se kontroluje pro každý island v UV mapě zvlášť. Zabraňuje se tedy tomu, aby došlo k nekonstatnímu rozlišení textury na 3D modelu i napříč mezi modely.

Ne všechny 3D programy defaultně nabízejí nástroje pro kontrolu *TD*. Velmi často lze najít tyto nástroje ve formě pluginů/addonů. Snadnou a velmi rychlou kontrolu *TD* nabízí

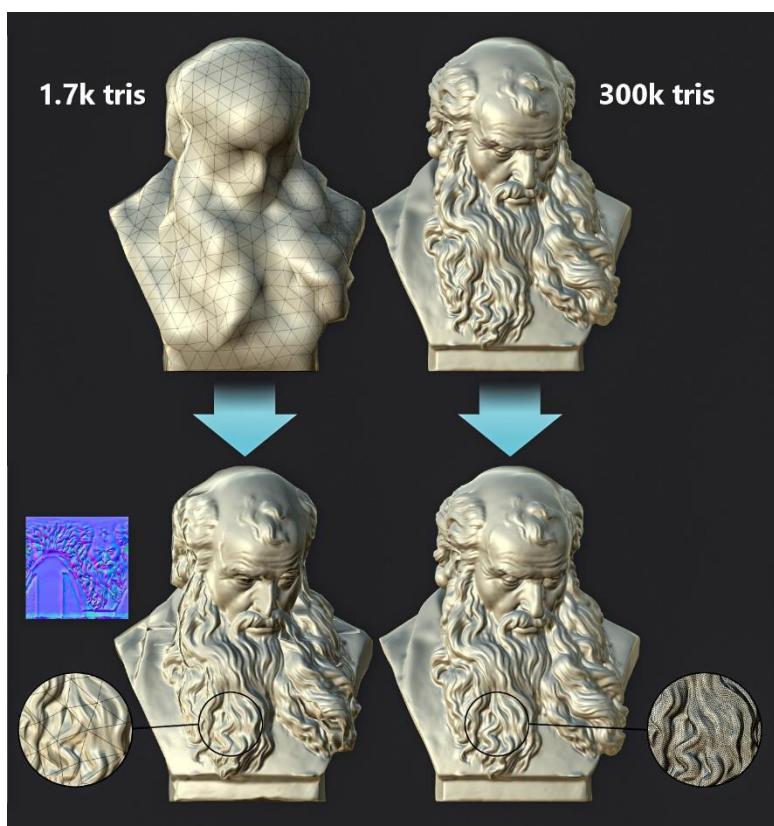
například RizomUV. Při absenci těchto nástrojů lze na model v 3D programu aplikovat texel density texturu, a tím vizuálně kontrolovat rozlišení jednotlivých částí UV mapy.

Unreal Engine 5 obsahuje *UV Editor*, který poskytuje sadu nástrojů pro vytváření a úpravu UV. V tuto chvíli ale nenabízí pokročilé funkce jako mají 3D programy. Mezi další možnosti tohoto editoru patří *Slučování hran*, *UDIM management* a *Multi-Asset workflow* (Epic Games, 2023).

5. Texture Baking

Neboli zapékání textur, je proces přenosu dat z jednoho 3D modelu na jiný 3D model. V tomto případě jsou informace z prvního modelu přeneseny na UV druhého modelu, a poté jsou uloženy do textury. Jednoduše řečeno, model s vysokým počtem polygonů (HP model) je převeden na model s nízkým počtem polygonů (LP model). Důvodem, proč se texture baking používá je, že lowpoly model obsahuje méně dat. Díky tomu má menší velikost souboru a také je méně výpočetně náročný, což vede k vyšší snímkové frekvenci ve hře (VNTANA, 2021).

Obrázek 7: Texture baking



Zdroj: Fra3point (2022)

Zapékání textur lze využít nejen v realtime aplikacích, ale i při běžném renderu scény. Mnoho 3D programů (např. Marmoset, Blender) nebo přímo 3D Texture painting programy (např. Substance Painter), nyní nabízí nástroje, které umožňují texture baking provést jednoduše a pomáhají celý proces urychlit.

V programu Substance Painter se zapečou detaily na lowpoly model v podobě *normal* textury. Dále se mohou generovat textury jako *ambient occlusion*, *thickness*, *curvature* apod. Tyto textury slouží především pro správné fungování *Smart materiálů* (smart material – inteligentní materiál přizpůsobující se podkladovému povrchu a geometrii) při 3D texturování.

6. Texturování

Texturování hraje v počítačové grafice důležitou roli a při tvorbě herních assetů tomu není jinak. Textury mohou velmi věrohodně simulovat detaily ve vysokém rozlišení, a to bez vysokého počtu polygonů. Díky nim může jednoduchý model vypadat komplexně (Trammel, 2023).

U texturování 3D modelů se nejčastěji využívá tzv. PBR textur (physical based rendering texture – textury uchovávající informace o povrchu a barvách). Soubor těchto textur tvoří tzv. materiál, který je aplikován na 3D model. Jednotlivé textury slouží k nastavení různých vlastností materiálu. Určují, jak materiál reaguje na světlo, případně jakým způsobem materiál vystupuje do prostoru. Tento způsob vytváření materiálů poskytuje vysokou úroveň realismu a konzistenci při jakémkoli typu osvětlení. Použití PBR materiálů se rozšířilo také popularizací naskenovaných textur z reálného světa (Mesquita, 2021).

Nejčastěji používané druhy textur: (Mesquita, 2021a; 2021b; 2021c)

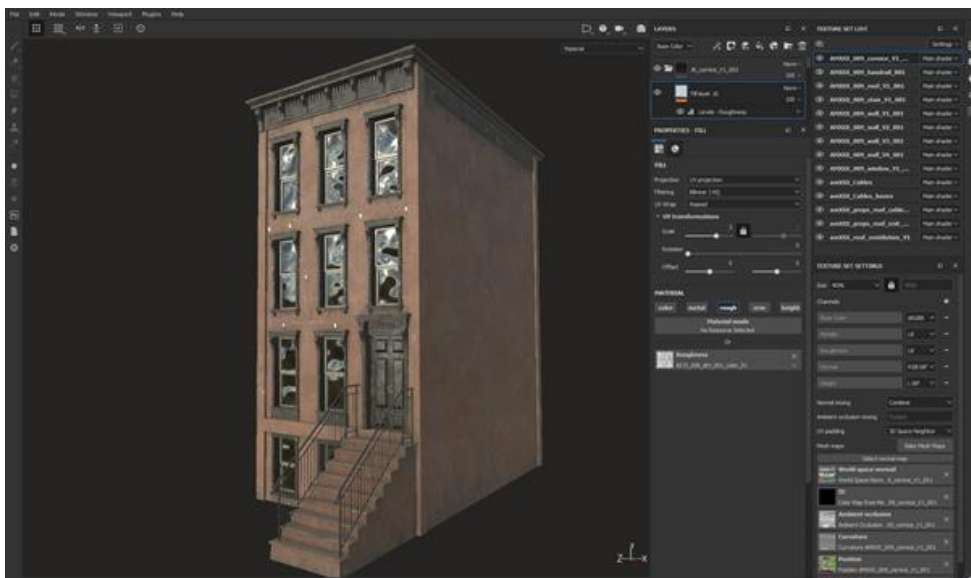
- **Color / Diffuse** – Definuje barvu materiálu. Jedná se textury zahrnující světlo a stíny při pořízení (např. po vyfocení kůry stromu).
- **Albedo** – Definuje barvu materiálu. Narozdíl od Diffuse ale neobsahuje informaci o světle nebo stínech a jedná se tak pouze o čistou barvu povrchu. Albedo textura je vhodnější pro PBR materiály.
- **Normal** – Simuluje interakci světla s povrchem. Vytváří tak iluzi detailů (hrboly, rýhy, zaoblení, škrábance) na povrchu materiálu, aniž by se změnila jeho geometrie.
- **Roughness** – Řídí ostrost odrazů tím, že určuje, jak je světlo šířeno po povrchu. Drsnější povrchy světlo rozptylují do více směrů, což má za následek

rozmazanější odrazy. Rozsah hodnot je 0.0 až 1.0. Pokud je drsnost rovna 0.0, jsou odrazy ostré. Pokud je drsnost 1.0, jsou odrazy rozmazané.

- **Metalness / Metallic** – Určuje, které části povrchu mají být považovány za kovové a které za nekovové. Funguje na podobném principu jako maska. Tam, kde je černá barva, je nekovová část povrchu. Kde je bílá barva, tam se nachází kovová část povrchu.
- **Displacement** – Na rozdíl od normal texture se nejedná pouze o iluzi detailů, ale skutečně posouvá geometrii do prostoru. Nevytváří však nové polygony, pouze posouvá polygony, které již objekt má. Pro použití této texture se v herních enginech využívá tzv. teselace sloužící pro dynamické zvýšení počtu polygonů na objektu.
- **Height map** – Má stejnou funkci jako Displacement. Odlišují se pouze použitím. Height textura obsahuje méně detailů a využívá se tak například u generování terénu, u kterého detaily nejsou nutné.
- **Ambient Occlusion** – Je textura ve stupních šedi. Používá se k vytvoření „umělého“ jemného stínování na nerovnostech povrchu. Lze ji tak použít ke ztmavení drobných stínů, například v místech kontaktu mezi objekty.

Model je nejčastěji texturován v 3D texture painting programech. Jedním z nejrozšířenějších je Substance Painter, který má mnoho užitečných funkcí jako například zapékání textur v rozlišení 8K, real time PBR workflow a mnoho tzv. pre-set materiálů (předvytvořené materiály, které nabídnou širokou škálu základních vlastností různých povrchů). Postup texturování je podobný běžné práci v Adobe Photoshop a dalším podobným programům. Zásadním rozdílem je možnost pracovat ve 3D prostoru přímo s modelem. To znamená, že se textury “malují” nejen ve 2D prostoru do UV mapy, ale také ve 3D prostoru na model, což umožňuje vidět výsledný vzhled modelu v reálném čase. Substance Painter zobrazuje a exportuje všechny textury ve formátu PBR, proto se dají importovat přímo do herního engine s naprosto stejným výsledkem, jaký je vidět v softwaru (Denham, 2023a).

Obrázek 8: Substance Painter



Zdroj: Franczak (2019)

U některých modelů je výhodné využít generické texturování přímo v engine. (např. zdi, podlahy, cesty).

V Unreal Engineu je možné bezplatně využívat knihovnu *Megascans* a nástroje *Bridge* a *Mixer* od firmy *Quixel*, kterou Epic Games v roce 2019 zakoupilo. *Megascans* je rozsáhlá knihovna nabízející fotorealistické PBR skeny povrchů, vegetace a 3D objektů. Unreal Engine 5 má tuto knihovnu přímo zabudovanou prostřednictvím pluginu *Quixel Bridge*. *Quixel Bridge* slouží pro snadný přístup k *Megascans*. Je k dostání i jako samostatná aplikace. *Quixel Mixer* je aplikace zdarma sloužící pro 3D texturování (zdarma pouze pokud jsou *Megascans* použity v rámci Unreal Engine nebo po zaplacení) (Waqar, 2018).

Unreal Engine využívá již zmíněné materiály. Materiály se v engine nevytvářejí pomocí kódu, ale prostřednictvím vizuálního skriptování v *Material editoru*. *Material editor* je grafické rozhraní založené na tzv. *nodech* (node – uzel). Umožňuje tak vytvářet *shadery*, jež lze aplikovat na geometrii (*static mesh*, *skeletal mesh* atd.) (Epic Games, 2023).

3.5.3 Animace

Odvětví animací ve videohrách je stále populárnější. S přibývajícími nároky ze strany uživatelů, kteří si dnes již téměř zvykli na důkladné a precizní ztvárnění hry, se řadí stále více k důležitějším nutnostem zhotovit animace, které splňují ta nejvyšší očekávání.

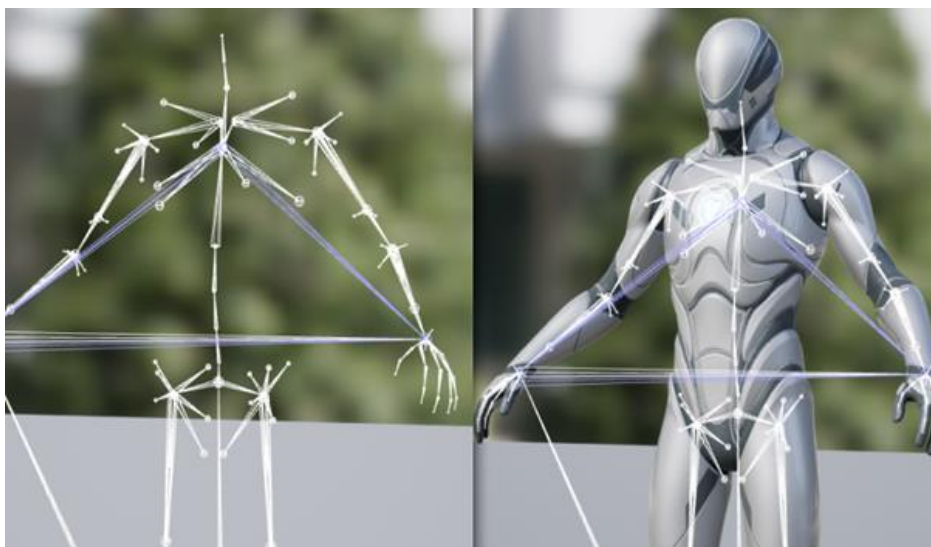
Animace jsou vytvářeny za pomoci různých programů, od jednoduchých, jako je Microsoft PowerPoint, až po specializované 3D modelovací programy, jako je Autodesk Maya, Blender nebo Cascadeur pro využití AI k simulování fyziky.

Konečný produkt zahrnuje scény, předrenderované animace, které běží před nebo po dokončení určitého levelu nebo animace ve hře používané pro pohyb při samotném hraní hry (Darvideo, 2021).

Animace využívané přímo při hraní hry mají určitá omezení. Nemohou využívat modifikátory jako u klasického renderování v 3D softwaru. Proto se u těchto animací spoléhá na klasické *skeletal* animace. U skeletal animací má objekt nebo postava svou kostru a může se s ní tak manipulovat stejně jako se skutečnou. Každá kost (bone) se ovládá pomocí animace, fyziky, nebo inverzní kinematiky. Každá kost má k sobě díky tzv. *weight paintingu* přiřazené polygony (weight painting – proces, při kterém se určí, na jaké polygony a s jakou intenzitou má kost vliv). Herní enginey nyní nabízí různé nástroje pro tvorbu procedurálních animací (Valve Corporation, 2022a).

Mimo skeletal animace je možné ve videohrách využívat tzv. *Morph target* animace (Morph target je též označováno jako „blend shapes“ nebo „shape keys“). Morph target animace je uložena jako řada pozic vertexů samotného modelu, nikoliv pozice „kostí“. V každém klíčovém snímku animace jsou vertexy modelu přesunuty do jiné pozice. V herních enginech je možné Morph targety aktivovat v průběhu hry dynamicky. S Morph targety je možné se setkat například u animací obličeje postavy (Epic Games, 2023; DigitalRune, 2016).

Obrázek 9: Unreal Engine Skeleton



Zdroj: Epic Games (2023)

Unreal Engine nabízí pokročilé animační nástroje kombinující již zmíněné skeletal animace a morph target animace. Pomocí *Skeletal Mesh Animation System* se upravují charaktery, kostry (skeletal) a animace. Za pomoci nástrojů jako *Blend Spaces*, *Animation Blueprints* a *State Machines* je možné vytvářet různé animace reagující na input hráče nebo na prostředí.

Předem vytvořené filmové a animované sekvence je možné vytvořit nástrojem *Sequencer*, který umožňuje vytvářet průlety levellem, animaci postav a mnoho dalšího.

Unreal Engine dále obsahuje nástroj *Control Rig*. Control Rig je sadou animačních nástrojů, které umožňují upravovat a animovat postavy. Je tak možné obejít nutnost rigování a animování v externích programech, a namísto toho animovat přímo v editoru *Unreal Editor* (Epic Games, 2023).

3.5.4 Audio

Pojem audio zahrnuje vše, co je ve hře slyšet. Může se jednat o:

- Zvukové efekty
- Dialogy
- Hudbu
- Zvuky v UI
- Zvuky okolí
- Voice-overs (hlasové nahrávky)

Nejjistější variantou, jak audio získat a zakomponovat do hry, je najmutí zvukového designéra (sound designéra). Ovšem indie vývojáři většinou nemají prostředky na to, aby si takovou externí pomoc mohli dovolit. Proto se uchylují k jiným možnostem.

První možností je najít na internetu již existující zvukové efekty nebo knihovny efektů. Existuje nespočet webových stránek nabízejících zvukové efekty (např. *boomlibrary.com*, *freesound.org*, *mixkit.co*, *productioncrate.com*, *sonniss.com*). Nevýhodou tohoto řešení je, že je mnohdy složité najít takový zvuk, který přesně vyhovuje požadavkům (efekt nemá správný tón, hlasitost nebo je příliš krátký). S tím je spojeno i to, že jsou efekty nahrány jiným zařízením nebo v jiném prostředí. To ovšem neznamená, že není možné nalézt kvalitní efekty.

Časově náročnější alternativou k již vytvořeným zvukům, je nahrání svých vlastních zvuků nebo ruchů (v angličtině *foley*). Foley je umění, ve kterém se zvukové efekty tvoří pomocí různých běžných předmětů (lámání celeru, házení příboru na zem apod.). Další

zvuky je možné vytvořit kombinací vlastních nahraných zvuků nebo těch pořízených. Investice do nahrávání vlastních zvuků nemusí být vysoká. Stačí mít k dispozici mobil, externí mikrofon a aplikaci pro záznam zvuku (Bridgett, 2017).

Unreal Engine 5 nabízí renderer zvuku *Audio Mixer*, který rozšiřuje původní renderer z Unreal Engine 4. Audio Mixer obsahuje funkce starého rendereru *Sound Cues*, *Sound Classes* a *Sound Mixes*. Nově přidanými funkcemi jsou *MetaSounds*, *Audio Modulation*, *Quartz* a *Audio Analysis* (Epic Games, 2023).

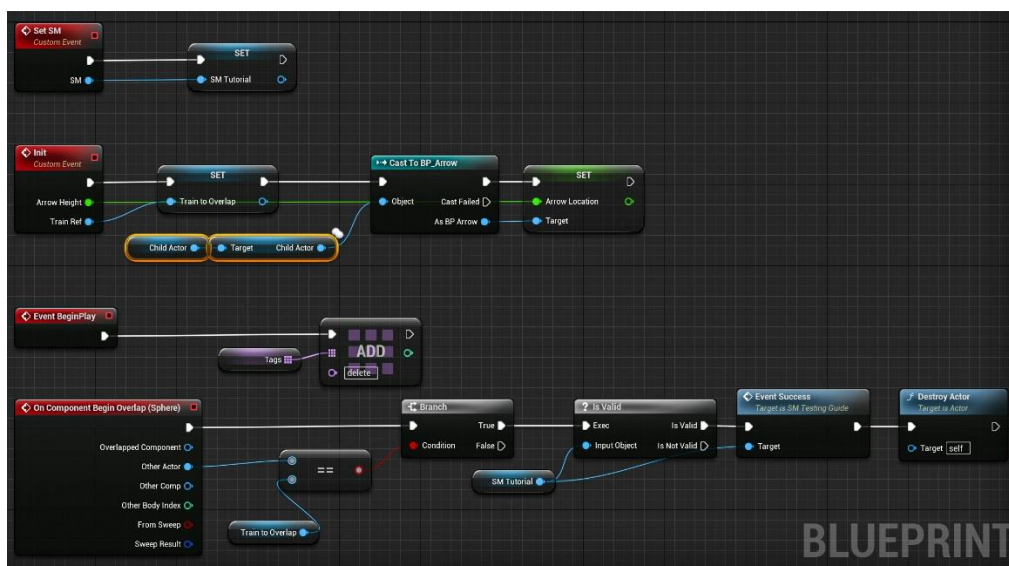
3.5.5 Programování a skriptování

I navzdory rozhodnutí použít pro vývoj hry již existující herní engine, programování a skriptování zabere v průběhu vývoje hry spoustu času a vyžádá si tak spoustu práce. Vytvářejí se prototypy, začleňují se nové funkce, mechaniky a opravují se chyby, které se v průběhu vývoje objevily.

Unreal Engine nabízí k vytváření herních prvků dvě metody, Blueprints a C++.

První metodou je Blueprints, což je vizuální skriptovací jazyk založený na tzv. nodech (buňka představující část kódu). Stejně jako mnoho jiných běžných skriptovacích jazyků, i Blueprints slouží k definování objektově orientovaných tříd a objektů v engine. Blueprints systém je velmi flexibilní. Poskytuje designérům možnost využívat prakticky celou škálu konceptů a nástrojů, které jsou obvykle dostupné pouze programátorům. Designéři díky Blueprint editoru dokáží i bez větších zkušeností s programováním rozšiřovat funkce nebo logiku hry.

Obrázek 10: Unreal Engine Blueprint



Zdroj: Autor

Ovšem největší síla tkví v kombinaci C++ a Blueprintů. Přestože je možné vytvořit celou hru za pomoci jen jedné z těchto možností, nejedná se o vhodné řešení. Běžnou praxí je, že programátor pomocí C++ ve zvoleném IDE (Visual Studio, JetBrains Rider) vytváří základní třídy (děděné podle potřeby z tříd jako *APawn*, *AActor*, *UObject*), které obsahují sadu základních funkcí, metod a vlastností. Programátor také předem počítá s tím, že se třída bude dále rozšiřovat. Zpřístupňuje proto určité metody a proměnné (použitím maker jako například *UPROPERTY* nebo *UFUNCTION*), se kterými bude moci designér nebo programátor pomocí *Blueprint editoru* pracovat (Epic Games, 2023).

3.5.6 Version Control

Česky správa verzí, nebo také verzování, slouží pro komplexní správu rozsáhlých projektů. Verzování funguje tak, že jsou k dispozici starší verze dat a souborů. V případě potřeby je možné vrátit se v čase a získat zpět původní verzi dat (365tipu, 2017).

Ať hru vyvíjí velké studio nebo indie vývojář, stále se jedná o rozsáhlý projekt, který zabere mnoho času a ve kterém se tak provede v průběhu vývoje mnoho změn. Z tohoto důvodu jsou pro verzování využívány specializované systémy.

Specializované systémy sloužící pro verzování, jsou softwary, které umožňují v průběhu času spravovat změny zdrojového kódu týmům i jednotlivcům. Verzovací software sleduje každou změnu kódu ve speciálním druhu databáze. Pokud dojde k nějaké chybě, může se vývojář vrátit v čase a porovnat dřívější verze kódu řádek po řádku.

Verzovací systémy dále umožňují, aby každý vývojář mohl v týmu pracovat na svém úkolu bez závislosti na jiném vývojáři. Jeden člen týmu tak pracuje například na nové funkci, zatímco jiný člen týmu opravuje chybu. Verzování týmům pomáhá řešit problémy tím, že sleduje každou jednotlivou změnu každého člena týmu a pomáhá zabránit souběžné práci a konfliktu (například, jestliže dva vývojáři provádí změny ve stejném souboru na stejném řádku) (Atlassian, 2023).

Nejoblíbenějším a nejrozšířenějším verzovacím systémem mezi programátory je Git. Důvodem toho je jeho open source kód a popularita GitHubu, cloudového nástroje pro ukládání Git repozitářů. Open source umožňuje uživatelům Git provozovat s nulovými nebo velmi nízkými náklady a nabízí obrovský ekosystém nástrojů, cloudových poskytovatelů (Bitbucket, Azure DevOps) a desktopových aplikací.

Git lze velmi dobře použít i při práci na projektu v Unreal Engine. Musí se ale počítat s tím, že neposkytuje robustní systém pro uzamykání souborů (tzv. *locking system*).

Uzamčení souboru zajišťuje, že s ním může pracovat pouze vývojář, který uzamčení souboru provedl. To zabrání konfliktům, kdy více lidí odešle různé verze stejného souboru. Toto je využívané především u binárních souborů, které nelze snadno sloučit. Uzamykání souborů systémem Git (což je vlastnost jeho modulu LFS) není příliš spolehlivé a desktopové aplikace pro uzamykání souborů velmi často nenabízí podporu.

Alternativou, kterou doporučuje Unreal Engine dokumentace, je využití Perforce. Perforce je proprietární verzovací systém, populární především pro vývoj AAA her. Je centralizovaný, což znamená, že je uživatel vždy připojen k serveru. Perforce lze ovládat přes příkazovou řádku, desktop aplikaci a je dokonce i přímo integrován v Unreal Engine. Perforce oproti Gitu lépe pracuje s uzamykáním souborů a lépe zvládá práci s binárními soubory. Dalšími plusy je jeho stabilita a pluginy do IDE (Visual Studio a Rider). Nevýhodami Perforce jsou naopak vyšší cena, malá komunita, málo informací a návodů a nutnost provozovat server (Niedoba, 2023).

3.6 User Experience

User Experience (UX - uživatelská zkušenost) je podle oficiální normy ISO 9241-210: 2010 definována jako „*Vnímání a reakce člověka vyplývající z použití nebo předpokládaného použití produktu, systému nebo služby.*“. Je to široký pojem, který zahrnuje všechny aspekty interakce koncového uživatele s produkty nebo službami. UX říká, jak dobře se uživatel v produktu orientuje, jak snadno se produkt používá nebo jak relevantní je zobrazovaný obsah. UX se řeší téměř v každém odvětví od webů, aplikací, forem placení až po automaty nebo panely aut a letadel (Norman, 2023).

3.7 Testování použitelnosti

Testování použitelnosti znamená hodnocení produktu nebo služby na základě testování s uživateli. Účastníci testování se obvykle snaží splnit předem stanovené úkoly. Přitom jsou sledováni moderátory, kteří je poslouchají a dělají si poznámky. Testování má za úkol shromáždit kvalitativní a kvantitativní údaje, identifikovat případné problémy s použitelností a určit spokojenost účastníků s produktem (GSA, 2023).

Metody testování použitelnosti lze obecně zařadit do tří kategorií:

- Kvalitativní nebo kvantitativní testování
- Moderované nebo nemoderované testování
- Vzdálené nebo osobní testování

3.7.1 Kvalitativní / kvantitativní

Veškerý prováděný uživatelský výzkum se dělí na kvalitativní nebo kvantitativní. Kvalitativní a kvantitativní testování použitelnosti se odlišuje daty, se kterými pracuje. Kvalitativní výzkum přímo posuzuje použitelnost systému. Výzkumníci pozorují, jak účastníci řeší konkrétní prvky systému a následně vyvozují, které aspekty aktuálního návrhu jsou problematické, a které fungují dobře. Vždy mohou účastníkům testování klást doplňující otázky, aby získali vhled do konkrétního problému, se kterým se účastník testování setkává. Zjišťují se odpovědi na otázku "proč?" – např. proč uživatelé neklikají na tlačítko „zpět“, nebo proč nevyužívají určitá nastavení.

Kvantitativní výzkum je základem pro srovnávání nového návrhu s referenčním bodem, tedy se známým standardem nebo s použitelností konkurenčního či předchozího návrhu. Zaměřuje se na čísla a statistiky pro zodpovězení otázek "co?", "kde?" a "kdy?" (např. časy dokončení úkolu, míra úspěšnosti, počet chyb, hodnocení spokojenosti).

Kvantitativní údaje nepoukazují na to, s jakými problémy se uživatelé setkali a neřeknou, jaké změny v návrhu provést, aby se dosáhlo lepšího výsledku. Proto jsou běžně kvantitativní data doplněna o data kvalitativní (Budiou, 2017).

3.7.2 Moderované / nemoderované

Moderované testování použitelnosti je testování prováděné v reálném čase vyžadující aktivní účast moderátora (člověk, který testování řídí a usnadňuje). Jedná se o rozhovor s někým, kdo testuje daný produkt. Nejlepší možností je provádět moderované testování v laboratoři nebo v pracovním prostředí, ovšem lze využít i videokonferenci.

Během nemoderovaného testování účastníci plní předem stanovené úkoly bez moderátora a jsou z pravidla ve svém zvoleném prostředí. Tato technika se více využívá u testování specifického scénáře nebo konkrétní části produktu. U tohoto testování není příležitost klást podrobnější otázky ve spojitosti se specifickými akcemi, které uživatel provedl (Schade, 2013).

3.7.3 Osobní / vzdálené

Osobně vedené rozhovory s účastníky testování se provádějí v prostředí laboratoře. Jedná se zpravidla o konverzaci vedenou moderátorem. Během osobního testování lze lépe pozorovat zejména chování a výrazy účastníků. Je také možné lépe odhadnout vhodný čas, kdy položit doplňující otázky. To umožňuje výzkumníkovi lépe kontrolovat tempo testování.

Vzdálené testování provádí uživatel sám, například ve své domácnosti. Účastník během testování odpovídá na předem stanovené otázky a plní stanovené úkoly. Domácnost (nebo jakékoliv prostředí, které si uživatel sám zvolí) napomáhá k větší uvolněnosti účastníka během testu. Díky tomu je možné získat realističtější zpětnou vazbu. Vzdálené testování je též dobrou variantou ve chvíli, kdy je obtížné sehnat jednotlivé respondenty (Sumeet, 2020).

3.7.4 System Usability Scale (SUS)

System Usability Scale (SUS) používá ke stanovení úrovně použitelnosti mezinárodně standardizovaný dotazník skládajícího se ze souboru deseti otázek, kterými respondent vyjadřuje míru souhlasu na stupnici 1-5, konkrétně od „Rozhodně souhlasím“ po „Rozhodně nesouhlasím“ (Bangor a kol., 2008).

3.7.5 Cognitive think-aloud protocol (scénáře)

Účastníci při plnění daného scénáře během testování verbalizují své myšlenky. Díky tomu je možné získat vhled do jejich myšlenkových pochodů, který je pouhým pozorováním obtížné získat (Fan a kol., 2020). Tato metoda byla použita v mnoha studiích, které se snaží porozumět kognitivním procesům uživatele během interakce s technologiemi (Tawfik a kol., 2022). Je doporučeno provádět testování pouze za přítomnosti uživatele a moderátora. Uživatel postupuje podle zadaného scénáře a moderátor zasahuje pouze v případě, že uživatel nemohl nebo by nebyl schopen scénář úspěšně dokončit.

Dle Nielsen (2012) a Virziho (1992) je třeba pěti testerů k identifikaci 80 % systémových chyb.

3.8 Post-production fáze

Post-production fáze přichází po vydání hry. Proces vývoje hry tak stále pokračuje. Šance, že hra bude vydána bez chyb, je velmi malá. Proto se v počátku převážně zaměřuje na údržbu hry, na opravy bugů a optimalizaci, aby hráčům neklesl herní zážitek.

Přidávání nového obsahu udržuje hráče v napětí a přivádí je ke hře i po jejím dohrání. V konkurenčním herním prostředí je toto nezbytná činnost. Proto se dále vytváří bonusový obsah, například ve formě DLC nebo v rámci časově omezených eventů, aby se zvýšila znovuhratelnost a aktivita hráčů ve hře (Pickell, 2019; MasterClass, 2021).

3.9 Komerční nasazení hry vytvořené v Unreal Engine

Jestliže je v plánu hru, nebo jiný obsah vytvořený v Unreal Engine, publikovat a zpeněžit, Epic Games vyžaduje, aby se provedly určité kroky: (Epic Games, 2023)

Oznámení o vydání hry

To se děje vyplněním krátkého formuláře obsahujícího základní informace o vydávaném produktu, jako například název, jméno vývojáře, kontaktní osoby, webovou stránku produktu, kdy bude hra pravděpodobně vydána atd.

Royalty payment and tracking

Po dosažení hrubých příjmů ve výši 1 milionu USD (počítá se pro každý produkt zvlášť), přichází povinnost platit licenční poplatky ve výši 5%. Podrobnější informace jsou k nalezení na následující stránce: www.unrealengine.com/en-US/eula/unreal

3.10 Vydání hry na platformu Steam

Existují desítky digitálních distribučních platform, na které jsou herní aplikace publikovány. Mnohé z nich mají svá omezení, například Origin nebo Uplay jsou přístupné pouze určitým studiím. Většina indie herních vývojářů volí mezi platformami Steam a Itch.io anebo Google Play či Apple Store.

Steam je místem, kam se vzhledem ke své popularitě indie hra dostane většinou jako první. S popularitou Steamu však přichází i velká konkurence, což vede k větší náročnosti ohledně toho, jak hra musí zaujmout a čím vyčnívá (Algadrie, 2021).

Samotné publikování na Steam je vcelku rychlý a jednoduchý proces, na který je ale potřeba se předem připravit. Před samotnou registrací a vytvořením vývojářského účtu je nutné si předem stanovit, zda hru bude vydávat fyzická, či právnická osoba. Tyto údaje se budou při registraci vyžadovat a bez nich nelze pokračovat dál. Déle je nutné počítat s nutností zaplatit 100 USD jako poplatek ve Steam Direct (kapitola 3.10.2.3).

3.10.1 Registrace

Prvním krokem u publikování hry na platformě Steam je vytvoření Steam vývojářského účtu. Registruje se do tzv. *Steamworks Distribution Program* na webové stránce <https://partner.steamgames.com>. K registraci lze využít již existující Steam účet, který je výhodnější ve chvíli, kdy si chce vývojář vybudovat značku na své jméno. Ve většině případů je lepší variantou vytvořit zcela nový vývojářský účet.

Velice důležitým údajem při vytváření vývojářského účtu je *jméno/název*. Ten se musí shodovat se jménem/názvem uvedeným v bankovních údajích, které se budou vyplňovat později. Jméno vývojáře a vydavatele lze později u jednotlivých publikovaných her upravit. Název účtu zůstává. Další důležitou kolonkou k vyplnění je *právní forma společnosti* (Valve Corporation, 2023b; 2023c).

3.10.2 Vyžadované informace

Po vytvoření vývojářského účtu se vyplňují některé důležité informace, konkrétně platební údaje a daňové informace. Bez nich nemůže vývojářský účet fungovat. Dále se platí již zmíněný Steam Direct poplatek, který činí 100 USD za jednotlivý vydávaný produkt (Valve Corporation, 2023b; 2023c).

3.10.2.1 Platební údaje

Aby Steam věděl, kam vyplácet částky z prodejů vydaného produktu, bude potřebovat platební údaje včetně identifikačního čísla banky, čísla bankovního účtu a adresy banky.

Jméno či název majitele bankovního účtu se musí shodovat se jménem/názvem subjektu uvedeným výše (*jméno/název účtu*). Při absenci bankovního účtu nelze v publikaci hry pokračovat (Valve Corporation, 2023b; Boylls, 2022).

3.10.2.2 Daňové informace

U vyplňování daňových informací velmi záleží, jestli hru vydává jednotlivec (živnostník) nebo společnost a také na umístění společnosti. Steam vyžaduje vyplnění formuláře, tzv. interview. Provádí se prostřednictvím *tax identity servise*.

Potřebné informace odpovídají tomu, co se nachází ve formuláři W-9 pro občany USA a ve formuláři W-8BEN pro občany nebo společnosti se sídlem v zemích, které mají s USA uzavřenou smlouvu o zamezení dvojího zdanění. Ke správnému vyplnění těchto dokumentů je vhodné využít odborné pomoci.

Při práci s těmito dokumenty přichází zmínka o TIN (Taxpayer Identification Number). Ověření TIN může nějakou dobu trvat, protože je prováděno speciálním subjektem. V některých případech může tento proces trvat až 30 dní (Valve Corporation, 2023b; Xsolla, 2023).

3.10.2.3 Steam Greenlight vs. Steam Direct

Před aktuálně fungující službou Steam Direct fungoval několik let Steam Greenlight, který společnost Valve zavedla, aby pomohla nezávislým vývojářům prodávat jejich hry. Steam Greenlight fungoval tak, že vývojář zaplatil společnosti Valve 100 USD za účast v procesu Greenlight. Vývojáři stačilo zaplatit tento poplatek pouze jednou a na službu Steam Greenlight mohl nahrát libovolný počet titulů. Tyto peníze nebylo možné získat zpět.

Po zaplacení poplatku byl vývojář připojen do Steam Greenlight, kde si vytvořil stránku Greenlight s informacemi o hře (název a popis), nahrál video, trailery a obrazovky, odkazy (například na demoverzi hry) a odkazy na sociální síť.

Jakmile byla stránka hry připravena, uživatelé hlasováním "Ano", "Ne" nebo "Možná" vyjadřovali, jak se jim hry líbí. Kromě toho mohli stránku sledovat, sdílet, zanechávat komentáře, číst příspěvky na blogu a číst oznámení tvůrce stránky. Všechny hry na Greenlight byly seřazeny podle počtu hlasů "Ano". Jakmile se hra na Steam Greenlight probjovala do první stovky, byla považována za potenciálně v budoucnu úspěšnou. Při probjování se do prvních pěti míst hra získala "Greenlit by the community.", vývojář dostal přístup k backendu Steamu a také hru na Steamu mohl vydat.

Bohužel byl tento přístup na podporu vývojářů velkým neúspěchem – na každou úspěšnou hru připadaly desítky špatně vytvořených a obecně nežádoucích titulů, z nichž mnohé nebyly hotové nebo byly poskládané z předem zakoupených assetů.

Společnost Valve proto představila kompletní náhradu za Greenlight - "Steam Direct". Ve Steam Direct se nyní platí 100 USD za hru. Už se tedy nejedná o jednorázový poplatek. Těchto 100 USD je možné získat nazpět, a to konkrétně v případě, kdy hra na Steamu vydělá 1000 dolarů a více.

Po kompletním vytvoření vývojářského účtu a odeslání poplatku za hru má vývojář přístup k funkcím Steamu a může pracovat na vydání hry (Crider, 2017; Perez, 2017).

3.10.3 Nahrání aplikace na Steam

Každá aplikace má na Steam Store svoji stránku, má své komunitní centrum a své unikátní ID. Tuto aplikaci vidí zákazník ve své knihovně. Aplikace může být vytvořena pouze po zaplacení poplatku Steam Direct (Valve Corporation, 2023c).

Steam nabízí pět typů aplikací:

Tabulka 1: Typy aplikací na platformě Steam

Game	Všechny hry včetně VR
Software	Produkty pro tvorbu zvuku, designu, ilustrací, úpravy fotografií
DLC	Stáhnutelný obsah pro již vytvořenou hru.
Video, Series and Episode	Produkty jako jsou filmy, videa, návody
Demo	Steam nabízí možnost poskytovat bezplatné demo verze her zákazníkům, aby si ji mohli před zakoupením vyzkoušet.

Zdroj: VALVE CORPORATION (2023c)

3.10.3.1 Depot

Depot je logické seskupení souborů, které jsou zákazníkovi doručeny jako jedna skupina. Je na vývojáři, jaká data Depot bude obsahovat. Obvykle Depot obsahuje danou aplikaci. Více Depotů lze využít ve chvíli, kdy bude aplikace vydávána na více OS. Aplikace pro Windows a MacOS tak bude mít dva Depoty, kde každý bude obsahovat jeden build hry pro daný systém (Depot (1) – obsahuje build hry pro systém Windows, Depot (2) - obsahuje build hry pro systém MacOS). Každý Depot je jednoznačně identifikován unikátním ID (Valve Corporation, 2023c).

3.10.3.2 Build

Build je výsledkem nahrání obsahu. Build je kolekce Depotů a jedná se tak v podstatě o jednotlivé verze hry (Valve Corporation, 2023c).

3.10.3.3 Branch

Neboli větev, definuje, jaký build hry zákazník dostane. Defaultně má každá hra pouze jednu větev. Dodatečně se mohou přidat další větve. Například větev, která obsahuje novější verzi hry a je přístupná pouze pro vybrané hráče (Valve Corporation, 2023c).

3.10.3.4 Package

Package říká, jak zákazník získá přístup do hry. Je to sada jedné nebo více aplikací, která se prodává na Store Page hry. Nejběžnějším způsobem je zakoupení hry zákazníkem ze Store Page. Z pohledu Package zákazník zakoupí defaultní Package obsahující aplikaci hry. Je možné vytvořit více takovýchto „balíčků“. Příkladem může být situace, kdy je na stránce k zakoupení defaultní Package se základní hrou a Package obsahující jak základní hru, tak i všechna DLC, a to za zvýhodněnou cenu (Valve Corporation, 2023c).

3.10.4 Store Page

Store Page lze vytvořit pouze po dokončení registrace a zaplacení Steam Direct poplatku pro danou hru. Store Page se skládá z několika záložek: (Valve Corporation, 2023c; Xsolla, 2023; Boylls, 2022)

- **Basic Info:** Jedná se o dlouhý formulář, kde se uvádí název hry, vývojáři, vydavatelé, odkazy na sociální média a externí webové stránky, klíčová slova, podporované jazyky, žánry, informace multiplayeru, informace o DRM, právní informace, specifikace a požadavky na PC, datum vydání a také kontaktní údaje na podporu dané hry.
- **Description:** Uvádí se dlouhý a krátký popis hry. Dlouhý popis je o dané hře. Na krátký popis se vztahuje omezení počtu znaků, konkrétně pouze 300 znaků. Zvláštní pozornost by se měla věnovat krátkému popisu, protože ho hráč vidí jako první. Zobrazuje se v horní části stránky obchodu. Popis by měl hráčům přiblížit, co budou ve hře dělat, o čem hra je, jaká bude hrátelnost a měl by vystihnout atmosféru hry.

Do dlouhého popisu je možné vložit bannery. Jedná se o další poutavý prvek pro potenciální hráče.

Služba Steam umožňuje stránky obchodů lokalizovat do různých jazyků, což zvyšuje návštěvnost dané hry. Je proto výhodné stránku obchodu lokalizovat, i když hra není v daném jazyce dostupná. K dispozici je také sekce pro recenze a ocenění.

- **Ratings:** Poskytnutí informací o ratingu není povinné. Pokud vydávaná hra obdržela oficiální hodnocení obsahu od ratingové agentury (tj. ESRB, PEGI, BBFC atd.), mělo by to být uvedeno. Bez tohoto oficiálního hodnocení se nic neuvádí. Může se také přidat věková hranice, aby se omezil prodej nezletilým.

- **Early Access:** Pokud je v plánu umožnit předběžný přístup do hry, vyplňují se v dotazníku *Early Access Q&A* informace, proč se využívá předběžný přístup, jak dlouho bude trvat, jak se bude lišit od plné verze, případný rozdíl v ceně mezi předběžným přístupem a plnou verzí a také aktuální stav hry vydávané v předběžném přístupu.
- **Trailers:** Před spuštěním stránky obchodu se musí přidat alespoň jeden trailer na hru. Trailer by neměl být příliš dlouhý a měl by především demonstrovat hratelnost. Maximální počet trailerů však není omezen a tím i možnost přidat „filmověji“ pojatý trailer zůstává. Trailer by měl mít poměr stran 16:9, rozlišení 1920x1080 a měl by mít vysokou přenosovou rychlost 5000 kb/s nebo vyšší. Může mít 30 nebo 60 snímků za sekundu. Měl by být také ve formátu MOV nebo WMV.
- **Special Settings:** Zde je možno přidat nástroje jako Google Analytics. Také je možno uvést informace o obsahu ke stažení.
- **Graphical Assets:** Zde se vkládají obrázky představující hru. Obrázkům jsou předem stanoveny různé rozměry podle toho, kde budou zobrazeny. V dokumentaci Steamworks je podrobně a srozumitelně popsáno, jak mají jednotlivé assety vypadat, k čemu slouží a jaká jsou konkrétní pravidla pro jejich vložení na Steam. Rozdělují se na:

1. Store Assets

Tabulka 2: Store Assets

Header Capsule	Povinné	460px × 215px – Pouze logo a grafika hry.
Small Capsule	Povinné	231px × 87px – Pouze logo a grafika vaší hry.
Main Capsule	Povinné	616px × 353px – Pouze logo a grafika vaší hry.
Vertical Capsule	Povinné	374px × 448px – Pouze logo a grafika vaší hry.
Screenshots	Povinné	1920px × 1080px nebo jiný preferovaný poměr stran 16:9. Podporována je jakákoliv velikost.

Page Background	Volitelné	1438px × 810px – Decentní (“jemná”) grafika. Neměla by být příliš světlá.
Bundle Images	Volitelné	Povinné, pokud je nastaven “bundle”

Zdroj: VALVE CORPORATION (2023c)

2. Community Assets

Tabulka 3: Community Assets

Community Icon	Povinné	32px × 32px – Malé logo hry nebo reprezentativní ikona (v aplikaci Steam).
Client Icon	Povinné	32px × 32px .ico – Pouze logo a grafika hry. Slouží jako zástupce na ploše.
Community Small Capsule	Povinné	184px × 69px – Logo hry a grafika. Stejně jako Small Capsule.

Zdroj: VALVE CORPORATION (2023c)

3. Library Assets

Tabulka 4: Library Assets

Library Capsule	Povinné	600px × 900px – Pouze logo a grafika hry.
Library Hero	Povinné	3840px × 1240px .png – Pouze logo a grafika hry.
Library Logo	Povinné	1280px × 720px .png – Pouze logo hry. Zobrazí se v horní části “Hero Graphic”.

Zdroj: VALVE CORPORATION (2023c)

4. Event Assets

Tabulka 5: Event Assets

Event Cover	Povinné	800px × 450px – Grafika a marketing pro nějakou událost nebo oznámení.
Event header	Volitelné	1920px wide × 622px – Grafika a logo události.
Library Spotlight	Volitelné	2108px wide × 460px – Banner upozorňující na událost nebo oznámení.

Zdroj: VALVE CORPORATION (2023c)

3.10.5 Nastavení ceny

Za správné nastavení ceny odpovídá každý partner platformy sám. Cena se nastavuje ve všech podporovaných měnách. Steam nabízí možnost nastavit všechny tyto ceny pomocí doporučení, které vygeneruje pro každou měnu ze základní ceny stanovenou v USD.

Společnost Valve kontroluje prvotní cenu a všechny její změny. Zpracování běžně trvá 1-2 dny. Během této kontroly se také zjišťuje, jestli nedošlo k neúmyslné chybě ze strany partnera. Ne vždy se ale taková chyba zachytí (Valve Corporation, 2023c).

3.10.6 DRM

DRM (Digital rights management) je způsob ochrany autorských práv digitálních médií. Zahrnuje použití technologie, která omezuje kopírování a prodej děl a softwaru chráněných autorskými právy. Například DRM zabráňuje vypálení hry na CD pro další sdílení nebo předprodej.

Většina herních společností DRM používá k ověření nákupu hry. Steam využívá své DRM fungující na principu ověření vlastnictví hry. Steam dále zajišťuje správné fungování funkcí Steamworks tím, že před spuštěním hry spustí službu Steam. Steam DRM se aplikuje na build hry přes tzv. *Steam DRM Wrapper*. Aplikace DRM se provádí na každý nový build hry (Valve Corporation, 2023c).

4 Vlastní práce

V této části práce bude popsán průběh vývoje základního prototypu herní aplikace v Unreal Engine 5. Při vývoji hry je postupováno za pomoci vývojových fází představených v teoretické části.

4.1 Návrh hry

Při návrhu hry se pracovalo s několika možnostmi, na jaké téma hru vytvořit a do jakého žánru ji zasadit. Pro příklad se pracovalo s návrhy na western FPS příběhovou hru či western Turret Shooter. Finálním rozhodnutím je však žánr RTS z alternativní historie. Prvotním vnuknutím tohoto nápadu byla hra Locoland (dříve Steamland) vydaná v roce 2003 (dostupné z: <https://store.steampowered.com/app/352130/Locoland/>). Po mnoha nápadech, jak by se hra mohla jmenovat, bylo její jméno stanoveno na Hell on Wheels.

Hell on Wheels je RTS s možností kampaně, bitev proti AI nebo hráčům. Hráč přebírá kontrolu nad obrněnými vlaky, inspirovanými těmi z první světové války. K dispozici má továrny a servisní depa, kde staví vlastní kompozice vlaků. Každý typ vagonu má svoji roli v boji. Těžká děla mají daleký dostřel a destruktivní účinek, ovšem za cenu omezení pohybu vlaku. Nejtěžší zbraně vyžadují, aby vlak stál. Zbraně menších kalibrů umožňují hráči, aby vlaky bojovaly za pohybu. K udržení vojenské mašinerie musí hráč získávat zdroje. Ty získává obsazováním strategických bodů na mapě a jejich držením. Vlaky dále vyžadují, aby je hráč udržoval bojeschopné. Mají omezené množství munice a v průběhu bitev se mohou poškodit. Pro tyto případy hráč musí poslat vlak do servisního depa, kde se opraví a doplní zásoby munice. Hráč může obětovat palebnou sílu vlaku za muniční vozy sloužící ke zvýšení množství munice a delší bojeschopnosti vlaku.

4.2 Game Design Document

Po vzoru kapitol *Pre-production fáze* a *Project management* byl vytvořen Game Design Document. K detailnímu textovému dokumentu se nepřístupilo, a to proto, aby se usnadnilo provádění změn. Proto byla zvolena aplikace Trello sloužící jako jakýsi sumář nápadů, mechanik, odkazů, konceptů apod. Hlavními důvody, proč je Trello v tomto projektu využíváno, je jeho přehlednost, vizualizace a agilní přístup. Vše je možno „za pochodu“ jednoduše doplňovat, nebo měnit. Dalšími faktory, proč bylo Trello vybráno, je jeho jednoduché a intuitivní prostředí. Trello má také zcela dostatečný plán zdarma, kde je

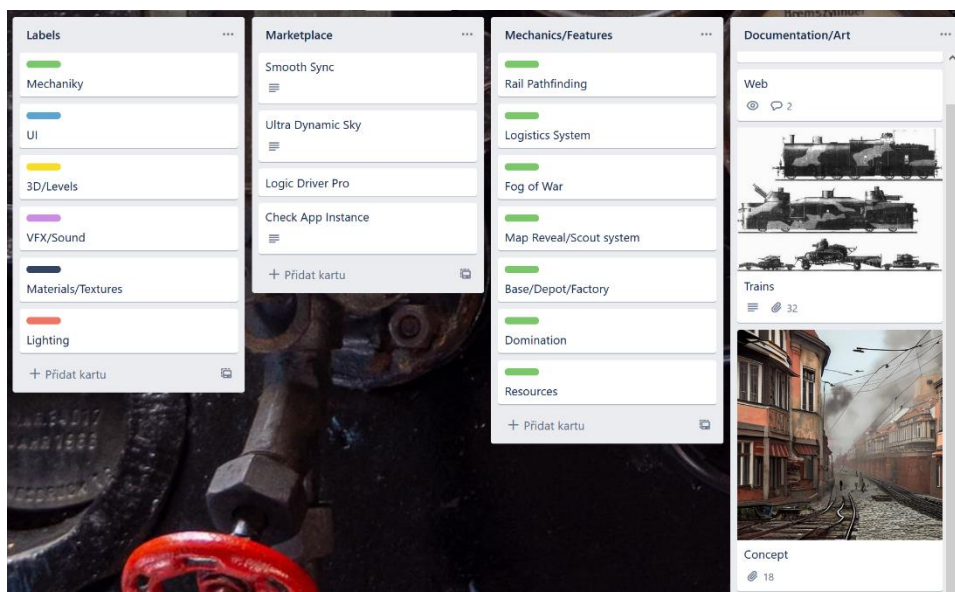
z pohledu autora jediným nedostatkem maximální velikost každého vkládaného souboru 10 Megabytů.

Na Trello nástěnce byly vytvořeny čtyři základní sloupce, které pokrývají základní informace a popis prototypu:

- **Labels** – Slouží pro větší přehlednost. Karty v ostatních sloupcích mohou dostávat jednotlivé štítky (barvy) podle toho co obsahují.
- **Marketplace** – Jaké pluginy se využívají nebo budou využívat.
- **Mechanics/Features** – Slouží pro souhrn základních mechanik.
- **Documentation/Art** – Obsahuje odkazy na různé weby, koncepty a vše, co při vývoji může pomoci.

Podoba nástěnky není finální. V průběhu projektu stále budou přibývat nové karty a sloupce.

Obrázek 11: Trello - GDD projektu



Zdroj: Autor

4.3 Project management

Jako primární nástroj pro project management byl zvolen GitLab a to na základě předchozí autorovi zkušenosti s tímto nástrojem. Alternativou by mohla být například mnohem běžněji využívaná aplikace Jira, která by plnila funkci stejným způsobem.

GitLab umožňuje používat agilní postupy od základního sledování problémů až po řízení projektů ve stylu Scrum a Kanban. Zároveň umožňuje využívat správu Git repozitářů.

V tomto projektu se Git repozitář a obecně verzování projektu v GitLabu nebude využívat a GitLab tak slouží pouze jako nástroj pro project management.

GitLab nevyužívá klasické pojmenování Scrum artefaktů. Pojmenování Scrumu a GitLabu lze ovšem dát do vzájemné spojitosti. Příkladem pojmenování může být:

Tabulka 6: Porovnání pojmenování Scrum vs. GitLab

Scrum	GitLab
Product backlog	Issue lists
Sprint	Milestone
Agile board	Issue board

Zdroj: Wu (2018)

Během projektu nebyl využit Scrum jako takový, protože je určený především pro agilní týmy, nikoliv pro jednotlivce. Byly zužitkovány klíčové funkce a “artefakty” GitLabu, které byly aplikovány pro potřeby autora (viz body níže).

Klíčovými funkcemi GitLabu jsou:

- **Projects (projekty)** – Projekty slouží k organizaci práce a pro správu Git repozitářů. Je to nejširší kategorizace v Gitlabu. Autorův projekt se jmenuje Hell on Wheels.
- **Issues** – Issues jsou v GitLabu základní jednotkou práce. Ve výchozím nastavení se používají ke sledování chyb, ale lze je použít ke sledování čehokoliv, co je třeba udělat. Ke každé Issue je možné připojit osoby, termíny plnění, váhu, poznámky a další.

V autorově projektu jsou Issues použity pro vše, na čem se pracuje nebo se má pracovat a zaznamenávají všechny nové úkoly nebo nalezené chyby. Issues mají zásluhu na tom, že na nic není zapomenuto. Konkrétnější specifikaci, o čem Issue jedná, udávají Labels.

- **Labels (štítky)** – Usnadňují sledování problémů a projektů. Labels umožňují flexibilně označovat práci, což umožňuje snadno sledovat a vyhledávat položky, které nás zajímají. Pomocí Labels je na první pohled hned jasné, na čem se aktuálně pracuje, co je vytvářeno a co má prioritu. Nestává se tak, že je rozpracovaných příliš mnoho Issues a soustředí se tak na důležité funkce.

V projektu bylo vytvořeno 15 Labels pokrývající všechny situace, které během vývoje mohou nastat. Jsou jimi:

Tabulka 7: GitLab Labels v projektu prototypu

As Designed	Funkce by se měla chovat jinak.
Blocker	Fatální problém. Pád hry, zaseknutí, nefunkčnost kritických funkcí
Bug	Pro filtraci chyb a problémů. Něco nefunguje tak jak má.
Critical	Něco, co má velký dopad na herní zážitek (FPS drop)
Development	Pro třídění rozpracovaných funkcí.
Duplicate	Zdali je dané issue duplikát již existujícího issue.
Fixed	Fixed znamená, že problém byl otestován a tester nebo jiný člen týmu potvrdil jeho opravu.
In Progress	Probíhá proces vývoje/probíhá ladění.
Is Test	Testování údajně opravené chyby/problému.
Major	Problém má velký vliv na pocit hráčů ze hry. Ničí atmosféru a způsobuje, že hratelnost je nevyrovnaná.
Minor	Hra je schopna běžet, ale není dobré danou issue vidět (světelný glitch).
Ready	Opravený problém. Připraveno k testování.

To Do	Problém, o který se musí postarat vývojový tým.
Trivial	Problém není důležitý a má malý dopad na hru nebo na zážitek a pocit hráčů.
Wont Fixed	Prohlášení, že problém nebo chyba ve hře není opravitelná a bude ignorována.

Zdroj: Autor

- **Board (nástěnka)** – Nástěnka poskytuje větší přehled o problémech a o průběhu projektu. Aktuálně je vytvořena jedna nástěnka Development sloužící pro filtraci Issues, které byly uzavřeny, a které jsou se štítky Development a Bug.
- **Milestones (milníky)** – slouží ke sledování projektu. Lze je použít k seskupení problémů a požadavků do jedné časové osy. Milníky také umožňují nastavit termíny plnění a osoby. V projektu Milestones představují jednotlivé verze hry.

4.4 Asety

Asety, které jsou použity v projektu, jsou kombinací assetů zdarma z Unreal Engine Marketplace, Megascans assetů a assetů vytvořených autorem v programech Blender, Rizom UV a Substance Painter. Postup vytváření assetu byl u každého z nich stejný. Autor zvolil jako příklad model Lokomotivy.

4.4.1 Modelování

Prvním vymodelovaným assetem pro prototyp hry byla parní lokomotiva. Podkladem pro modelování sloužila polská obrněná parní lokomotiva TI-3 z první čtvrtiny 20. století.

Modelování v Blenderu započalo vložení referenčního obrázku lokomotivy, kde je možné vidět lokomotivu z boku. Díky tomu bylo možné snadno vytvořit základní tvar a proporce lokomotivy.

Jednotlivé části lokomotivy jsou modelovány rozdílně v závislosti na jejich velikosti a tvaru. U větších částí, jako například trup vlaku, byl vymodelován zjednodušený model s menším počtem polygonů pro stanovení základního tvaru a následně byl aplikován modifikátor Subdivision Surface. U menších, nebo jednodušších částí, jako jsou poklopy apod., se využila především kombinace modifikátorů Bavel (Zaoblení) s různými parametry.

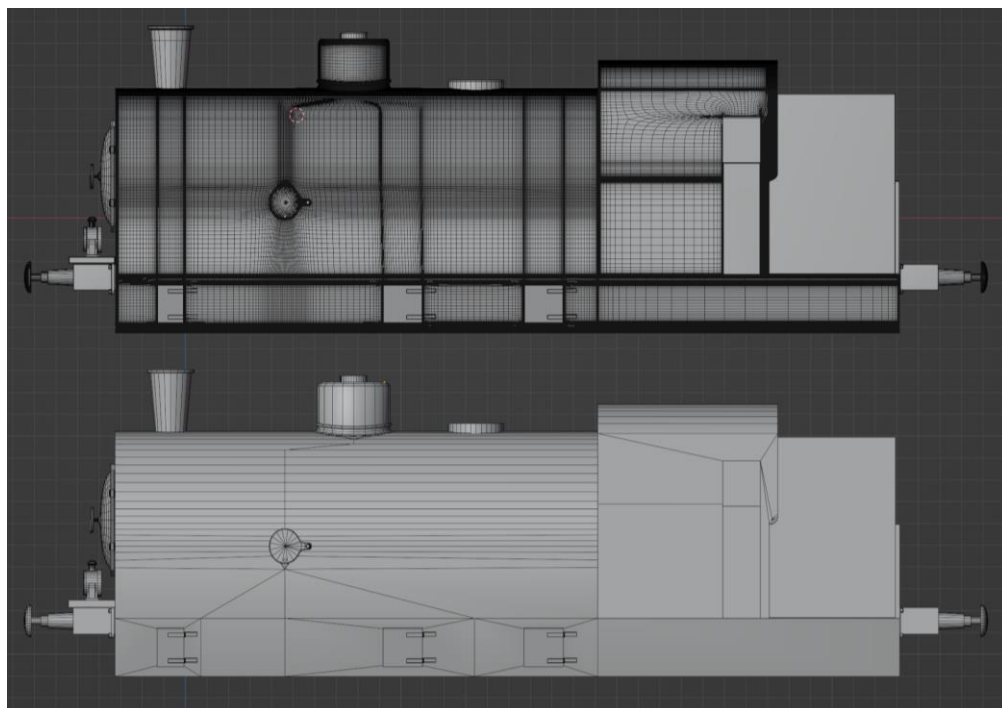
4.4.2 Příprava modelu na UV mapping

Výslednému High Poly (HP) modelu byla vytvořena kopie, aby bylo možné vytvořit Low Poly (LP) model. Autor pracoval s HP i LP modely v jednom projektu, protože mohl lépe porovnávat tvar LP modelu vůči HP modelu a zachytit tím nežádoucí rozdíly v geometrii.

U všech částí LP modelu byly odstraněny modifikátory přidávající geometrii. Dále Autor na modelech odstranil všechny pomocné hrany sloužící původně pro Subdivision Surface. Všude tam, kde to bylo možné, byly změněny čtvercové polygony na trojúhelníkové polygony za účelem jednoduchosti modelu a došlo se tak k co nejmenšímu počtu polygonů. U částí LP modelu, kde se zásadně lišila geometrie od HP modelu, především Modely, na kterých byl Subdivision Surface modifikátor, se tento modifikátor aplikoval a následným rozpouštěním hran se model postupně zjednodušil. Částem LP modelu, u kterých Subdivision Surface modifikátor nebyl původně přiřazen a zároveň se zásadně lišila geometrie od HP modelu, byla tato geometrie vytvořena ručně (například zaoblená hrana).

HP model bylo nutné připravit pro export pro budoucí texture baking. Proto se na všech HP částech modelu aplikovaly modifikátory upravující geometrii. Blender nabízí při exportu funkci “Apply modifiers”, který by tuto činnost mohl zautomatizovat. Autor tuto funkci nevyužil z důvodu, aby nedošlo k nechtěným špatným výsledkům, například pokud je špatně nastavená osa pro zrcadlení modelu. Autor zároveň mohl při ručním aplikování modifikátorů výsledný model zkontrolovat.

Obrázek 12: Blender – High poly (nahore) vs. Low poly model lokomotivy



Zdroj: Autor

4.4.3 UV mapování

UV mapování bylo provedeno v programu Rizom UV, který se zaměřuje čistě na UV unwrapping. Před samotným unwrappingem, byly v programu Blender všechny LP modely označeny a byla vybrána možnost exportování do formátu FBX (*File / Export / .fbx*), kde se dále vybrala možnost exportu *selected objects*.

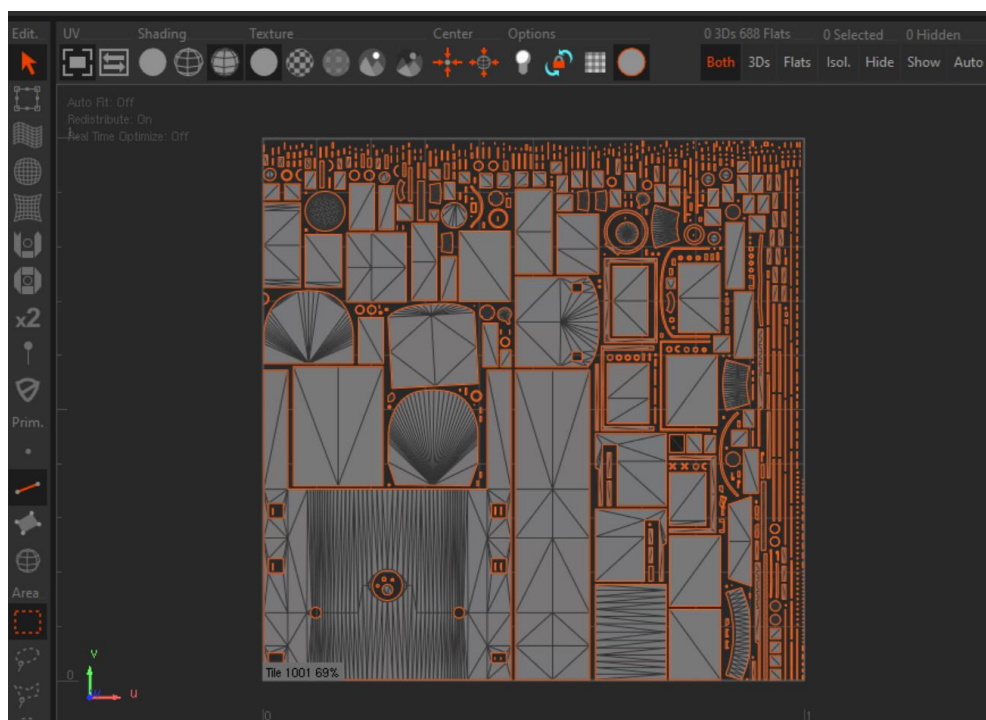
FBX soubor byl importován do programu Rizom UV. Postup UV unwrappingu, kterým se autor řídil, by se dal shrnout takto:

1. Byly vybrány všechny Islands (režim výběru Islands – zkratka F4) pomocí CTRL+A. Jelikož UV mapa modelů nebyla v Blenderu generována, označují se celé jednotlivé modely, čímž se usnadňuje výběr.
2. Klávesou CTRL + levým kliknutím byla odznačena část modelu, kterou Autor vybral pro unwrapping.
3. Klávesovou zkratkou “H” byly vybrané modely schovány. Zůstal tak pouze jeden nebo více modelů, na které se autor soustředil.
4. Vybíráním hran a klávesovou zkratkou “C” byly vytvořeny švy, případně klávesou “W” byly švy zrušeny.

5. Po vytvoření švů byly označeny všechny Islands modelu a posloupností příkazů Unfold (U), O (Optimize) a P (Pack) byla vytvořena UV mapa pro daný model.
6. Klávesou “Y” byly zobrazeny schované modely a celý proces se opakoval od bodu 1.

Po vytvoření UV map pro všechny části lokomotivy byly označeny Islands pro aplikování texel density. Islands se tak zvětšily nebo zmenšily na stejný poměr vůči modelu. Následně byly Islands klávesou P rozmístěny do UV mapy tak, aby co nejefektivněji zabíraly celý prostor. Model bylo následně možné uložit. (RizomUV neukládá soubor ve svém vlastním formátu, ale pracuje přímo s FBX souborem). Posledním krokem bylo model importovat zpět do Blenderu.

Obrázek 13: RizomUV – UV mapa lokomotivy



Zdroj: Autor

4.4.4 Texturování

Před samotným texturováním bylo nutné provést další úpravy modelů v Blenderu, konkrétně pojmenování. HP modelům byl přidán suffix “_hp” a LP modelům suffix “_lp”. Je důležité, aby měly všechny části modelů stejné pojmenování. Jako příklad lze uvést dvojici modelů “poklop_lp” a “poklop_hp”.

LP modely byly označeny a exportovány stejným způsobem jako tomu bylo u exportování do Rizom UV. Stejný postup byl zvláště proveden u HP modelů.

LP model ve formátu FBX byl importován do programu Substance Painter. První činností, kterou se autor po importování zabýval, byl Texture Baking. Při nastavení texture bakingu autor zvolil generování všech textur, které Substance Painter nabízí. Dále autor zvolil možnosti *Match: "Always"*, *Antialiasing: "4x"*, *High poly mesh suffix: "_hp"* a *Low poly mesh suffix: "_lp"*.

Po zapečení detailů na LP model, byly na lokomotivu aplikovány různé kombinace Smart Materiálů, Materiálů a pomocí Height parametru byly vytvořeny detaily jako rozdělení plechů nebo nýty.

Obrázek 14: Substance Painter – textury lokomotivy



Zdroj: Autor

4.5 Prototyp

Pro vývoj prototypu hry byla zvolena verze enginu Unreal Engine 5, konkrétně verze 5.1.1. Prototyp prezentuje především funkční pohyb vlaků a s tím spojené nalezení cesty po kolejích.

4.5.1 Vytvoření projektu

Projekt prototypu hry byl vytvořen pomocí *Unreal Project Browser*. V okně aplikace byla zvolena možnost *Games* a následně typ projektu *Blank*. V *Project Defaults* bylo zvoleno

C++. Prázdný projekt autor zvolil proto, protože nebylo třeba základních assetů a logiky hry, které ostatní projekty nabízí, a tudíž to bylo nadbytečné. *Starter Content* byl ponechán, aby byly k dispozici základní materiály.

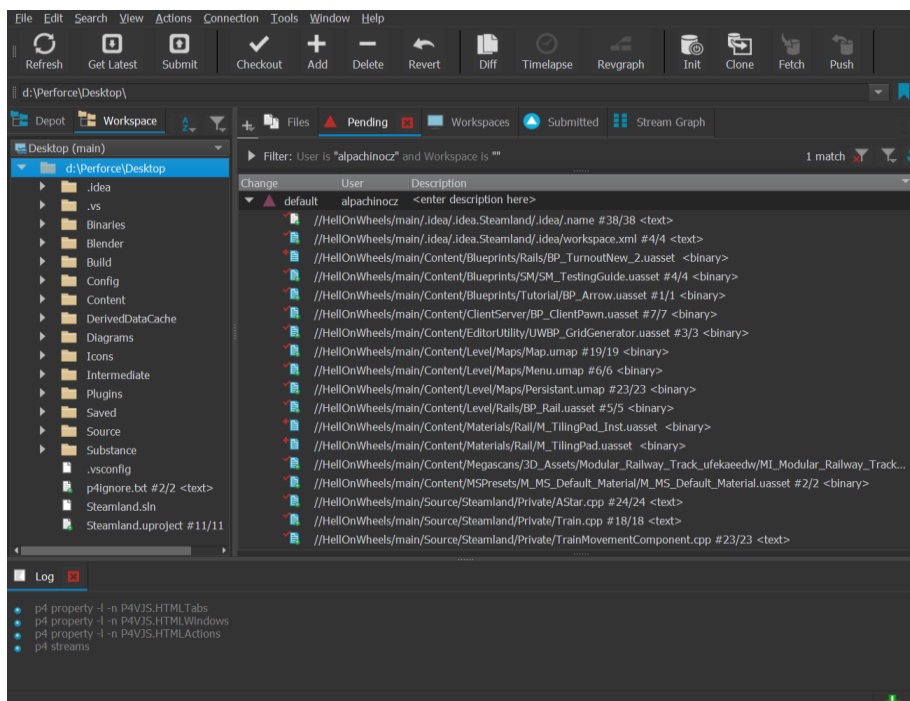
4.5.2 Verzování

Zvoleným verzovacím systémem je Perforce. Jelikož chtěl mít autor během vývoje co nejmenší náklady, použil domácí server NAS jakožto Perforce server. Perforce je na NAS serveru spuštěn jako *Image* v *Dockeru*. Použitým *Image* je *froyok-perforce*.

Před prvním spuštěním Docker *Image*, autor nastavil všechny potřebné parametry, jako jsou jméno *admina*, heslo *admina*, IP adresu, port a umístění *Volume*. Po spuštění Perforce serveru (*Image*) se k němu autor připojil přes aplikaci *P4Admin*, ve které vytvořil nového uživatele s novými právy. Dále byl v aplikaci vytvořen nový *Depot*, sloužící jakožto úložiště pro verzování prototypu hry.

Pod nově vytvořeným účtem se autor skrze aplikaci *P4V* připojil k serveru. Byl vytvořen nový *Workspace* odkazující na složku s prototypem hry. Do složky prototypu byl přidán textový soubor *p4ignore* sloužící pro filtraci souborů a složek, které se na server nemají odesílat, a na které se během verzování nebere ohled. Autor takto vyfiltroval složky *Binaries*, *DerivedDataCash*, *Intermediate*, *Saved* a soubory s koncovkou *.pdb*, *.fbx*, *.psd*, *.tga* a *.sln*. Tyto soubory a složky nejsou potřebné ke spuštění Unreal Engine projektu.

Obrázek 15: P4V



Zdroj: Autor

4.5.3 Pathfinding

Prvním rozsáhlým a základním úkolem, který autor při vývoji prototypu řešil, bylo nalezení cesty (pathfinding) vlaku na kolejích z bodu A do bodu B. Klasický pathfinding, který Unreal Engine nabízí, tedy pathfinding fungující přes *Navigation Mesh*, je v případě hry s vlaky nedostačující a je vytvořen primárně pro navigaci postav, zvířat a dalších živých bytostí. Z tohoto důvodu bylo nutné vytvořit zcela nový systém pro hledání cesty. Autor se rozhodoval mezi algoritmy Dijkstra a AStar (A*) sloužící pro nalezení nejkratší cesty v grafu vycházející z Teorie Grafů. Následně zvolil algoritmus A*, který je využíván ve vývoji her nejčastěji a tak k tomuto algoritmu našel nejvíce podkladů.

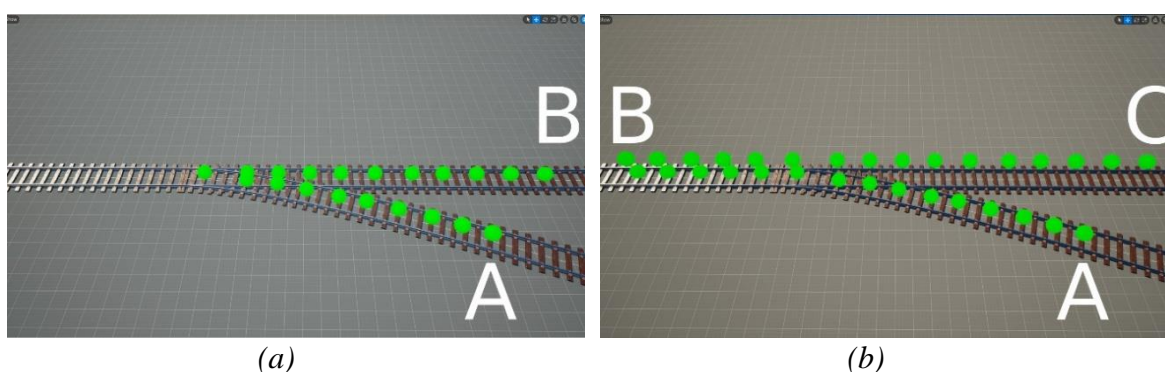
Nejprve bylo nutné vytvořit graf, na který by bylo možné A* algoritmus aplikovat. Prvním krokem pro jeho vygenerování bylo umístit objekty děděné z třídy *Actor*, představující jednotlivé vrcholy grafu (stále se nejedná o výsledný graf), podél kolejí. Tyto Actory, slouží nejen pro generování grafu, ale také pro jednotlivé vlaky, které pomocí *Line Trace* a kolizí s těmito Actory ví, kde se nachází. Autor vytvořil vlastní generátor grafu přes *Editor Utility Widget*, obsahující tlačítka *Initialize Neighbours* a *Generate Graph*. Po kliknutí na *Initialize Neighbours* se všem Actorům podél kolejí nastaví jejich sousedé. Výhybkám autor nastavoval sousedy ručně. Kliknutím na *Generate Graph* jsou

vygenerovány dvě pole (TArray). První obsahuje zmiňované Actory podél kolejí. Druhé pole obsahuje kopie těchto Actorů, ovšem ne ve formě ukazatele na objekt v levelu, ale ve formě struktury (Struct), do které jsou vloženy pouze důležité informace (jako sousedé apod.). Druhé pole struktur již představuje výsledný graf.

Na vygenerovaný graf byl aplikován A* algoritmus, který byl následně rozšířen o rozsáhlou část, která pokrývá korekci počátku nalezené cesty (rozšíření cesty na základě umístění vlaku) a hledání cest z výhybek. Dodatečné hledání cest z výhybek je nutné proto, že se vlak nepohybuje stejně jako běžná herní postava a nemůže tak na výhybce zabočit rovnou na jinou kolej.

Obrázek 16 ukazuje rozdíl v nalezené cestě. Část (a) představuje nalezenou cestu základním A* algoritmem. Vlak by vyjel v této situaci z bodu A, a jakmile by to bylo možné, na rozcestí by nepřírozně zastavil a vyjel směrem k bodu B, na jehož konec by dojel. Část (b) ukazuje nalezenou cestu pomocí rozšířené verze A* algoritmu. Vlak jede z bodu A do bodu B a následně z bodu B do bodu C. Nedojde tak k nepřírozné situaci, kdy by vlak ve skutečnosti nezamířil přímo za výhybkou k bodu C, ale nechá si prostor. Třída obsahující implementaci algoritmu nedědí z žádné třídy a všechny funkce jsou statické.

Obrázek 16: Původní nalezená cesta vs. cesta nalezená rozšířeným algoritmem



Zdroj: Autor

Obrázek 17: Implementace A*

```
PathFinding:MinHeap_openSet;
TArray<FGraphVertex>_closedSet;

openSet.Add(↪ startVertex);

while (openSet.Num() > 0)
{
    FGraphVertex_currentVertex = railGraph[openSet.Remove_first()];
    bool_junctionSetFinished = false;

    closedSet.Add(currentVertex);

    if (currentVertex == endVertex)
    {
        TArray<FGraphVertex>_debug;
        TArray<TArray<FGraphVertex>>_retraced = retrace(graph, ↪ railGraph, startVertex: ↪ railGraph[startVertex.PositionInArray], endVertex: ↪ railGraph[endVertex.PositionInArray]);
        return retraced;
    }

    for (int32_neighbour : currentVertex.Neighbours)
    {
        if (!railGraph[neighbour].IsWalkable || closedSet.Contains(railGraph[neighbour]))
        {
            continue;
        }

        if (currentVertex.IsTurnout && !junctionSetFinished)
        {
            if (closedSet.Num() == 1){...}
            else{...}
            junctionSetFinished = true;
        }

        const double_newCostToNeighbour = currentVertex.gCost + GetDistance(graph, currentVertex:A: currentVertex, currentVertex:B: railGraph[neighbour]);
        if (newCostToNeighbour < railGraph[neighbour].gCost || !openSet.Contains(railGraph[neighbour]))
        {
            railGraph[neighbour].gCost = newCostToNeighbour;
            railGraph[neighbour].hCost = GetDistance(graph, currentVertex:A: railGraph[neighbour], currentVertex:B: endVertex);
            railGraph[neighbour].ParentVertex = currentVertex.PositionInArray;

            if (!openSet.Contains(railGraph[neighbour]))
                openSet.Add(↪ railGraph[neighbour]);
        }
    }
}

return TArray<TArray<FGraphVertex>>();
```

Zdroj: Autor

Samotný pohyb vlaků podél koleje funguje na základě sledování křivky tvořené přes *Spline Component*. Vlak má stanovenou *maximální rychlost* a *aktuální rychlost*. *Aktuální rychlost* je zjištěna jako násobek *maximální rychlosti* a čísla v rozmezí 0 až 1. Toto číslo (0 až 1) udává procentuální rychlost vlaku. Díky tomu je možné, aby se vlak rozjížděl a zastavoval (pomocí *Timeline Component*). Vlak každý *Tick* (frame – snímek) aktualizuje svou pozici podél křivky a to takovou rychlostí, kterou mu udává *aktuální rychlost*. Nová pozice vlaku musí být násobena *Delta Time*, aby nedošlo k nekonzistenci rychlosti v závislosti na FPS hry.

4.5.4 User Interface

UI lze v Unreal Engineu nejjednodušeji vytvářet přes třídu *User Widget*. Další složitější možností je využití *Slate*, což je UI programovací framework. Autor se rozhodl využít *User Widget*, jelikož pro potřeby prototypu není nutné tvořit komplexní UI.

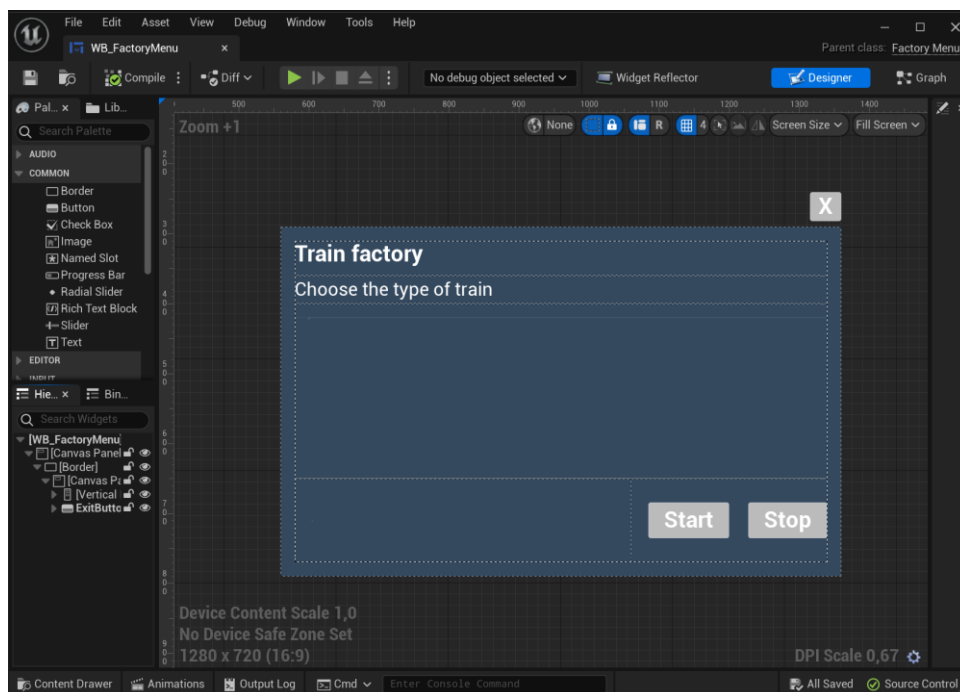
Po vytvoření dědičné třídy z *User Widget* se každý nový widget v Unreal Engineu zobrazí zavoláním *Create Widget* a následným *Add to Viewport*. Pro odstranění widgetu ze zobrazení je volána funkce *Remove from Parent*. Tento způsob skrytí widgetu není ovšem příliš výkonný a autor shledal lepší variantou (především s výhledem do budoucna) widget skrýt pomocí funkce *Set Visibility* s parametrem *Collapsed*. Widget autor stejným způsobem zobrazuje pomocí parametru *Visible*. Jedinou možnou výhodou použití funkce *Remove from parent* je situace, kde je nutné widget dostat do popředí. Widgety jsou totiž hierarchicky poskládané (například jako listy papíru) a to podle pořadí volání *Add to Viewport*.

Pro jednodušší zobrazování UI, autor vytvořil speciální třídu děděnou z *Actor* pojmenovanou *Widget Manager*. *Widget Manager* má za úkol zachytit různé situace, jako zobrazování různých *Widgetů* přes sebe nebo stanovení *ZOrder*, a tím zautomatizovat a urychlit tvoření skriptů.

Pro každý widget ve *Widget Manageru* byly vytvořeny funkce *Hide* a *Unhide*, které v sobě skrývají, krom zobrazení nebo skrytí widgetu, i další logiku jako například, jaký input je od uživatele povolen (*UI Only*, *Game Only*, *UI and Game*), nebo jaké další widgety se mají skrýt/odkrýt. Widgety jsou vytvořeny v *Begin Play* a reference na ně zůstává po celou dobu spuštěné hry.

Všechny widgety podle potřeby používají sloty jako *Canvas Panel*, *Button*, *Text*, *Size Box*, *Scale Box*, *Image* nebo autorem vytvořený *User Widget*. Loading screen widget, který se zobrazuje během načítání levelu, obsahuje animaci načítacího kolečka a animaci tří teček za textem "Loading". Animace UI byly velmi jednoduše vytvořeny přes *keyframe editor* nacházejícím se ve *Widget Blueprintu*, konkrétně v oknech *Animations* a *Timeline*. U načítacího kolečka se animovala rotace obrázku a u teček byla animovaná viditelnost textu. Stejným způsobem byla animována pozice widgetu obsahujícího popis aktuálního úkolu.

Obrázek 18: User Widget – menu továrny



Zdroj: Autor

4.5.5 Levely

V prototypu byl použit level streaming, který umožňuje načtení i smazání levelu z paměti počítače během hraní. Autor zvolil level streaming namísto normálního načítání levelu (BP node *Open Level*), aby bylo možné jednoduše zachytit, kdy je level načten/smazán, což umožnilo vytvořit základní Loading Screen sloužící k zakrytí nenačteného pozadí. Unreal Engine pro level streaming nabízí Blueprint funkce *Load Stream Level* a *Unload Stream Level*. Výstupem těchto funkcí je “*Success*”, který je zavolán po načtení celého levelu – jedná se o latentní funkci. Loading screen v prototypu funguje tak, že je před zavoláním *Load/Unload Stream Level* zobrazen Loading Screen, a po zavolání *Success* je Loading screen odstraněn a následně je podle aktuálního levelu zobrazeno nové UI a další logika.

Autor vytvořil první level, který slouží jako *Persistent level* a je tak vždy načten (Always Loaded). V tomto levelu se nachází pouze autorem vytvořený Widget Manager (kapitola 4.5.4). Dalším levelem bylo *Main Menu*. Tento level je načítán vždy při spuštění aplikace po inicializaci všech UI a proměnných nebo po navrácení do menu (například z jiného levelu). Třetím levelem je samotná mapa prototypu, ve které jsou dále testovány mechaniky hry. Tato mapa je načtena po kliknutí na tlačítko *Play* v *Main Menu* a z paměti se odstraňuje po stisknutí tlačítka *Exit* v *Pause Menu*.

Obrázek 19: Level načtený v hlavním menu (main menu)



Zdroj: Autor

4.5.6 Gameplay

Zachytávání vstupů od uživatele, a s tím spojená logika hry, byla vložena do *Player Controller*. Unreal Engine nabízí dva systémy pro zachytávání vstupů od uživatele, tzv. *Input System* a *Enhanced Input System*. *Input system* je starší verze. Novější varianta *Enhanced Input System* umožňuje jednodušší práci s komplexními vstupy nebo např. přemapování ovládacích prvků během hry. Pro prototyp nebylo potřeba komplexních vstupů, proto Autor použil pro zachytávání vstupů od hráče klasický *Input System*.

Mapování kláves bylo vytvořeno pro Esc (zastavení hry), W-A-S-D (pohyb kamery), F (změna polohy kamery k vybranému vlaku), T (ukázka mapování kláves), 0-9 (výběr lokomotivy v daném slotu), levé kliknutí myši (výběr prvku ve hře), pravé kliknutí myši (pohyb vlaku).

5 Výsledky a diskuse

5.1 Testování

Po dokončení první základní verze prototypu hry bylo provedeno testování použitelnosti a zjišťovaly se celkové dojmy účastníků testování na prototyp hry. Testování prototypu se zúčastnilo 10 respondentů.

Jelikož verze testovaného prototypu obsahuje pouze zcela základní funkce, jako jsou pohyb vlaku, základní UI a vytváření lokomotiv, byla hra jednotlivým respondentům představena před samotným testováním, aby si o hře udělali představu a nehodnotili ji jako hotový produkt.

Testování prováděl autor práce a probíhalo formou moderovaného online rozhovoru. V první části testování dostal jednotlivý respondent k dispozici scénář, podle kterého následně postupoval. Dále měl respondent za úkol myslet nahlas, aby autor dostal bezprostřední zpětnou vazbu o jeho pocitech z hraní (*Cognitive think-aloud protocol*, kapitola 3.7.5). Při postupu scénářem respondent sdílel svoji obrazovku, kterou autor spolu se zvukem zaznamenával.

Scénář

1. Spustit aplikaci prototypu hry
2. Změnit grafické nastavení hry
3. Spustit hru
4. Seznámit se s ovládáním hry
5. Vybrat nebo označit v levelu poškozenou lokomotivu
6. Poškozenou lokomotivu přemístit na místo zelené šipky
7. Opravit poškozenou lokomotivu v „Repair Depot“
8. Vytvořit novou lokomotivu ve „Factory“
9. Nově vytvořenou lokomotivu přemístit na místo zelené šipky
10. Zastavit hru a odejít zpět do hlavního menu
11. Vypnout hru

V druhé části testování, po úspěšném dokončení scénáře, každý respondent dostal k vyplnění dva dotazníky vytvořené autorem v Google Forms. Prvním byl standardizovaný dotazník SUS (kapitola 3.7.4), který měří míru vnímané snadnosti používání prototypu. Cílem autora bylo získat v tomto dotazníku celkové skóre SUS vyšší jak 68. Takové skóre je v porovnání s ostatními testovanými aplikacemi nadprůměrné (*Tabulka 8*). Druhý

dotazník obsahuje autorem položené otázky, které se zaměřují na pocity respondentů ze hraní a z hry jako takové, na to co se respondentům líbilo, nelíbilo nebo jim naopak ve hře chybělo.

Průměrný čas testování, zahrnující jak průchod scénářem, tak vyplnění dotazníků, byl cca 15 minut.

5.1.1 SUS dotazník

Vyplněním dotazníku byla získána data sloužící pro měření míry použitelnosti. Respondenti odpovídali na 10 standardizovaných otázek. Výsledkem testování SUS je skóre pohybující se v rozmezí 0-100. Každá otázka nabízí odpověď (kapitola 3.7.4) na stupnici 1 až 5 (1-Rozhodně nesouhlasím, 2-Nesouhlasím, 3-Nevím, 4-Spíše souhlasím, 5-Rozhodně souhlasím).

Otázky obsažené v dotazníku SUS jsou tyto:

1. Myslím si, že bych rád hrál tuto hru opakovaně.
2. Myslím si, že je prototyp hry zbytečně složitý.
3. Myslím si, že se prototyp hry snadno používá.
4. Myslím si, že bych potřeboval/a pomoc člověka z technické podpory, abych mohl/a prototyp hry používat (hrát).
5. Myslím si, že jsou funkce do prototypu hry dobře začleněny.
6. Myslím si, že je prototyp hry nekonzistentní.
7. Myslím si, že se s prototypem hry většina lidí naučí rychle pracovat.
8. Myslím si, že se prototyp hry těžko ovládá (je těžkopádný).
9. Při hraní prototypu se cítím jistě.
10. Musím se naučit hodně věcí, abych mohl/a začít prototyp hrát.

Dosažením hodnot z odpovědí dotazníku do vzorce bylo zjištěno skóre jednotlivých respondentů. Všech deset respondentů dosáhlo vysokého skóre. Tento výsledek napovídá, že použitelnost prototypu hry je na dobré úrovni.

Graf 1: Finální skóre jednotlivých respondentů



Zdroj: Autor

5.1.2 Vlastní dotazník

Výsledné skóre dotazníku SUS říká, jak je aplikace použitelná, ale nezachycuje konkrétní informace a zpětnou vazbu respondentů na hru. Proto autor vytvořil svůj vlastní dotazník, ve kterém tyto dodatečné informace od respondentů zachycuje.

Otázky vlastního dotazníku jsou:

1. Hry hraji: (Často, Příležitostně, Vůbec)
2. RTS (real-time strategy) hry: (Vyhledávám, Hraji zřídka, Nehraji, Jiné)
3. Prototyp hry mě svým tématem: (stupnice Vůbec nezaujal – Velmi zaujal)
4. Ve hře potenciál: (stupnice Rozhodně nevidím – Rozhodně vidím)
5. Měli jste při hraní prototypu pocit, že Vám chybí nějaké funkce v ovládaní?
6. Měli jste při hraní prototypu dostatek vizuální zpětné vazby?
7. Poskytoval Vám prototyp hry dostatek informací?
8. Je něco, co byste ve hře rádi viděli?
9. Je něco, co byste ze hry odstranili?

5.1.3 Vyhodnocení

Prototyp hry se zaměřuje především na funkční navigaci vlaků po kolejích. Z tohoto důvodu neobsahuje příliš funkcí, které může respondent testovat. Prototyp je jednoduchý, a proto autor předpokládal, že hra bude mít v testu použitelnosti kladné výsledky.

Tento předpoklad se naplnil. Všichni respondenti získali vyšší skóre nežli je průměr (68) a celkové skóre SUS činilo 88. Autor se při hodnocení výsledků řídil touto tabulkou:

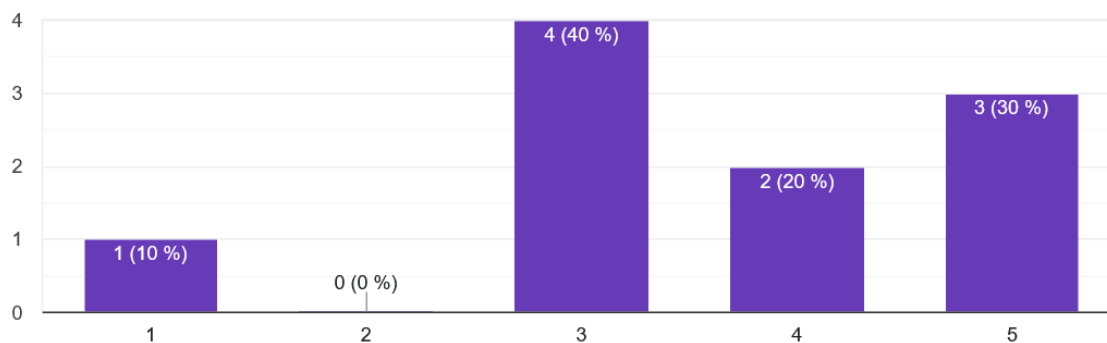
Tabulka 8: Hodnocení skóre SUS

SUS skóre	Hodnocení
> 80.3	Výborně
68 – 80.3	Dobře
68	Průměrné hodnocení SUS
51 – 68	Špatné
< 51	Nedostačující

Zdroj: Will T (2023)

Celkové skóre spadá do hodnocení “Výborně”. Toto hodnocení říká, že se hra vydává správným směrem a použitelnost aplikace je excelentní. Ovšem i přes výborný výsledek použitelnosti lze najít nedostatky. Například odpovědi na první otázku “Myslím si, že bych rád hrál tuto hru opakovaně.” naznačují (Graf 2), že téměř polovina respondentů nemá dostatečnou představu o tom, jak hra bude vypadat a jestli by ji v budoucnosti hráli.

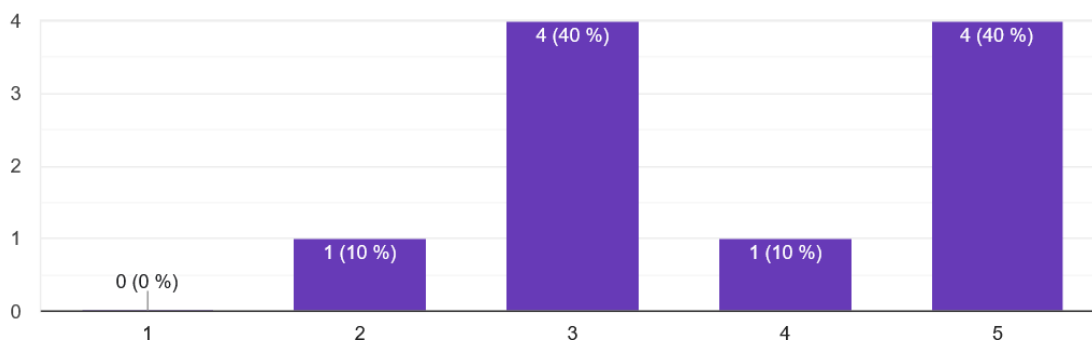
Graf 2: SUS – Otázka č.1



Zdroj: Autor

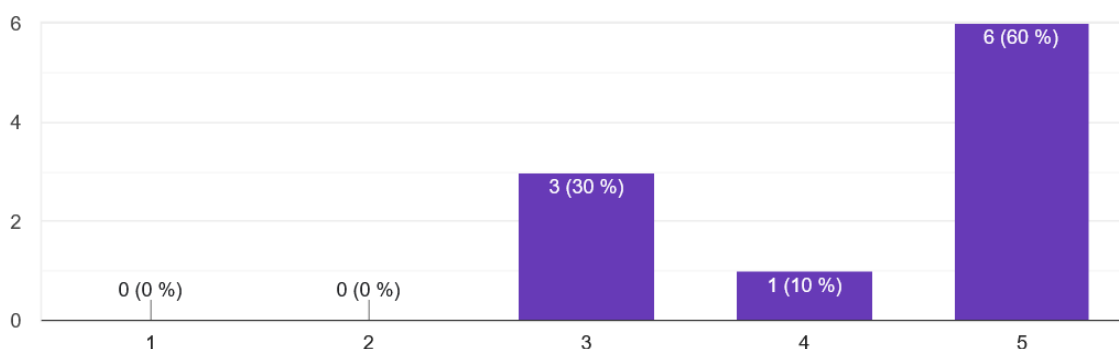
Toto tvrzení potvrzuje i autorův vlastní dotazník, ve kterém na třetí otázku odpovídá 40% respondentů “nevím” (*Graf 3*). Ovšem potenciál ve hře vidí 70% respondentů a koncept prototypu hry tak není špatně navržen (*Graf 4*).

Graf 3: Vlastní dotazník – Otázka č.3



Zdroj: Autor

Graf 4: Vlastní dotazník – Otázka č.4



Zdroj: Autor

Na základě odpovědí respondentů z vlastního dotazníku a rozhovoru s respondenty během hraní bylo stanoveno, čím se bude vývoj hry v budoucnu zabývat. Vývoj hry bude soustředěn především na implementaci a přidávání nových mechanik a funkcí, aby byl lépe vidět celý koncept hry a bylo možné ve hře dělat více aktivit.

Hlavní funkce, které by respondenti ocenili, jsou:

- více druhů vlaků
- vagóny
- spojování vlaků
- možnost z vlaků střílet
- větší level
- kolize vlaků
- rozmanité prostředí
- více úkolů.

Vývoj se bude dále zabývat funkcemi, které respondentům chyběly. Jsou jimi:

- zlepšení ovládání kamery
- přidání oddálení a přiblížení kamery
- přidání funkce, která umožní pohyb kamery po najetí myši na okraj obrazovky

Budoucí vývoj hry bude implementovat více informací a vizuálních prvků, které hráče usměrní v tom, co má dělat. Dle odpovědí z dotazníku se bude jednat o:

- lepší viditelnost, kde se v levelu nachází jednotlivé úkoly (např. šipka, která ukazuje na okraji obrazovky směr, kterým se má hráč pohybovat)
- přehlednější ovládání (lepší vizualizace toho, co se stane po najetí myši na kolej)
- celkové zvýšení přehlednosti (výraznější rozlišení budov, jasnější zobrazení stavu vlaků, více kontrastní tlačítka v UI)
- více nápověd během hraní (jak spustit tutorial, jaké klávesy v určitý moment použít)
- zpětnou možnost zobrazení úkolů.

Hlavní funkci hledání cesty a pohyb vlaků, na který byl prototyp soustředěn, žádný z respondentů nekritizoval. Systém navigace vlaků byl, krom vizuálních nedostatků, hodnocen kladně. Výsledné skóre SUS bylo excelentní. Na základě těchto výsledků lze konstatovat, že prototyp hry v testování uspěl a autor se může plně soustředit na další vývoj.

6 Závěr

Tato bakalářská práce se zabývá popisem vývoje herní aplikace ve spojitosti s herním enginem Unreal Engine. Cílem práce bylo charakterizovat proces vývoje 3D hry, jeho důležité součásti, shrnout důležité pojmy a popsat publikaci hry na platformu Steam. Tento cíl je splněn teoretickou částí práce.

První část teoretické práce představuje různé pojetí project managementu, který je ve vývoji herní aplikace, vzhledem k velikosti projektu, téměř nutností. Pokud chce indie vývojář pracovat efektivně, měl by využít agilního přístupu k project managementu. Kapitola Pre-Production čtenářům popisuje, na jaké otázky je nutné před samotným vývojem najít odpovědi a jakým způsobem tyto informace shrnout. Nejvíce rozsáhlou částí je Production fáze. Ta zahrnuje největší část procesu vývoje. Kapitola rozepisuje důležité disciplíny, se kterými se indie vývojář během vývoje 3D hry setkává. Bylo popsáno workflow tvorby 3D assetů, které patří k důležitým dovednostem indie vývojáře, především pokud se chce vyhnout nákladům spojených s jejich pořizováním. V současné době je k dispozici Blender, který je zcela zdarma a každý má tak možnost vytvořit assety dle svých představ. Práce se dále zabývala tématem UX a testováním použitelnosti, u něhož byly představeny dvě metody System Usability Scale (SUS) a Cognitive think-aloud protocol. Závěr teoretické části práce je věnován publikaci hry na platformu Steam. O tomto tématu není příliš informací ani návodů. Informace o publikaci jsou navíc k nalezení pouze v anglickém jazyce. Bakalářská práce shrnuje jednotlivé kroky, které je nutné pro publikaci hry udělat, od vytvoření vývojářského účtu až po vytvoření stránky obchodu (Store Page).

Praktická část práce se zabývá plněním dílčího cíle bakalářské práce, tedy vytvořením prototypu hry a jeho následným testováním. Byly využity poznatky z části teoretické, především z kapitol project management, pre-production fáze, production fáze a testování použitelnosti. Po dokončení návrhu hry byl vytvořen Game Design Document v aplikaci Trello a project management v aplikaci GitLab. Project management držel celý projekt prototypu hry ucelený a spolu s Game Design Documentem zajistil, že nic nebylo během vývoje opomenuto. Autor do prototypu hry vymodeloval a otexturoval vlastní assety, jako například model vlaku nebo kolejí. U tvorby assetů bylo postupováno tak, jak je popsáno v teoretické části. Samotný prototyp hry byl vyvíjen v Unreal Engine 5.1.1. Největším problémem, který autor řešil a pokryl většinu času vývoje, bylo vytvoření funkčního hledání nejkratší cesty vlaků po kolejích a jejich následný pohyb. Tento úkol byl vyřešen

implementací AStar algoritmu s mnoha úpravami, aby byl přizpůsoben pro navigaci vlaků. Celý vývoj v Unreal Engine usnadňovalo verzování, které zajišťoval Perforce server.

Závěr praktické části práce se věnuje testování prototypu hry. Testování použitelnosti (SUS) dalo autorovi informaci o tom, že je prototyp na excelentní úrovni a vyvíjí se správným směrem. Na základě výsledků testování (informací získaných v průběhu toho, kdy respondenti přemýšleli nahlas, a informací získaných z vlastního dotazníku) bylo určeno, čím se bude vývoj nadále zabývat a na jaké podněty bude reagovat.

Mezi důležité body budoucího vývoje hry, které byly zjištěny testováním, patří oprava stávajících chyb (především u hledání cesty, kdy se vlak rozjede špatným směrem a hra spadne), přidání funkce spojování vlaků, kolize vlaků a přidání možnosti střelby z vlaků. Dále do plánu vývoje hry bude zařazeno zpřehlednění ovládání, zlepšení herního UI a přidání obsahu, který nabídne větší počet druhů vlaků nebo rozmanitou nabídku zajímavých úkolů.

Tato bakalářská práce může pomoci začínajícím vývojářům se zorientovat v tématu vývoje 3D herní aplikace a ulehčit jim tak začátky tvorby na jejich vlastním projektu. Další velkou pomocí začínajícím vývojářům jsou také důležité pojmy, které mohou použít jako záchytné body. Zpětná vazba respondentů testování, a s tím spojené výsledky dotazníků, byly velmi důležité pro stanovení budoucího vývoje hry a poskytla informaci o tom, že hra je vyvíjena správným směrem a má smysl s vývojem i nadále pokračovat.

7 Seznam použitých zdrojů

365TIPU, 2017. TIP#757: Co je to verzování a jak vám v tom pomohou cloudová úložiště?. In: @365tipu [online]. [cit. 2023-03-08]. Dostupné z: <https://365tipu.cz/2017/04/04/tip757-co-je-to-verzovani-a-jak-vam-v-tom-pomohou-cloudova-uloziste/>

3D-ACE, 2021. 3D MODELING FOR VIDEO GAMES: HOW TO CREATE BEAUTIFUL ASSETS. In: 3D-Ace [online]. [cit. 2023-03-05]. Dostupné z: <https://3d-ace.com/blog/3d-modeling-for-video-games-how-to-create-beautiful-assets/>

8BIT, 2023. What the heck is Level Design? And what does a level designer do?. In: 8Bit Recruitment Agency [online]. [cit. 2023-03-05]. Dostupné z: <https://8bitplay.com/blog/what-the-heck-is-level-design-and-what-level-designer-does>

AGILE ALLIANCE, 2023. Scrum. In: Agile Alliance [online]. [cit. 2023-03-04]. Dostupné z: <https://www.agilealliance.org/glossary/scrum/>

ALGADRIE, SYED HASSAN, 2021. 8 PLATFORMS FOR PUBLISHING INDIE GAMES. In: Pokde.Net [online]. [cit. 2023-03-05]. Dostupné z: <https://pokde.net/gaming/game-industry/8-platforms-for-publishing-indie-games>

ALTEXSOFT, 2023. Agile Project Management: Best Practices and Methodologies. In: AltexSoft [online]. [cit. 2023-03-04]. Dostupné z: <https://www.altexsoft.com/whitepapers/agile-project-management-best-practices-and-methodologies/>

ATLASSIAN, 2023. What is version control?. In: Atlassian [online]. [cit. 2023-03-08]. Dostupné z: <https://www.atlassian.com/git/tutorials/what-is-version-control>

AUTODESK, 2023. Display UV distortions. In: Maya Help [online]. [cit. 2023-03-12]. Dostupné z: <https://help.autodesk.com/view/MAYAUL/2022/ENU/?guid=GUID-637CCCB0-3513-4DFD-A8D5-EB1AB5CCEB9D>

AYMAN, Zaref, 2022. Unreal Engine 5 Will Breathe New Life Into Virtual Productions In Southeast Asia. In: IGN Southeast Asia [online]. [cit. 2023-03-05]. Dostupné z: <https://sea.ign.com/epic-games/185672/news/unreal-engine-5-will-breathe-new-life-into-virtual-productions-in-southeast-asia>

BANGOR, Aaron, Philip T. KORTUM a James T. MILLER, 2008. An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction* [online]. 24(6), 574-594 [cit. 2023-03-06]. ISSN 1044-7318. Dostupné z: doi:10.1080/10447310802205776

BOYLLS, Travis, 2022. A Detailed Guide to Getting Your Game on Steam. In: WikiHow [online]. [cit. 2023-03-05]. Dostupné z: <https://www.wikihow.com/Get-Your-Game-on-Steam>

BRIDGETT, Rob, 2017. Getting Game Audio Right: The Big Picture. *Game Developer* [online]. [cit. 2023-03-10]. Dostupné z: <https://www.gamedeveloper.com/audio/bootstrapping-sound-for-your-indie-game>

BUDIU, Raluca, 2017. Quantitative vs. Qualitative Usability Testing. *Nielsen Norman Group* [online]. [cit. 2023-03-06]. Dostupné z: <https://www.nngroup.com/articles/quant-vs-qual/>

CORRALES, Enrique, 2021. Best Project Management Software For Game Developers. In: *Developer.com* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.developer.com/project-management/project-management-game-developers/>

CREATIVE COMMONS ČESKÁ REPUBLIKA, 2023a. Licenční prvky. In: *Creative Commons Česká republika* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.creativecommons.cz/licence-cc/licencni-prvky/>

CREATIVE COMMONS ČESKÁ REPUBLIKA, 2023b. Úvod. In: *Creative Commons Česká republika* [online]. [cit. 2023-03-05]. Dostupné z: <http://www.creativecommons.cz/uvod/>

CRIDER, Michael, 2017. What Is Steam Direct, and How Is It Different from Greenlight?. In: *How-To Geek* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.howtogeek.com/311311/what-is-steam-direct-and-how-is-it-different-from-greenlight/>

DARVIDEO, 2021. VIDEO GAME ANIMATION. In: *Darvideo* [online]. [cit. 2023-03-05]. Dostupné z: <https://darvideo.tv/dictionary/video-game-animation/>

DEMIRÇIN, Yiğithan, 2022. What is level design?. In: *Mobidictum* [online]. [cit. 2023-03-05]. Dostupné z: <https://mobidictum.com/game-industry/level-design/>

DENHAM, Thomas, 2023a. What is Substance Painter?. In: *Concept Art Empire* [online]. [cit. 2023-03-05]. Dostupné z: <https://conceptartempire.com/what-is-substance-painter/>

DENHAM, Thomas, 2023b. What is UV Mapping & Unwrapping?. In: *Concept Art Empire* [online]. [cit. 2023-03-05]. Dostupné z: <https://conceptartempire.com/uv-mapping-unwrapping/>

DIGITALRUNE, 2016. Morph Targets. In: *DigitalRune* [online]. [cit. 2023-03-05]. Dostupné z: <https://digitalrune.github.io/DigitalRune-Documentation/html/c961412f-43c7-4b82-a68d-b20dd2d5131f.htm>

DIGITÉ, 2023. What Is Scrum Methodology? & Scrum Project Management. In: *Digité* [online]. [cit. 2023-03-04]. Dostupné z: <https://www.digite.com/agile/scrum-methodology/>

DRIES, Timothy, 2022. TEXEL DENSITY. In: *Beyond Extent* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.beyondextent.com/deep-dives/deepdive-texeldensity>

EPIC GAMES, 2023. Unreal Engine 5.1 Documentation. Unreal Engine 5.1 Documentation [online]. [cit. 2023-03-13]. Dostupné z: <https://docs.unrealengine.com/5.1/en-US/>

- EVAN, 2016. This is Unreal Engine. In: *Hot Gates* [online]. [cit. 2023-03-05]. Dostupné z: <https://hotgates.eu/this-is-unreal-engine/>
- FAN, Mingming, Serina SHI a N. TRUONG, 2020. Practices and Challenges of Using Think-Aloud Protocols in Industry: An International Survey. *Journal of Usability Studies* [online]. **15**(2), 85-102 [cit. 2023-03-05]. Dostupné z: <https://uxpajournal.org/practices-challenges-think-aloud-protocols-survey/>
- FRA3POINT, 2022. Total Baker - Texture Baking System. In: Unity Forum [online]. [cit. 2023-03-12]. Dostupné z: <https://forum.unity.com/threads/total-baker-texture-baking-system.546341/>
- FRANCZAK, Michal, 2019. Texturing Brooklyn Building - Tip of the Week. In: ArtStation [online]. [cit. 2023-03-12]. Dostupné z: <https://evermotion.org/tutorials/show/11536/texturing-brooklyn-building-tip-of-the-week>
- FRENCH, John, 2022. How to write a game design document (with examples). In: *Game Dev Beginner* [online]. [cit. 2023-03-05]. Dostupné z: <https://gamedevbeginner.com/how-to-write-a-game-design-document-with-examples/>
- GAROFALO, Emma, 2022. An Introduction to Hard Surface 3D Modeling. In: *Make Use Of* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.makeuseof.com/what-is-hard-surface-modeling/>
- GSA, 2023. Usability Testing. In: *Usability.gov* [online]. [cit. 2023-03-06]. Dostupné z: <https://www.usability.gov/how-to-and-tools/methods/usability-testing.html>
- ITUTS EDITOR, 2022. Unreal Engine for Beginners | Is It Good? (Full Guide). *InspirationTuts* [online]. [cit. 2023-03-06]. Dostupné z: <https://inspirationtuts.com/unreal-engine-for-beginners/>
- JAIMOVICH, Desirée. The 5 Phases of Project Management. In: *InvGate* [online]. 2022 [cit. 2023-03-04]. Dostupné z: <https://blog.invgate.com/phases-of-project-management>
- JUEGOADMIN, 2021. GET ASSETS FOR YOUR GAME WITH THESE 5 METHODS. In: *Juego Studios* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.juegostudio.com/blog/get-assets-for-your-game-with-these-5-methods>
- KANBAN TOOL, 2023. History of Kanban. In: *Kanban Tool* [online]. [cit. 2023-03-05]. Dostupné z: <https://kanbantool.com/kanban-guide/kanban-history>
- KANBANIZE, 2023a. What Is a Kanban Board and How to Use It? Basics Explained. In: *Kanbanize* [online]. [cit. 2023-03-05]. Dostupné z: <https://kanbanize.com/kanban-resources/getting-started/what-is-kanban-board>
- KANBANIZE, 2023b. What Is a Kanban Card? Details and Example. In: *Kanbanize* [online]. [cit. 2023-03-05]. Dostupné z: <https://kanbanize.com/kanban-resources/getting-started/what-is-kanban-card>

- KANBANIZE, 2023c. What Is a WIP Limit in Kanban, and Why Is It Important?. In: *Kanbanize* [online]. [cit. 2023-03-05]. Dostupné z: <https://kanbanize.com/kanban-resources/getting-started/what-is-wip>
- KANBANIZE, 2023d. What Is Kanban? Explained for Beginners. In: *Kanbanize* [online]. [cit. 2023-03-05]. Dostupné z: <https://kanbanize.com/kanban-resources/getting-started/what-is-kanban>
- KEVURU GAMES, 2021. 6 KEY STAGES OF GAME DEVELOPMENT: FROM CONCEPT TO STANDING OVATION. In: *Kevuru Games* [online]. [cit. 2023-03-05]. Dostupné z: <https://kevurugames.com/blog/6-key-stages-of-game-development-from-concept-to-standing-ovation/>
- KLIUCH, Denys, 2022. Creating A Game Design Document: Key Points To Consider. In: *Whimsy Games*. [online]. [cit. 2023-03-05]. Dostupné z: <https://whimsygames.co/blog/game-design-instructions-examples/>
- LEI, Vincent, 2017. Study of Low Poly/High Poly (WIP). In: *Pinterest* [online]. [cit. 2023-03-15]. Dostupné z: <https://cz.pinterest.com/pin/544443042455444996/>
- MANAGEMENTMANIA. Řízení projektů (Project Management). In: *ManagementMania.com* [online]. 2016 [cit. 2023-03-04]. Dostupné z: <https://managementmania.com/cs/metody-rizeni-projektu>
- MANY WATERS LTD, 2022. GAME DEVELOPMENT STAGES: HOW ARE VIDEO GAMES CREATED?. In: *Walla Walla Studio* [online]. [cit. 2023-03-05]. Dostupné z: <https://wallawallastudio.com/article/game-development-sta>
- MASTERCLASS, 2021. How Video Game Development Works: 8 Key Roles Explained. In: *MasterClass* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.masterclass.com/articles/how-video-game-development-works#8-key-roles-in-video-game-development>
- MESQUITA, Luis, 2021a. Everything About PBR Textures And A Little More - PART 1. In: *ArtStation* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.artstation.com/blogs/luismesquita/PwEm/everything-about-pbr-textures-and-a-little-more-part-1>
- MESQUITA, Luis, 2021b. Everything About PBR Textures And A Little More - PART 2. In: *ArtStation* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.artstation.com/blogs/luismesquita/jGXd/everything-about-pbr-textures-and-a-little-more-part-2>
- MESQUITA, Luis, 2021c. Everything About PBR Textures And A Little More - PART 3. In: *ArtStation* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.artstation.com/blogs/luismesquita/RYV0/everything-about-pbr-textures-and-a-little-more-part-3>
- NIEDOBA, Matthäus, 2023. Git vs Perforce for game development. In: *Anchorpoint* [online]. [cit. 2023-03-08]. Dostupné z: <https://www.anchorpoint.app/blog/git-vs-perforce-for-game-development>

NIELSEN, Jakob, 2012. How Many Test Users in a Usability Study?. *Nielsen Norman Group* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.nngroup.com/articles/how-many-test-users/>

NORMAN, Don a Jakob NIELSEN, 2023. The Definition of User Experience (UX). *Nielsen Norman Group* [online]. [cit. 2023-03-06]. Dostupné z: <https://www.nngroup.com/articles/definition-user-experience/>

NUCLINO, 2023a. Game Design Document Template and Examples. In: *Nuclino* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.nuclino.com/articles/game-design-document-template>

NUCLINO, 2023b. Project Management Software for Game Development. In: *Nuclino* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.nuclino.com/solutions/game-development-project-management>

NVIDIA, 2023. UV Layout Best Practices. In: *NVIDIA Omniverse Documentation* [online]. [cit. 2023-03-05]. Dostupné z: https://docs.omniverse.nvidia.com/prod_simready/prod_simready/simready-asset-creation/uv-best-practices.html

ON MARS 3D, 2021. High to Low Poly Workflow in 3D Modeling. In: *Youtube* [online]. [cit. 2023-03-05]. Dostupné z: https://www.youtube.com/watch?v=YDu9pYMkkSM&t=1051s&ab_channel=OnMars3D

PEREZ, Sheena, 2017. Steam Greenlight vs. Steam Direct: What indies need to know. In: *Game Developer* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.gamedeveloper.com/business/steam-greenlight-vs-steam-direct-what-indies-need-to-know>

PICKELL, Devin, 2019. The 7 Stages of Game Development. In: *G2* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.g2.com/articles/stages-of-game-development>

PILGWAY, 2023. Co Je UV Mapping?. In: *3DCoat* [online]. [cit. 2023-03-05]. Dostupné z: <https://3dcoat.com/cs/articles/article/what-is-uv-mapping/>

PLAYCENT GAMES, 2022. How to Make a Game Design Document (GDD). In: *Playcent Games* [online]. [cit. 2023-03-05]. Dostupné z: <https://playcentgames.com/how-to-make-a-game-design-document-gdd/>

RADIGAN, Dan, 2023a. Agile vs. waterfall project management. In: *Atlassian* [online]. [cit. 2023-03-04]. Dostupné z: <https://www.atlassian.com/agile/project-management/project-management-intro>

RADIGAN, Dan, 2023b. Kanban. In: *Atlassian* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.atlassian.com/agile/kanban>

RIAD, 2022. Writing Modern Game Design Documents (+Examples). In: *Codecks* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.codecks.io/blog/2020/writing-modern-game-design-documents/>

ROOM 8 STUDIO, 2020. STEP-BY-STEP PROCESS OF GAME DEVELOPMENT PRE-PRODUCTION PHASE. In: *Room 8 Studio* [online]. [cit. 2023-03-05]. Dostupné z: <https://room8studio.com/news/game-pre-production-core-steps/>

SENNETT, Phil. Agile vs. traditional project management. In: *University of Rochester* [online]. 2022 [cit. 2023-03-04]. Dostupné z: <https://www.rochester.edu/emerging-leaders/agile-vs-traditional-project-management/>

SEYDAMET, Eldar, 2021. *Create Game Ready Assets in a Real-Time Pipeline Using Unreal Engine* [online]. In: . [cit. 2023-03-05]. Dostupné z: <https://discover.therookies.co/2021/07/21/create-game-ready-assets-using-unreal-engine/>

SCHADE, Amy, 2013. Remote Usability Tests: Moderated and Unmoderated. Nielsen Norman Group [online]. [cit. 2023-03-13]. Dostupné z: <https://www.nngroup.com/articles/remote-usability-tests/>

SLÁDEČEK, David, 2017. Co je to projektový management?. In: *Jobfairs.eu* [online]. [cit. 2023-03-04]. Dostupné z: <https://www.jobfairs.eu/magazin/co-je-to-projektovy-management>

STEFYN, Nadia, 2022. How video games are made: the game development process. In: *CG Spectrum* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.cgspectrum.com/blog/game-development-process>

STRIHOVÁ, Barbora, 2020. Scrum jednoduše. In: *Medium* [online]. [cit. 2023-03-04]. Dostupné z: <https://medium.com/@barborastrihova/scrum-jednodu%C5%A1e-b55ff145bb35>

SUMEET, 2020. Remote vs. In-Person Usability Testing in 2 minutes. In: *UX Planet* [online]. [cit. 2023-03-13]. Dostupné z: <https://uxplanet.org/remote-vs-in-person-usability-testing-in-2-minutes-9b6c13555c3b>

T, Will, 2023. Articles, UX Research Measuring and Interpreting System Usability Scale (SUS). In: *UIUX Trend* [online]. [cit. 2023-03-12]. Dostupné z: <https://uiuxtrend.com/measuring-system-usability-scale-sus/>

TAWFIK, Andrew A., Jessica GATEWOOD, Jaelyn J. GISH-LIEBERMAN a Andrew J. HAMPTON, 2022. Toward a Definition of Learning Experience Design. *Technology, Knowledge and Learning* [online]. **27**(1), 309-334 [cit. 2023-03-06]. ISSN 2211-1662. Dostupné z: doi:10.1007/s10758-020-09482-2

TEAMHOOD, 2022. Kanban Pull System Guide: Definition & Examples. In: *Teamhood* [online]. [cit. 2023-03-05]. Dostupné z: <https://teamhood.com/kanban-resources/kanban-pull-system/>

TECHOPEDIA, 2017. Level Design. In: *Techopedia* [online]. [cit. 2023-03-05]. Dostupné z: <https://www.techopedia.com/definition/88/level-design>

THE LEVEL DESIGN BOOK, 2023. Blockout. In: *The Level Design Book* [online]. [cit. 2023-03-12]. Dostupné z: <https://book.leveldesignbook.com/process/blockout>

- TRAMMELL, Kent, 2023. Texturing Game Assets with Blender. In: *CG Cookie* [online]. [cit. 2023-03-05]. Dostupné z: <https://cgcookie.com/courses/texturing-game-assets-with-blender>
- UNITY TECHNOLOGIES, 2023. GAME DEVELOPMENT TERMS. In: *Unity* [online]. [cit. 2023-03-05]. Dostupné z: <https://unity.com/how-to/beginner/game-development-terms>
- VALVE CORPORATION, 2022a. Skeletal animation. In: *Valve Developer Community* [online]. [cit. 2023-03-05]. Dostupné z: https://developer.valvesoftware.com/wiki/Skeletal_animation
- VALVE CORPORATION, 2023b. Steam Direct. In: *Steamworks* [online]. [cit. 2023-03-05]. Dostupné z: <https://partner.steamgames.com/steamdirect>
- VALVE CORPORATION, 2023c. Steamworks Documentation. In: *Steamworks* [online]. [cit. 2023-03-05]. Dostupné z: <https://partner.steamgames.com/doc/home>
- VIRZI, Robert A., 1992. Refining the Test Phase of Usability Evaluation: How Many Subjects Is Enough?. *Human Factors: The Journal of the Human Factors and Ergonomics Society* [online]. **34**(4), 457-468 [cit. 2023-03-05]. ISSN 0018-7208. Dostupné z: doi:10.1177/001872089203400407
- VNTANA, 2021. What Is “Texture Baking” And Why Is It Important?. In: *AR/VR Journey* [online]. [cit. 2023-03-05]. Dostupné z: <https://arvrjourney.com/what-is-texture-baking-and-why-is-it-important-a9fcadec89f>
- WAQAR, 2018. What is Megascans?. In: *Quixel* [online]. [cit. 2023-03-05]. Dostupné z: <https://help.quixel.com/hc/en-us/articles/115000607525-What-is-Megascans->
- WIKIPEDIA, 2022. UV mapping. In: *Wikipedia* [online]. [cit. 2023-03-12]. Dostupné z: https://en.wikipedia.org/wiki/UV_mapping
- WRIKE, 2023. What Is Scrum in Agile?. In: *Wrike* [online]. [cit. 2023-03-04]. Dostupné z: <https://www.wrike.com/project-management-guide/faq/what-is-scrum-in-agile/>
- WU, Victor, 2018. How to use GitLab for Agile software development. In: *GitLab* [online]. [cit. 2023-03-12]. Dostupné z: <https://about.gitlab.com/blog/2018/03/05/gitlab-for-agile-software-development/>
- XSOLLA, 2023. SELF-PUBLISH ON STEAM: THE ULTIMATE GUIDE. In: *Xsolla* [online]. [cit. 2023-03-05]. Dostupné z: <https://xsolla.com/blog/self-publish-on-steam-the-ultimate-guide>
- YOUSSEF, Ali, 2022. Level Design Basics — Part 1. Pre-Production. In: *GameDev DOU* [online]. [cit. 2023-03-05]. Dostupné z: <https://gamedev.dou.ua/blogs/level-design-basics-part-1/?hl=en>

8 Seznam obrázků, tabulek a grafů

8.1 Seznam obrázků

Obrázek 1: Blocking-out level (nahore) a hotový level (dole)	24
Obrázek 2: Unreal Engine Marketplace.....	26
Obrázek 3: Porovnání High poly a Low poly	28
Obrázek 4: Rozložení modelu do 2D prostoru.....	30
Obrázek 5: Deformace UV mapy.....	30
Obrázek 6: Texel Density	31
Obrázek 7: Texture baking.....	32
Obrázek 8: Substance Painter	35
Obrázek 9: Unreal Engine Skeleton.....	36
Obrázek 10: Unreal Engine Blueprint.....	38
Obrázek 11: Trello - GDD projektu.....	52
Obrázek 12: Blender – High poly (nahore) vs. Low poly model lokomotivy	57
Obrázek 13: RizomUV – UV mapa lokomotivy.....	58
Obrázek 14: Substance Painter – textury lokomotivy.....	59
Obrázek 15: P4V	61
Obrázek 16: Původní nalezená cesta vs. cesta nalezená rozšířeným algoritmem.....	62
Obrázek 17: Implementace A*	63
Obrázek 18: User Widget – menu továrny.....	65
Obrázek 19: Level načtený v hlavním menu (main menu).....	66
Obrázek 20: Hlavní menu prototypu hry	85
Obrázek 21: Grafické nastavení prototypu hry	85
Obrázek 22: Ovládání prototypu hry	86
Obrázek 23: Navigace vlaku v levelu	86

8.2 Seznam tabulek

Tabulka 1: Typy aplikací na platformě Steam.....	46
Tabulka 2: Store Assets	48
Tabulka 3: Community Assets.....	49
Tabulka 4: Library Assets.....	49
Tabulka 5: Event Assets	50
Tabulka 6: Porovnání pojmenování Scrum vs. GitLab.....	53
Tabulka 7: GitLab Labels v projektu prototypu	54
Tabulka 8: Hodnocení skóre SUS.....	70

8.3 Seznam grafů

Graf 1: Finální skóre jednotlivých respondentů.....	69
Graf 2: SUS – Otázka č.1.....	70
Graf 3: Vlastní dotazník – Otázka č.3.....	71
Graf 4: Vlastní dotazník – Otázka č.4.....	71

9 Přílohy

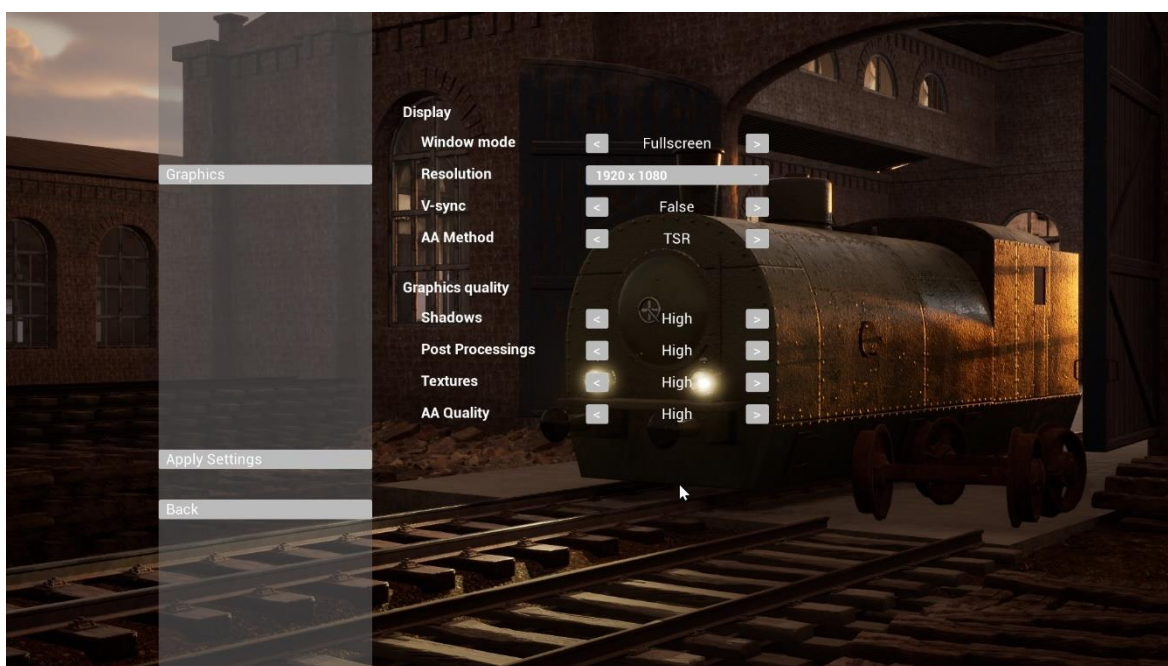
9.1 Snímky z prototypu hry

Obrázek 20: Hlavní menu prototypu hry



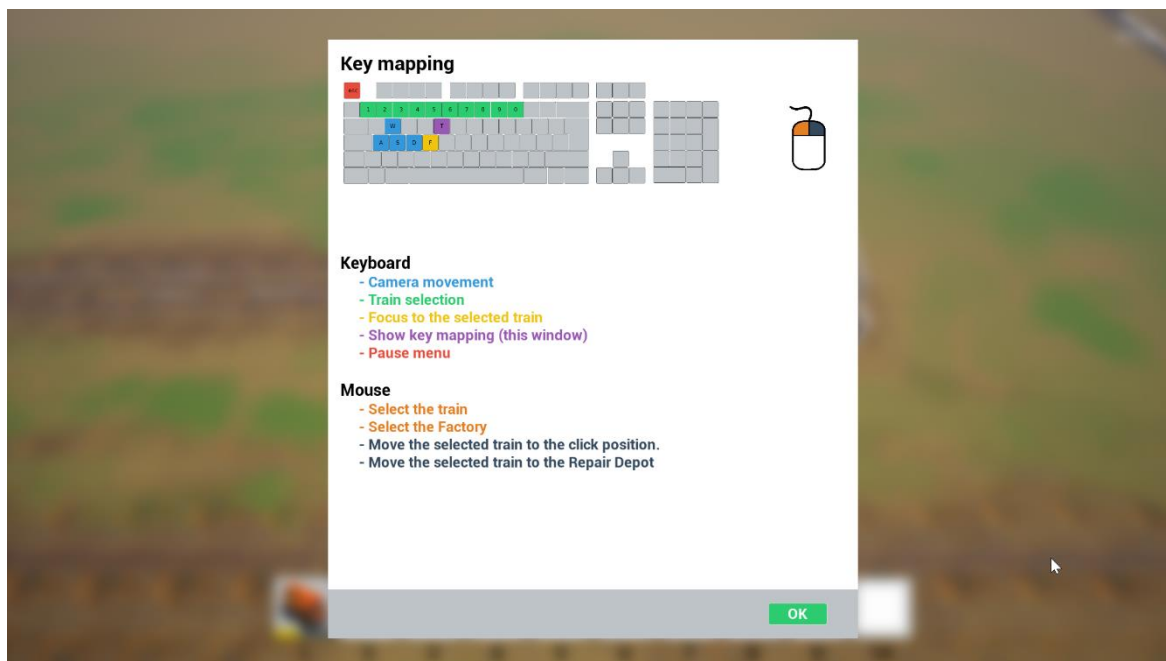
Zdroj: Autor

Obrázek 21: Grafické nastavení prototypu hry



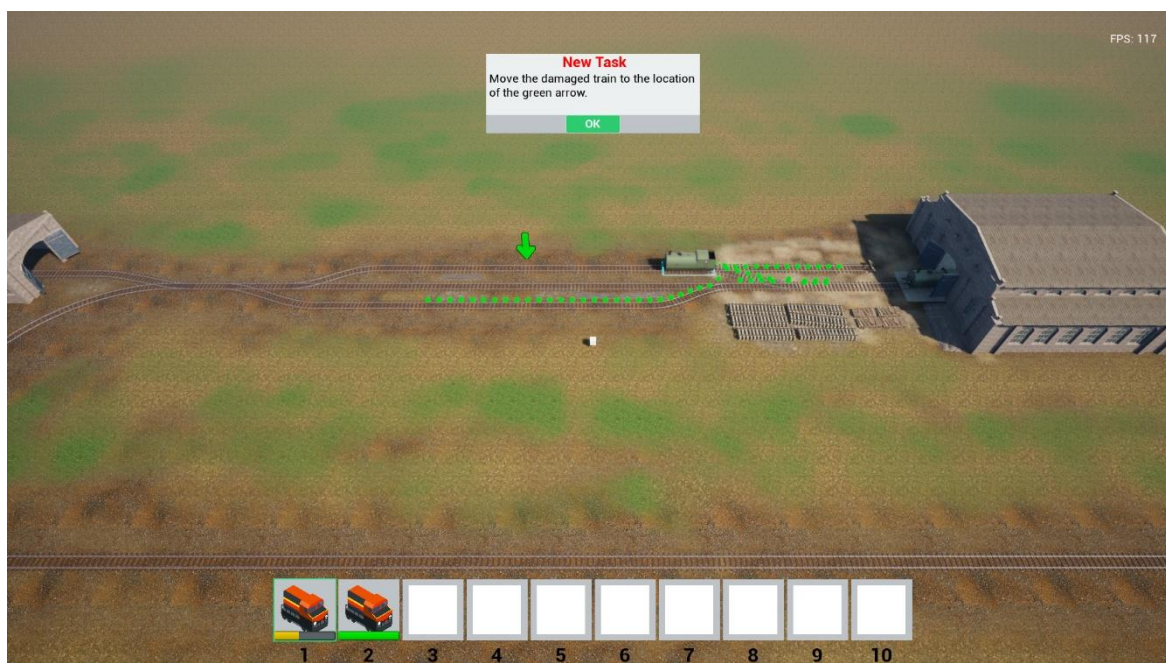
Zdroj: Autor

Obrázek 22: Ovládání prototypu hry



Zdroj: Autor

Obrázek 23: Navigace vlaku v levelu



Zdroj: Autor