



Pedagogická
fakulta
Faculty
of Education

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

**Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta
Katedra informatiky**

Bakalářská práce

Výukově orientovaná 3D hra v prostředí Českého Krumlova

Wypracoval: Aleš Velek, Dis.

Vedoucí práce: Ing. Tomáš Dolanský, Ph.D.

České Budějovice 2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Aleš VELEK**
Osobní číslo: **P11077**
Studijní program: **B7507 Specializace v pedagogice**
Studijní obor: **Informační technologie a e-learning**
Název tématu: **Výukově orientovaná 3D hra v prostředí Českého Krumlova**
Zadávající katedra: **Katedra informatiky**

Z á s a d y p r o v y p r a c o v á n í :

Při zpracování bakalářské práce bude student postupovat následujícím způsobem:

1. Shromáždí informace o možných existujících prostředích pro vývoj 3D interaktivních her a metodách tvorby 3D hry a provede jejich základní porovnání.
2. Student ve spolupráce s odborníky na historii města vybere vhodný příběh a vytvoří scénář hry s výukovými prvky. Hra bude mít za úkol zábavnou formou přiblížit dětem ve věku 6-16 let historický vývoj Českého Krumlova, stavební slohy, šlechtu v Českém Krumlově a pověsti, báje a strašidla na starém městě.
3. Student vytvoří na základě existující části 3D modelu historického jádra v České Krumlově hru ve zvoleném herním prostředí, doplní scénu o další prvky a nastaví metody interakce uživatele s prostředím.
4. Výslednou hru publikuje ve spolupráci s občanským sdružením KRAJINAK a prakticky otestuje na vybraném vzorku cílové skupiny a zhodnotí dosaženého výsledku. Cíle práce: Cílem práce je vytvoření 3D hry pro děti ve věku 6-16 let. Hra bude zábavnou formou přibližovat historii města.

Rozsah grafických prací: CD ROM

Rozsah pracovní zprávy: 40

Forma zpracování bakalářské práce: tištěná

Seznam odborné literatury:

1. ŽÁRA, J. VRML97 - Laskavý průvodce virtuálními světy. Brno: Computer Press, 1999, ISBN 80-7226-143-6.
2. Unity3D student [online]. [cit. 2013-03-10] Dostupné z: <http://www.unity3dstudent.com/>
3. Unity3D answers [online]. [cit. 2013-03-10] Dostupné z: <http://answers.unity3d.com/index.html>
4. CREIGHTON, Ryan Henson. Unity 3d game development by example: Lite: Beginner's Guide. 2010. vyd. 32 Lincoln Road Olton, Birmingham, B27 6PA, UK.: Packt Publishing, 2011. ISBN 978-1-849691-60-4.

Vedoucí bakalářské práce: Ing. Tomáš Dolanský, Ph.D.
Katedra informatiky

Datum zadání bakalářské práce: 16. dubna 2013

Termín odevzdání bakalářské práce: 30. dubna 2014



Mgr. Michal Vančura, Ph.D.
děkan



doc. PaedDr. Jiří Vaníček, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 16. dubna 2013

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb., v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě fakultou elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne.....

.....

Aleš Velek

Poděkování

Rád bych poděkoval svému vedoucímu práce panu Ing. Tomáši Dolanskému, Ph.D. za cenné rady a celkové vedení práce. Dále bych rád poděkoval Mgr. Alexandře Macháčové za spolupráci na tvorbě hry a panu Tomáši Zemanovi za tvorbu zvuků a hudby do hry.

Abstrakt

Dnes se již v herním světě neklade důraz jen na postřeh, nebo logické myšlení hráče, ale i na to, aby si hráč po odehrání hry zapamatoval příběh, který mu hra předá. Vyprávění příběhu ve hře je trochu jiné než ve filmu, ale konečný důsledek může být stejný.

Tato práce „Výukově orientovaná hra v prostředí Českého Krumlova“ je zaměřena na tvorbu hry umožňující volný pohyb po 3D prostředí a na technologii pro tuto práci využitou. Pro programování hry bude využito technologie, která dovoluje tvorbu aplikací ve více platformách. Pohyb hráče bude založen na základních herních prvcích, jako je průlet či chůze. Dále budou tvořeny skripty pro interakci uživatele s prostředím. Pro import celého města bude hledán nejvhodnější formát jak samotných modelů, tak textur. Podle možností zvoleného prostředí pro vývoj bude navržen příběh hry (scénář), který bude korespondovat s historií i současností města Český Krumlov, a bude hráče motivovat v dalším postupu hrou.

Klíčová slova

Spirity, Počítačová hra, hra pro Android, hra pro webový prohlížeč, herní engine, tvorba hry, Unity 3D, Shiva 3D, UnrealEngine, Český Krumlov

Abstract

Today in gaming does not focus only on observation or logical thinking of player, but also to allow a player after playing games remember the story that his game forward in an entertaining way. Telling the story of the game is a little different than in the movie, but the end result may be the same.

This work " Educationally oriented game in Český Krumlov " is aimed at creating the game that allows free movement of the 3D environment and the technology used for this work. For game programming technology will be used Unity3D, which allows the creation of applications in multiple platforms. Moving the player will be based on the basic game elements, such as fly or walk. They will create scripts for user interaction with the environment or texture loading from external sources. To import the entire city will be find the most appropriate format on their own models and textures. By the chosen development environment is designed game story (script) that will correspond with the history and present of Český Krumlov, and player will be motivace to further progress in the game.

Key words

Spirity, PC game, game for Android, game for web broker, game engine, game making, Unity 3D, Shiva 3D UnrealEngine, Český Krumlov

Obsah

1	Úvod	9
1.1	Cíle práce	9
1.2	Východiska práce.....	9
2	Vývoj počítačových her	11
2.1	Herní enginey	12
2.1.1	Popis herních engineů	13
2.1.2	Výčet herních engineů využitelných pro bakalářskou práci	14
	Unreal Engine	14
	Shiva 3D	14
	Unity 3D	14
3	Metodika práce	16
3.1	Technologie pro vývoj multimediálních her.....	16
3.2	Práce s 3D modelem města.....	16
3.3	Programování hry	16
3.4	Export pro cílové platformy.....	17
3.5	Testování hry	17
4	Praktická část.....	18
4.1	Porovnání 3D herních engineů.....	18
4.2	Unity 3D základy	19
4.3	Import modelu města do Unity 3D	23
4.4	Nasazení shaderů na model	24
4.5	Modelování herních objektů	27
4.6	Scénář hry	29
4.7	Animace.....	29
4.8	Zvuky	31
4.9	Programování skriptů	32
4.9.1	Scripty pro odladění výkonu hry	33

4.9.2	Scripty pro dávkovou úpravu objektu	33
4.9.3	Scripty pro ovládání hry	34
4.9.4	Programování GUI (uživatelského rozhraní)	37
4.9.5	Tvorba jednotlivých mini her.....	39
	Hra skořápky	39
	Hra oběšenec.....	40
	Hra házení na cíl	40
	Hra vybarvování polí	40
	Spojování bodů	41
	Posuvná skládačka	42
	Hra stírání omítky	42
	Tančení do rytmu hudby.....	42
4.10	Export hry pro různé platformy.....	43
4.11	Testování hry.....	44
5	Závěr.....	47
6	Použitá literatura	48
7	Seznam obrázků	50
8	Přílohy.....	51

1 Úvod

Hra je interaktivní software, který uživatel (hráč) může ovládat (hrát). Hra s hráčem komunikuje prostřednictvím obrazových a zvukových vjemů. Měla by být formou zábavy, relaxace, uvolnění nebo vzdělávání. Hru je možné spouštět na různých zařízeních jako je počítač, IOS, herní konzole a v dnešní době i nějaké mobilní zařízení, jako chytrý telefon nebo tablet. Distribuce her je možná mnoha způsoby. V dnešní době se jedná zejména o prodej her přes internet (stažení hry přímo z internetu), na Blu-ray nosičích nebo DVD. Kvůli stoupající kvalitě grafiky a objemu dat ve hrách, se zvyšují nároky na hardware zařízení, na kterém se spouští. Vyžadují vysoké frekvence procesoru, grafické akcelerátory a další.[1]

V této práci se zabývám tvorbou 3D hry pro internetový prohlížeč a mobilní zařízení s operačním systémem Android. Tvorba této hry zahrnovala mnoho problémů, které bylo třeba vyřešit. Na začátku práce jsem musel vybrat vhodný herní engine pro tvorbu hry, do něj poté importovat 3D modely potřebné pro hru. Dále jsem modeloval další 3D objekty, které byly potřeba pro doplnění herního prostředí o detaily a funkční prvky a naprogramoval interakci hráče s herními objekty a jednotlivé prvky hry, podle zadaného scénáře. Hru jsem poté publikoval a testoval na cílové skupině hráčů.

1.1 Cíle práce

Populárně naučná hra, lehce dostupná na internetu, která nebude potřebovat výkonný počítač k jejímu spuštění. Celá hra se bude odehrávat v prostředí Českého Krumlova a zábavnou formou předá i informace vztahující se k místu, kde se odehrává. Samotný herní styl by měl u hráče motivovat k dalšímu postupu hrou. Proto je důležitý příběh a herní mechanismy, které se zde budou mísit.

1.2 Východiska práce

Celá tato práce vychází z již známých herních stylů, postupů a jejich různých kombinací. K vytvoření této práce bylo nutné se opírat o historická fakta dočtená z literatury a o technické údaje související se samotnou tvorbou hry v prostředí tomu určeném.

V neposlední řadě se v této práci bude vycházet z již existujícího 3D modelu Českého Krumlova, ve kterém se hra bude odehrávat. Tento model zahrnuje celé vnitřní město, krumlovský zámek a několik dalších významných budov, které se přímo nenacházejí ve vnitřním městě Českého Krumlova. Tento 3D model jsem

převzal z výsledků předchozích bakalářských prací: pana Jaroslava Erharta, který vytvořil 3D model vnitřního města, pana Petra Machta, který ve své práci vytvořil textury pro tento model a pana Pavla Srnského, jehož práce se zabývala tvorbou 3D modelu zámku.

Východiskem budou dále historické podklady fasád budov. Jedná se o dobové fotografie budov nebo o překreslené fasády podle historických informací. Tyto materiály mohou být použity pro porovnání vzhledu budov dnes a dříve.[15]

2 Vývoj počítačových her

Tvorba hry zasahuje do několika odvětví jako je 2D a 3D grafika, programování, ozvučení a další. Je potřeba celý herní svět vytvořit / nakreslit. Poté je nutné popsat logiku hry, jak se dané objekty ve hře budou chovat, jak s nimi hráč bude moci manipulovat a jak se bude celá hra ovládat. Hry mohou mít příběh, proto je nutné napsat ke hře scénář, namluvit dialogy a složit hudbu.[1]

Hry lze dělit podle několika faktorů. První z nich je platforma, pro kterou jsou určeny.

- PC, Mac – Počítačová hra
- Herní konzole – Video hra
- Mobilní zařízení (mobilní telefon, tablet) – mobilní hra
- Webový prohlížeč – Online hra

Dále se hry dělí podle žánrů. Těch existuje mnoho a hlavním rozdílem je styl ovládání hry, její tematické zaměření, náročnost na hraní atd. Žánry her se pak mohou prolínat, nebo jejich spojením vznikat nové. Výpis jednotlivých herních žánrů:

- adventura
 - textovka
 - grafická textovka
 - grafická point and click adventura
 - 3D adventura
 - 3D akční adventura
- akční počítačová hra
 - 3D akční
 - FPS (střílečka z pohledu první osoby)
 - TPS (střílečka z pohledu třetí osoby)
- survival horror
- taktický simulátor
- arkáda
- plošinovka
- bojovka
- logická
- sportovní
- závodní
- hry typu arkanoid
- strategie

- tahová strategie (TBS)
- reálnodobá strategie (RTS)
- budovatelská strategie
- tower defense (TD)
- simulátor
 - simulátor života (The Sims, Second Life atp.)
 - letecký simulátor
 - virtuální bojiště (simuluje bojiště složené nejméně ze dvou složek armád – pozemní, letecké, námořní)
 - závodní
- desková hra (proti počítači nebo online po síti)
- sportovní hra
- sportovní manažer
- hra na hrdiny (RPG)
 - akční RPG, „diablovky“ - akce a vývoj postavy má přednost před příběhem
 - epické RPG - příběh je velice silnou stránkou hry
 - MMORPG (Massive multiplayer online role play game – masivně multiplayerová online hra na hrdinu)
- nezávislé (např.: Minecraft, Terraria, atd.) [1]

Obrat v Herním průmyslu se dnes pohybuje v řádech miliard dolarů. Vývoj hry bývá časově, finančně a personálně náročnou prací. Může trvat i několik let a stát milióny dolarů. Herní průmysl se v zásadě skládá ze dvou subjektů a to z vývojářů a vydavatelů. Vývojáři hru tvoří a vymýšlí. Vydavatelé hru financují a poté ji distribuují prodejším. [2]

Hry se dnes vyvíjí za pomoci programů k tomu určených. Jedná se o takzvaný herní engine. Často se jedná o komplexní nástroj, pomocí kterého lze hru vytvořit a který slouží jako motor celé hry (engine). Herních enginů existuje celá řada a budu se jimi zabývat podrobněji níže v mé práci.

2.1 Herní enginy

Engine (z anglického slova „motor“), je v počítačový termín pro jádro nějakého programu, hry, databáze. Může se jednat o sadu knihoven pro nějaký programovací jazyk nebo o komplexní program s uživatelským rozhraním. Vývoj hry je často komplexní a časově náročná činnost. Vývojář se musí rozhodnout, zda využije již hotové herní engine, nebo si vytvoří vlastní. Vytvořit vlastní herní engine

bývá finančně, personálně a časově velmi náročné. Proto se většina herních vývojářů uchyluje k výběru již vytvořených herních engineů, které značně zjednoduší vývojáři práci a ušetří čas. Většinu herních engineů si vývojáři mohou vyzkoušet zdarma, některé z nich mohou použít v komerční sféře. Jejich obsah je však omezen o některé funkce a pro tvorbu složitější hry je nutné si engine dokoupit v plné verzi. **Chyba! Nenalezen zdroj odkazů.**

V aktuální době se na trhu vyskytuje celá řada herních engineů lišících se hlavně v licenčních ujednáních. Jedná se především o licenci, která je zdarma pro nekomerční použití a pro komerční musí vývojář odvádět část zisku. Při koupi plné licence je možné takto dostat od vývojářů přístup do kódu engineu a tím si ho poupravit podle vlastních potřeb.[4]

2.1.1 Popis herních engineů

V herním engineu se pracuje často ve dvou krocích. V prvním se navrhne rozmístění objektů po scéně a jejich vykreslování a v druhém se určí pomocí skriptu, jak se budou za běhu hry chovat. Vzhled jednotlivých objektů se určí pomocí materiálů. Většina herních engineů obsahuje editor materiálů, ve kterém je možné nastavit jeho vlastnosti jako hrubost, průhlednost atd. Dále je kladen důraz na jeho optimalizaci, tzn., jak bude například engine reagovat na kritické vytížení hardwaru počítače. Je také důležité na jaké platformy je možné hru vyvíjet.

Ve hrách mohou nastat náročné situace pro hardware a engine na ně musí zareagovat a upravit například vlastnosti fyzických výpočtů, které mohou být při nízké snímkovací frekvenci vysoce nepřesné.

Skriptovací jazyk je důležitou součástí každého engineu. Jedná se o jazyk, kterým se popisuje logika hry. Nejčastěji se používá C++, Javascript, C#, Boo atd. Bez psaní kódu není možné vytvořit rozsáhlejší hru. Kódem se popisuje ovládání hry (vstup uživatele), stavy ve kterých se mohou objekty nacházet atd. Je možné si do některých engineů doinstalovat grafický skriptovací jazyk, který je vhodný pro začátečníky. Pro úplné začátečníky existují již hotové nástroje, které jsou připraveny pro určitý druh herního mechanismu, a vývojář pouze nastavuje parametry daného nástroje.

Herní trh se vyvíjí velice rychle a na trhu se objevují nové funkce (především grafické). Proto je většina engineů pravidelně aktualizovaná. Nejčastěji se jedná o vykreslování různých odlesků, fyzika osvětlení scény atd.

2.1.2 Výčet herních enginů využitelných pro bakalářskou práci

Unreal Engine

Je sofistikovaný a komplexní program s rozsáhlou uživatelskou komunitou využívající skriptovacího jazyku C++. Je poskytován zdarma pro nekomerční účely. Pro vývoj her s vidinou zisku je nutné si za příplatek dokoupit plnou licenci programu a odvádět část zisku ze hry. Mezi jeho přednosti patří propracované grafické efekty, editor terénu a vegetace a detailní generátor částicových systémů. Tento engine je na herním trhu velice populární díky titulům, které v tomto enginu byly vyrobeny, jmenovitě: Dead Space, Unreal Tournament, Mass Efekt, atd. Platformy, na které lze s tímto enginem vyvíjet, jsou následující: Windows PC, Mac, iOS a Android. [6],[8]

Shiva 3D

Tento engine je primárně určen pro tvorbu her pro webový prohlížeč a mobilní zařízení. V poslední době byla rozšířena i na ostatní platformy jako jsou desktopové aplikace pro Windows, Mac a Linux. Jedná se o populární herní engine často porovnávaný s Unity 3D. Shiva je hojně využívána pro tvorbu 3D her na mobilní zařízení a ve webových prohlížečích. Licenčně vychází ve třech verzích: free, basic, advanced. První z těchto verzí je zdarma. Za další dvě je nutné si připlatit a to od 200 až do 2000 dolarů. Ve verzi zdarma je možné tvořit hry pouze pro webový prohlížeč.

Jazyk pro tvorbu skriptů je zde C++. Pro render je zde využíván OpenGL, OpenGL ES nebo Direct 3D. Dále také podporuje standardní pluginy jako je Nvidia PhysX, F-Mod a ARToolKit. [10]

Unity 3D

Unity 3D je dnes podle Gamasutra.com jeden z nejpoužívanějších herních enginů. Má širokou základnu uživatelů a rozsáhlé fórum, ze kterého se vývojář může mnohé dozvědět a novému se naučit. Dále jsou zde webové stránky Unity 3D, které obsahují obchod tzv. Asset store. Zde je možné najít již hotové demo projekty, 3D modely, shadery, textury, zvuky, atd. Dále je zde možné jak nakupovat, tak i prodávat své vlastní práce. Další výhodou je široké spektrum platform, pro které lze s Unity 3D vyvíjet hry.[16]

Licenčně je program rozdělen na Unity 3D a Unity 3D pro. Unity 3D je pro komerční i nekomerční projekty zcela zdarma. V placené verzi Unity se především

setkáme s lepšími nástroji na tvorbu grafických efektů a nástroji na odlaďování výkonu hry. Pro tvorbu logiky hry jsou zde 3 jazyky a to C#, JavaScript a Boo. [11]

3 Metodika práce

První částí mé práce bude popsat a vybrat vhodný herní engine pro tvorbu hry dle zadání. Výběr vhodného herního engine je velice důležitý. Je potřeba zahrnout všechny faktory a cíle, které budou na hru kladeny. V rozpracované hře je pozdější přechod mezi enginey velice obtížný, mnohdy zcela nemožný.

3.1 Technologie pro vývoj multimediálních her

Pro porovnávání herních engineů použiji faktory jako: cílové platformy, pro které lze hru s engineem vytvořit, licenční ujednání a skriptovací jazyk, kterým bude herní engine disponovat. Platformy budou důležitou součástí výběru engineu, jelikož hra by měla být vytvořena pro internetový prohlížeč a pro mobilní zařízení s operačním systémem Android. Licenční ujednání zahrnují do výběru z důvodů jednoduché dostupnosti engineu a volného šíření her na něm vyvinutých. Skriptovací jazyk bude také důležitým prvkem výběru. Z důvodů časové náročnosti na tvorbu hry se budu více zaměřovat na programovací jazyky, se kterými mám větší zkušenosti. Pro vybraný herní engine popíši podrobnější postup při tvorbě her, jeho základní vlastnosti a uživatelské rozhraní.

3.2 Práce s 3D modelem města

Pro 3D model města Český Krumlov zvolím vhodný formát pro převod do vybraného herního engineu. Zdrojové soubory k modelu jsou ve formátu programu SketchUp. Pokud zvolený herní engine tyto soubory nebude umět zpracovat, budu hledat vhodný formát pro převod 3D modelu do herního engineu. U výběru vhodného formátu pro převod modelu budu dbát na správné zachování textur na modelu, správné natočení polygonů a koncovou velikost souboru.

3.3 Programování hry

Po zpracování 3D modelu města naprogramuji interakci hráče s prostředím podle zadaného scénáře a další mechanismy potřebné pro cílovou platformu. Pro programování budu využívat skriptovacího jazyka, který mnou vybraný herní engine bude podporovat. Dále bude potřeba naprogramovat skripty pro ovládání hry. Hra se bude ovládat především pomocí myši a klávesnice nebo dotyky na dotykovém displeji. Následně bude třeba přidat skripty pro interakci hráče s prostředím jako je:

sběr předmětů do inventáře, spouštění miniher, sbírání informací o objektech, na které hráč během hraní narazí apod.

3.4 Export pro cílové platformy

Hru posléze vyexportuji pro zvolené platformy. Ze zadání vyplývá, že se bude jednat o Internetový prohlížeč a mobilní zařízení s operačním systémem Android. Tato hra bude volně dostupná na internetu ke stažení či odehrání online přímo v internetovém prohlížeči.

3.5 Testování hry

Dále hru otestuji po stránce technologické, zda hra běží stabilně na cílových platformách. Pomocí měření FPS (frames per second) snímků za sekundu při hraní hry na zvolené platformě. Testování proběhne i v rovině hratelnosti hry a to na vybrané věkové kategorii hráčů pomocí dotazníkového šetření, a osobním pozorováním hráčů při hraní hry. Pomocí dotazníku budu zjišťovat, zda hra vyhovuje zadání po stránce doby hratelnosti, jak dlouho hra trvala hráčům v průměru odehrát, zda hra naučila hráče něčemu novému, a celkové dojmy ze hry.

4 Praktická část

V praktické části jsem se zabýval výběrem herního engine pro tvorbu samotné hry. Při vybírání herního engine jsem se zaměřil na požadavky na hru v zadání. Hra by měla běžet ve webovém prohlížeči, a na mobilních zařízeních s operačním systémem Android. Dále jsem řešil importování 3D modelu města Český Krumlov do zvoleného engine. Nejdůležitější součástí mé praktické části bylo naprogramovat logiku hry. Bylo potřeba vyřešit ovládání hry hráčem a interakci s objekty. Nakonec bylo nutné hru otestovat na cílové skupině hráčů a podle odezvy hru upravit.

4.1 Porovnání 3D herních engineů

Popisu herních engineů jsem se dostatečně věnoval v kapitole 2.3.2 *Výčet herních engineů*.

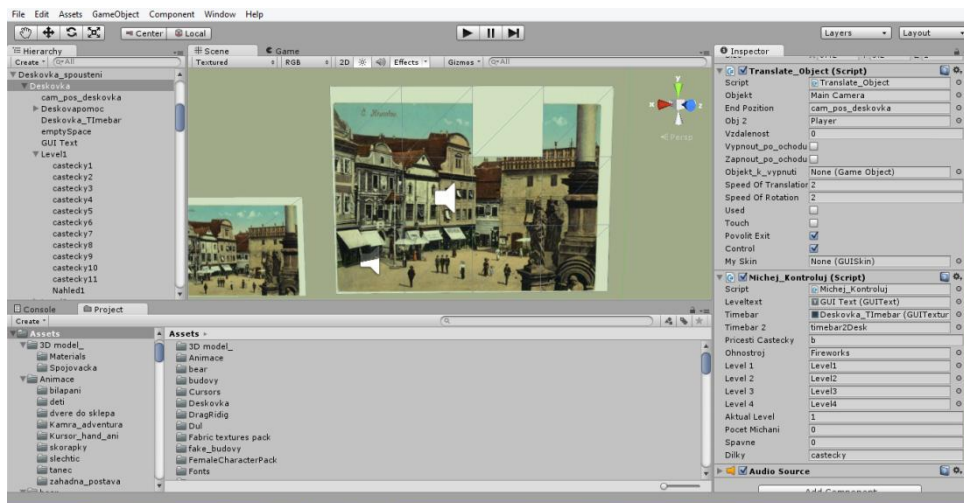
Pro svoji práci jsem zvolil do užšího výběru pouze ty enginey, které jsou zdarma a které jsou mezi vývojáři her vysoce populární z důvodu vysoké podpory ve fórech na internetu. Do širšího výběru se tedy dostaly enginey: Shiva 3D, UnrealEngine a Unity 3D.

Poté jsem zjišťoval, který z těchto engineů podporuje mnou zvolené platformy (web player, Android). Po důkladném porovnání všech 3 engineů jsem zvolil Unity 3D. Unity 3D, v porovnání s UnrealEngine a Shiva 3D, podporuje tvorbu her pro webový prohlížeč a pro Android již ve verzi, která je zdarma.

Název	Platformy	Licence	Programovací jazyk
Unreal Engine	Windows PC, Mac, iOS a Android	Free - pouze nekomerčně	C++
Shiva 3D	PC, Mac, Linux, iOS, Android, BlackBerry, Bada, Symbian, Marmalade, Windows Phone	free, basic, advenced	C, C++, Objective C
Unity 3D	PC, Mac, Linux, Android, Windows Phone, IOS, Playstation, Xbox, BlackBerry, Webplayer	Free, Pro	C#, JavaScript, Boo

4.2 Unity 3D základy

Unity 3D je opatřeno přívětivým uživatelským rozhraním, které je možné libovolně upravovat přesouváním jednotlivých oken. Jednotlivé okna lze z mřížky oddělit a pracovat tak na více monitorech. První, čeho by si uživatel měl všimnout, je panel nástrojů. Nachází se zde volba pro přesouvání kamery pomocí kliku a táhnutí myši, nástroj pro přesouvání objektu po scéně, nástroj pro rotaci objektu a nástroj pro změnu velikosti.

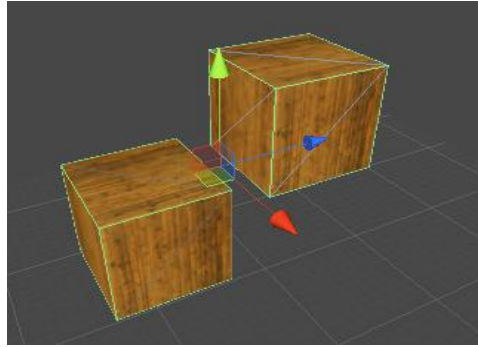


Obr. 1 GUI Unity 3D

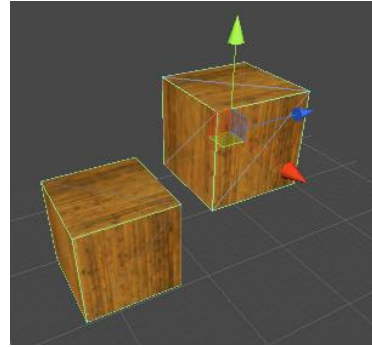


Obr. 2 Panel nástrojů

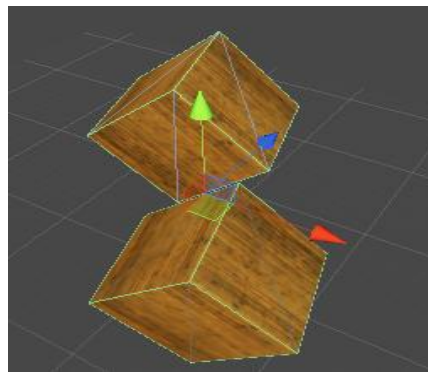
Veškerá manipulace s objekty se provádí za pomoci os. Jedná se o pomocné prvky ve scéně. Například při přesouvání objektu po ose X se používá nástroj pro přesun objektu a kliknutím a tažením na danou osu se objekt přesouvá po námi zvolené ose. Dále jsou zde tlačítka „Pivot / Center“ a „Local / Global“. Volba pivot se po kliku změní na „center“ a zajišťuje polohu os vůči objektu. Volba „pivot“ znamená, že se osy objektu budou protínat v jeho původní nulové pozici oproti tomu volba „center“ protne osy ve středu objektu. Druhá volba „Local / Global“ určuje natočení os vůči objektu. „Local“ natáčí osy podle natočení objektu a „Global“ je zanechává kolmo ke scéně. To znamená, že při volbě „Global“ budou u všech objektů osy natočené stejným směrem.



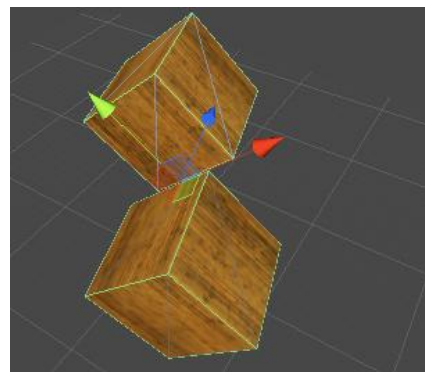
Obr. 3 Center



Obr. 4 Pivot



Obr. 5 Global



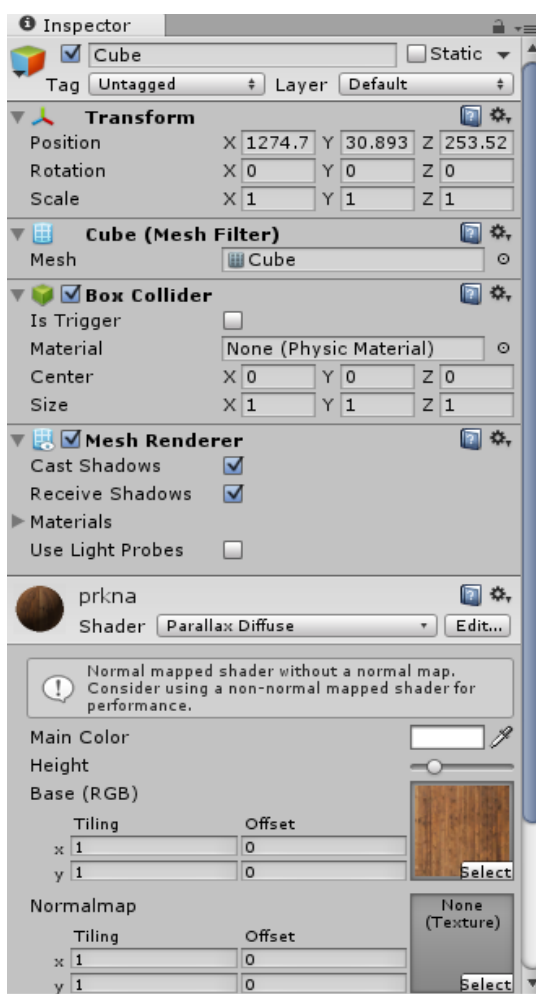
Obr. 6 Local

Dále je zde okno „*hierarchie*“. V tomto okně je výčet všech objektů ve scéně a jejich hierarchická spojitost. Pomocí přesunu jednotlivých položek na jiné je možné určovat rodiče a děti objektů. Je zde také možné objekty přejmenovávat a podle jmen vyhledávat.



Obr. 7 Okno hierarchie

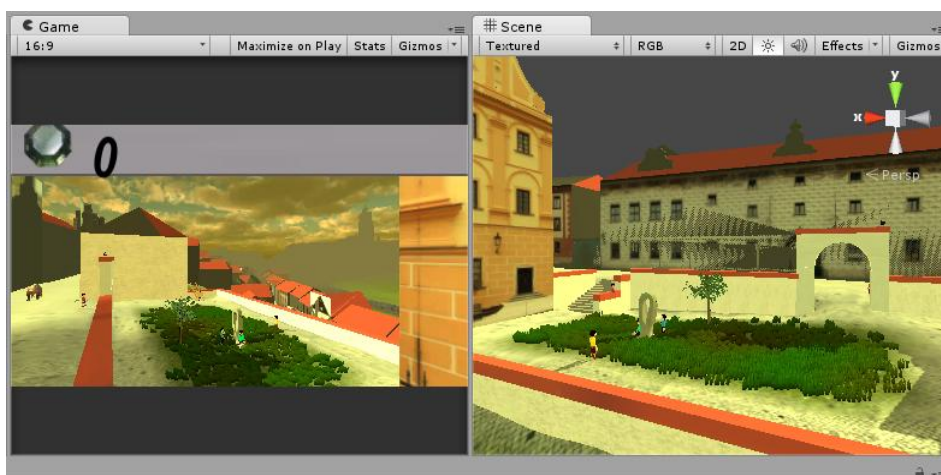
Dalším důležitým oknem je Inspektor. V tomto okně se zobrazují veškeré vlastnosti / komponenty právě aktivního objektu. Jako první lze spatřit základní informace o objektu jako je jeho jméno, tag, v jaké se nachází vrstvě a jestli se jedná o statický objekt. Jméno objektu slouží k jeho identifikaci mezi ostatními objekty. Tag slouží pro přidávání objektů do pomyslných skupin jako například strom, nepřítel, položka inventáře atd., nebo pro označení důležitých objektů jako je například hráč, kamera, začátek mise atd. Nejčastěji je tak používán pro identifikaci objektů ze skriptu. Dále zde nalezneme vrstvy. Vrstvy se nejčastěji používají při vykreslování scény kamerou. Přiřazením objektů do určité vrstvy můžeme později říci kameře, aby je nevykreslovala nebo světlu, aby od nich nevrhlo stíny. Dále se vrstvy mohou použít na rozdělení interakce fyzikálních vlastností. Kdy například objekty v jedné vrstvě nebudou fyzikálně interagovat s objekty z druhé apod. Další základní informací o objektu zda je statický. Pomocí zaškrtnutí políčka static určujeme, zda je objekt statický či nikoliv. Statické objekty se používají při automatickému vypočítávání cesty mezi objekty v scéně, nebo pro výpočet statických stínů.



Obr. 8 Inspektor

V inspektoru lze manipulovat s komponentami / vlastnostmi objektu. Mezi nejzákladnější komponenty patří „*Transform*“. Tato komponenta určuje polohu, rotaci a velikost objektu na scéně. Dále je zde komponenta „*Mesh filter*“. Ta určuje tvar objektu a s kombinací s komponentou „*Mesh Renderer*“ je model vykreslován a stínován na obrazovce. Komponenta „*Mesh Renderer*“ dále zajišťuje pokrytí objektu texturou za pomoci materiálu. V okně inspektoru se mohou objektu přidávat další komponenty a tím rozšiřovat jeho vlastnosti.

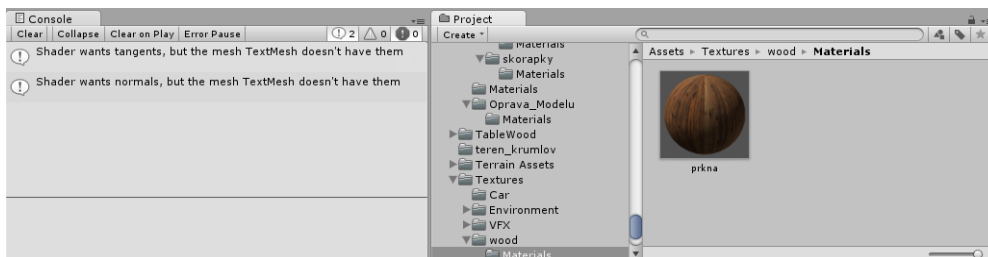
Dále jsou zde okna pro vykreslování scény ve dvou módech. První mód je editační pro manipulaci s objekty a celkové rozvržení scény. V druhém módu je náhled hry, kde je možné pozorovat výsledek vyvíjené hry. Aby bylo možné pozorovat náhled hry je nutné do scény přidat kameru a nastavit její polohu ve scéně. Variabilita oken umožňuje seřadit editační a náhledové okno vedle sebe a tak manipulovat s objekty za běhu hry a hned pozorovat výsledek.



Obr. 9 Náhled hry a editační okno

V poslední řadě jsou zde okna se souborovým průzkumníkem a vývojářská konzole. V průzkumníku lze prohlížet a přidávat soubory do scény, a v konzole se zobrazují chybové hlášky nebo do ní lze přímo vypisovat pomocné výsledky skriptů.

Zdrojové soubory jako jsou obrázky, zvuky a 3D objekty potřebné pro tvorbu hry se nazývají „*Assety*“. Je možné ve scéně vytvořit několik objektů s jejich vlastnostmi a skripty a poté je zabalit do balíku. Tento balík je pak možné importovat do různých scén nebo projektů.



Obr. 10 Konzole a souborový průzkumník

Objekty se na scénu přidávají několika způsoby. První je přetažení souboru s objektem z průzkumníka souborů do scény, nebo do hierarchického okna. Pokud přesouváme asset přímo do scény, objeví se v poloze vůči myši. Pokud ho přesuneme do hierarchického okna, zobrazí se na scéně v nulových souřadnicích. Dalším způsobem je přidání objektu z menu `GameObject / Create Other`. Zde se nachází výčet primitiv (základních objektů) jako je krychle, plocha, koule, generátor částic atd. Kliknutím na jeden z nich se přidá do scény.

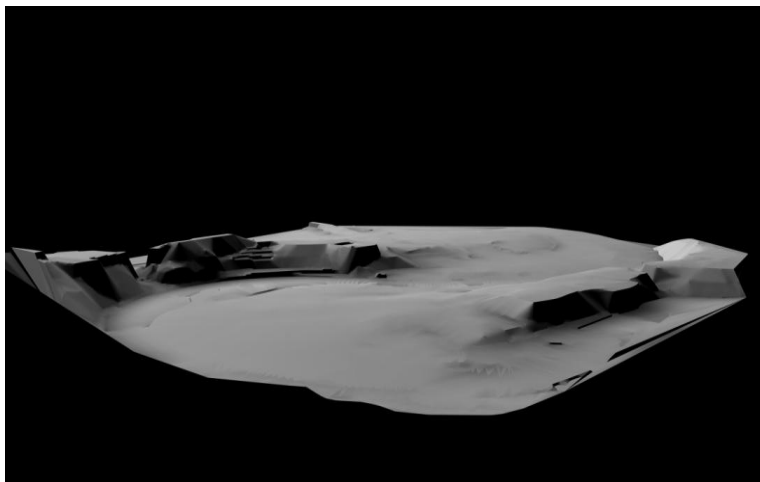
4.3 Import modelu města do Unity 3D

Na úplném začátku jsem dostal soubor 3D modelů města Český Krumlov. Jednalo se o celé vnitřní město, kde byly některé budovy vymodelované s většími detaily než ostatní. Dále mi byl poskytnut detailní model Českokrumlovské Synagogy a Českokrumlovského zámku. Všechny budovy měly formát „.skp“, což je formát pro program *SketchUp*. Pomocí tohoto programu jsem hledal vhodný formát pro import do Unity 3D. Po několika pokusech jsem zvolil formát „.dae“ z důvodů zachování všech textur na každé budově. Tento formát nevyhovoval všem budovám, proto jsem zvolil další formáty, a to „.fbx a .3ds“.

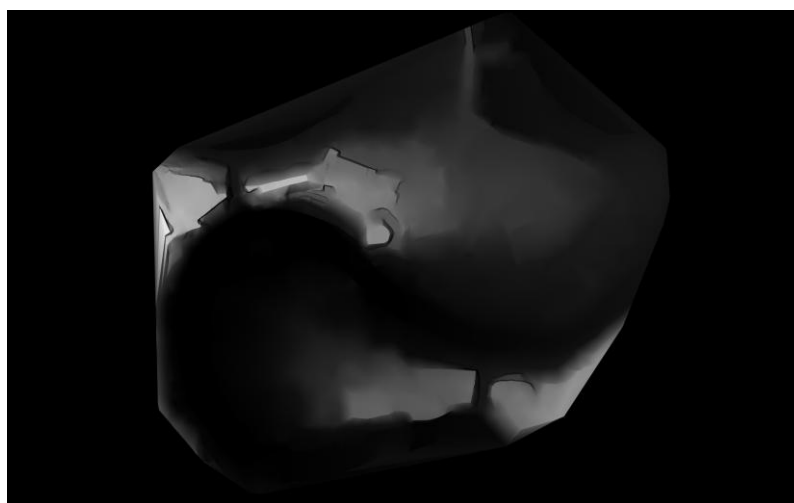
Problémem byla orientace polygonů. Jelikož Unity 3D vykresluje polygony pouze z vrchní strany, musel jsem je dorovnat ručním natočením. Dále jsem upravoval textury na budovách a jejich nanesení.

U terénu, na kterém stojí celý model, jsem zvolil rozličnou metodu importu. Unity 3D obsahuje svůj vlastní nástroj na tvorbu terénu. Konkrétně se jedná o tvorbu terénu pomocí výškových map. Výšková mapa je bitmapový obrázek ve stupních šedi, kde zbarvení určité části obrázku určuje výšku terénu. Čím světlejší tím vyšší a naopak. Proto jsem potřeboval převést 3D model terénu na výškovou mapu. To jsem provedl pomocí barevného přechodu ve směru osy Y aplikovaného na tento model. Po vyexportování pohledu shora, bez perspektivy, jsem získal výškovou mapu. Unity 3D pro formát výškových map zvolilo formát „.raw“, do kterého jsem

výškovou mapu uložil. Dále jsem potřeboval zjistit výškové rozmezí terénu, abych mohl v Unity 3D určit, jak vysoko mají být bílé části obrázku od těch černých. Poté jsem založil nový terén, nastavil rozlišení výškové mapy a naimportoval obrázek. Viz obrázky níže.



Obr. 11 3D objekt terénu



Obr. 12 Výšková mapa terénu

4.4 Nasazení shaderů na model

Shardery jsou malé skripty (programy), které určují, jak se daný objekt ve scéně bude vykreslovat na obrazovku. Veškeré vykreslování v Unity 3D je zpracováno za pomoci těchto shaderů. Zajišťují ovlivnění objektu světlem a stíny, povrchové vlastnosti objektu jako je hrbolatost, odrazivost, průhlednost, atd.

Pro moji hru jsem nejčastěji používal *shader alpha průhlednosti*. Jedná se o zneviditelnění průhledné části obrázku (textury). Nejčastěji bylo nutné tento shader použít u budov, které měly složitěji tvarovaný vikýř. Textury k těmto budovám měly vrstvu průhlednosti v místech, kde končily kontury budovy. Shader umožňující tento jev, se v Unity 3D nachází ve skupině shaderů pod názvem *Transparent*, v podskupině *Cutout*. Na následujících obrázcích (*Obr. 13 a Obr. 14*), můžeme vidět použití shaderu v praxi.



Obr. 13 Diffuse

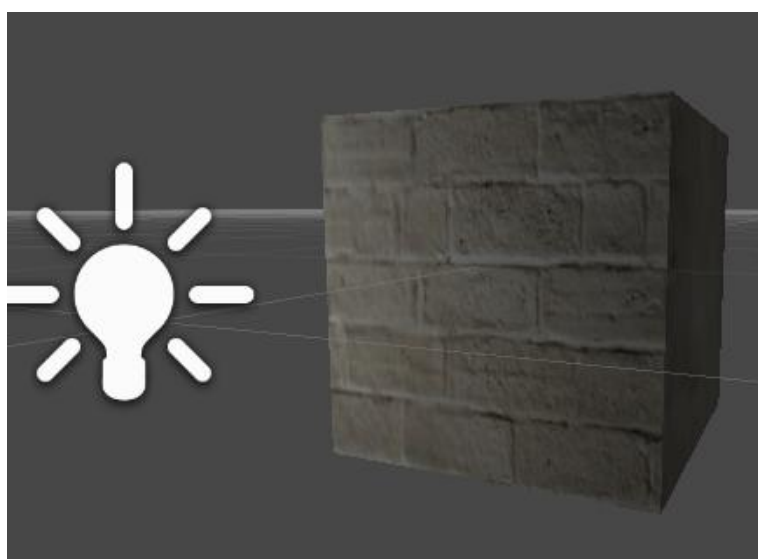


Obr. 14 Cutout Diffuse

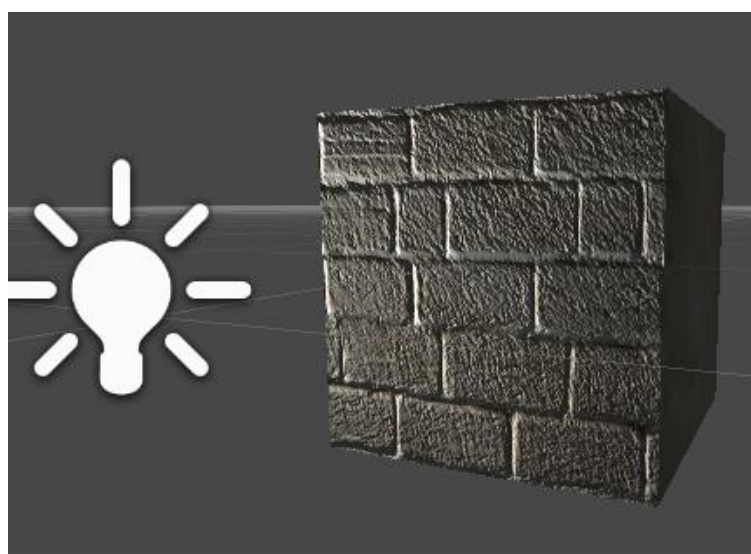
Dalším shaderem, který jsem použil, byl „*Bumped Diffuse*“. Jeho hlavní výhodou je tvorba iluze plastičnosti jinak plochého objektu, za pomoci textury a výškové mapy. Textura v tomto případě udává barvu objektu a výšková mapa jeho hrboľatost (hloubku). Nejčastěji jsem ho používal na členité textury, jako jsou cihlové stěny, nebo na vyjádření tvaru objektu bez jeho modelování.

Dalším častým shaderem byl *Diffuse*. Tento shader nanese pouze texturu na objekt a bere v potaz nasvícení objektu.

Na obrázcích můžeme vidět rozdíl mezi shaderem *Diffuse* a *Bumped diffuse*. Je zde také patrný zdroj světla, podle kterého se určuje vystínování textury.

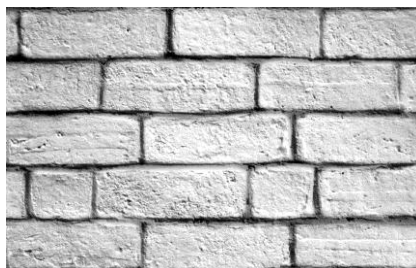


Obr. 15 Diffuse



Obr. 16 Bumped Diffuse

Na tyto dva obrázky jsem použil dvě textury. Obrázek 17 je výšková mapa ve stupních šedi, použitá pro zvýraznění hloubky textury a na obrázku 18 je textura vyjadřující barvu objektu.



Obr. 17 Výšková mapa



Obr. 18 Textura

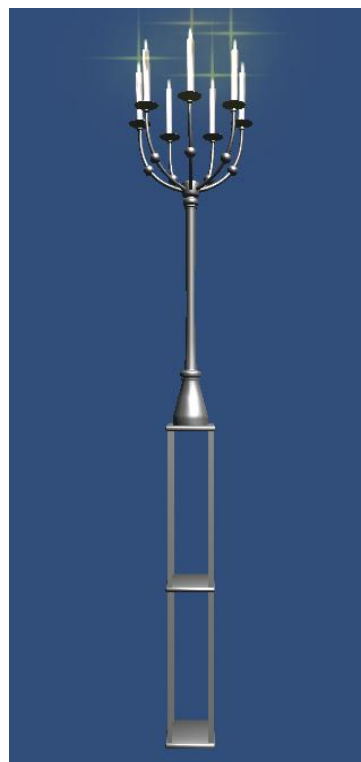
4.5 Modelování herních objektů

3D modely které mi byli poskytnuty obsahovali otexturované budovy. Proto bylo nutné doplnit herní svět o další 3D objekty spojené s hrou. Některé jsem musel vymodelovat, jiné jsem stáhnul z internetu. U objektů stahovaných z internetu bylo nutné dbát na jejich licenci užití, abych je mohl komerčně využívat zdarma. Jednalo se hlavně o objekty postav lidí a zvířat.

Pro modelování 3D objektů jsem používal program Blender. Vytvářel jsem objekty doplňující prostředí o některé detaily. V synagoze chyběl oltář, některá zábradlí, řečnický pult a dva druhy sedmiramenných svícňů včetně svící. Dále bylo potřeba vymodelovat sochu se zvonečkem, která se nachází v Českém Krumlově na vyhlídce. Jako poslední jsem tvořil model pětilisté růže, který byl využit pro jednu z miniher. Při modelování objektů bylo nutné dbát na nízký počet polygonů, aby zbytečně nevytěžovali hardware zařízení, na kterém bude hra spouštěna.



Obr. 17 Svícen



Obr. 20 Vysoký svícen



Obr. 21 Synagoga interiér

4.6 Scénář hry

Scénář je založen na skutečné události vyloupení synagogy neznámou osobou. Jistá osoba si vypůjčila klíče od synagogy a odnesla z ní doposud neznámé předměty. Že byla synagoga vykradena, poznali až dělníci, kteří v synagoze druhý den objevili tajné schránky, které byly otevřené a prázdné. Scénář jsem dostal zpracován od A. Macháčové a podle něj vypracoval technický scénář jednotlivých scén. Jednalo se především o rozvržení jednotlivých herních objektů po scéně, nasvícení scény a poloha kamery. [[10]]

Ve hře se tedy jedná o dopadení této zlodějky, která okrádá hráče a některé herní postavy. Hráč jim pak musí pomáhat nebo naopak herní postavy pomáhají hráči. Herní postavy jsou tvořeni především dětmi.

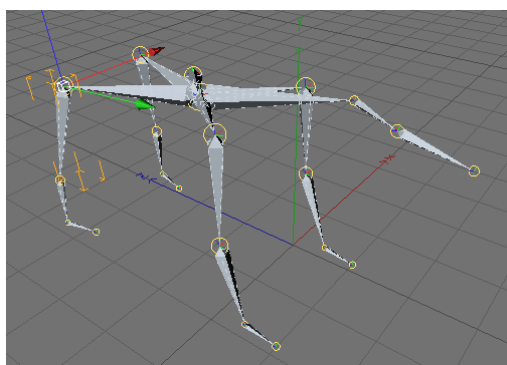
4.7 Animace

Aby herní svět nebyl příliš statický, bylo třeba do hry přidat animace postav a dalších objektů. Ve 3D grafice se animací rozumí měnící se poloha a vlastnosti objektu v čase. Nejčastěji se pro animaci ve 3D používá metoda „*keyframing*“. Tato metoda je založena například na určení klíčových bodů polohy objektu v určitém čase a počítač poté vytvoří mezi nimi plynulý přechod. Takto lze ovládat i vlastnosti jednotlivých objektů. Pokud určím, že v čase 0 s bude intenzita světla 100% a v čase 10 s bude 0%, počítač vytvoří mezi těmito body plynulý přechod tak, že se intenzita světla bude plynule snižovat po dobu 10 s. Vlastnosti přechodu se mohou lišit podle nastavení jednotlivých klíčových bodů. Přechod může být buď lineární, nebo se na začátku či na konci animace může lehce zpomalit. Animací se také rozumí pohyb vypočítaný na základě fyzikálních jevů. Unity 3D obsahuje komponentu na výpočet fyziky pevných těles. Pohyb objektu je zde závislý na nastavení jeho fyzikálních vlastností. Poté je určena jeho počáteční pozice a jeho další pohyby jsou vypočítávány podle interakce s prostředím. Může se jednat například o sud, do kterého hráč může narazit a převrátit ho, a on se skutálí z kopce a naráží cestou do překážek. [14]

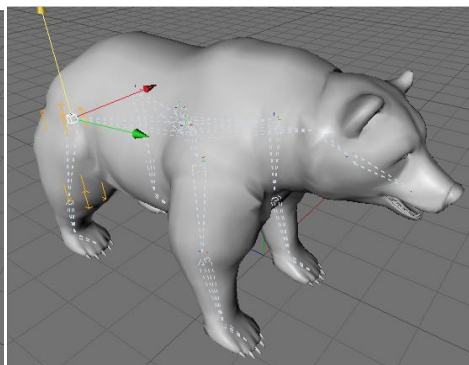
Unity 3D dále obsahuje nástroj na tvorbu animací. Mohu zde určit polohu nebo vlastnosti jednotlivých objektů v čase. Každý objekt, který má vykonávat animaci má v inspektoru komponentu „*Animation*“. Ta zajišťuje spouštění a načítání animací. Každá animace se v Unity 3D ukládá jako externí soubor s příponou „*.anim*“. Tento soubor obsahuje jména objektů a jejich pozice v klíčových bodech. Dále může být nastaven typ přehrávání animace jako například přehrávání ve smyčce, přehrávání pouze jednou, apod. Animace lze také spouštět a míchat pomocí

skriptů. Třída „Animation“ obsahuje celou řadu metod na práci s animacemi jako Play(), Stop(), Blend() apod. Velice užitečná je metoda Blend. Tato metoda vytváří plynulý přechod mezi různými animacemi.

Pro animaci postav se v Unity 3D využívá tzv. „kostry“. Stejně jako u živých organismů i zde je nutné pro pohyb 3D postavy využít kosti. Jedná se o objekty, které jsou spojené s určitou částí polygonální sítě objektu, kterým pohybují. Často jsou hierarchicky spojené s ostatními kostmi. To znamená, že pokud pohnu ramenní kostí, pohne se i zbytek paže, jako je loketní kost, prsty atd. Unity 3D neobsahuje nástroj na tvorbu kostí. Bylo proto nutné dohledávat modely postav již s kostrou, nebo takovou kostru vyrobit v 3D programu jako je Blender nebo Cinema 4D. V mé práci se jednalo o model medvěda. Tento model neobsahoval kostru, a proto bylo nutné jí vytvořit. Pro tvorbu kostí jsem zvolil program Blender. Nejdříve bylo nutné podle modelu vytvořit kostru a poté jí uchytit na model viz Obr. 22 a 23.



Obr. 22 Kostra



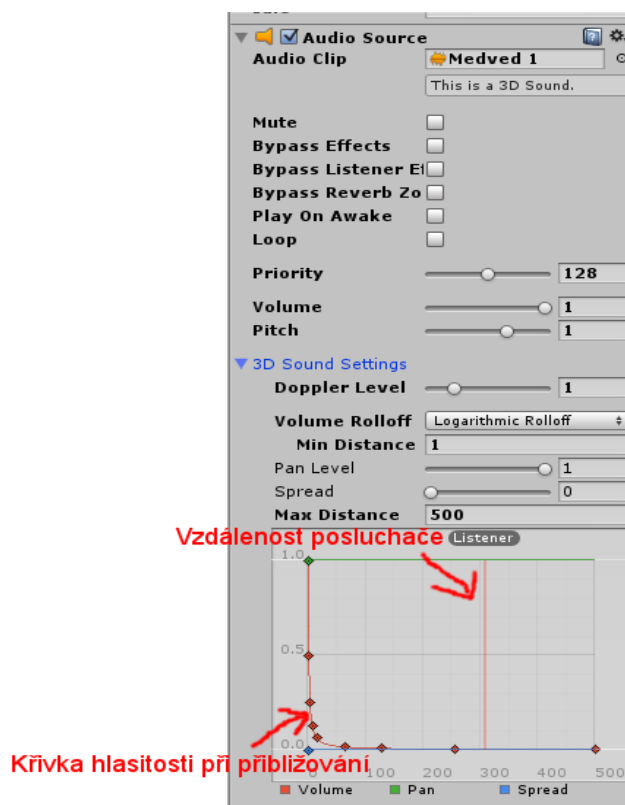
Obr. 23 Kostra s objektem

Tento medvěd slouží ve hře jako překážka cestě. Je usazen všude tam, kam hráč nemusí chodit, aby zbytečně nebloudil v částech města, ve kterých není žádná herní aktivita. Pro medvěda jsem vytvořil 2 druhy animace. První je klidový stav tzv. „idle“. Jedná se o cyklickou animaci, která určuje, jak se medvěd bude pohybovat ve stavu, kdy neprovádí žádnou akci. Druhá animace nastává, když se hráč přiblíží a snaží se kolem něho projít. Medvěd se postaví na zadní a pohrozí hráči, aby tudy nechodil. Tento proces zajišťují dva skripty, které jsem vytvořil. První se stará o plynulý přechod mezi těmito dvěma animacemi a spouštění cyklické animace. Druhý spouští animaci pohrožení pokud se hráč k medvědovi přiblíží. Tomu to skriptu se podrobněji věnuji v kapitole (Skripty pro ovládání hry).

4.8 Zvuky

Zvuk je důležitou součástí většiny her. Pomocí ambientních ruchů se dotváří atmosféra prostředí. V některých klíčových situacích může hudba přidávat na dramatičnosti děje, apod.

V Unity 3D je zvuk rozdělen na dva druhy a to 2D zvuk a 3D zvuk. Načítání a spouštění zvuků je zprostředkováváno komponentami „Audio source“ a „Audio listener“. „Audio source“ je komponenta, která je přiřazená objektu, ze kterého by měl zvuk vycházet. „Audio listener“ je komponenta, která je přiřazená objektu, který by měl zvuky poslouchat. Pokud se tedy jedná 3D zvuk, tak komponenta „Audio source“ zjišťuje v jaké vzdálenosti vůči ní je objekt s komponentou „Audio listener“ a jakou rychlostí se přibližuje či vzdaluje. Podle toho přehrává výsledný zvuk. Pokud je tedy posluchač vzdálen například 20 metrů od zdroje zvuku, bude se zvuk přehrávat méně hlasitě než z 5 - ti metrů. Dále se zde také vypočítává rychlost, s jakou se posluchač přibližuje nebo oddaluje od zdroje. Podle toho se také mění vlastnosti zvuku. Komponenta posluchače pak také zjišťuje své natočení vůči zdroji zvuku a podle toho přehrává zvuky spíše v pravém, levém nebo obou reproduktorech. Zvuk typu 2D je pak ve všech situacích stejný. Používá se často na přehrávání hudby. Viz obr. 24.



Obr. 24 Komponenta Audio Source

4.9 Programování skriptů

Pro interakci hráče s prostředím bylo zapotřebí naprogramovat vhodné skripty. Unity 3D umožňuje programování v jazycích C# a JavaScript. Zvolil jsem si C# z důvodů větších zkušeností s tímto jazykem.

Skript v Unity 3D se na objekt přikládá jako další komponenta v inspektoru objektu. Zde je pak možné ručně nastavovat hodnoty veřejných proměnných, nebo sledovat, jakých nabývají hodnot během běhu hry. Unity 3D také podporuje ladění za běhu programu. To znamená, že je možné měnit kód skriptu a po jeho uložení se změny ihned projeví v běžící hře. Využíval jsem této funkce hlavně pro formátování uživatelského rozhraní neboli „GUI“. Některé jeho prvky jsem tvořil skriptem, proto jsou vidět až po spuštění hry a jejich formátování by bez ladění za běhu programu bylo obtížné.

Pro tvorbu skriptů jsem zvolil program MonoDevelop, který se instaluje spolu s Unity 3D. Tento program obsahuje prvky spjaté s jazykem C#, jako je našeptávač, který je stejný jako Visual Studiu. Program MonoDevelop podporuje také ostatní jazyky, jako je JavaScript, Boo, apod.

Základní struktura skriptu v C# v Unity 3D obsahuje dvě metody a je odvozená od třídy MonoBehaviour. První metoda v šabloně pro C# skript je metoda **Start**. Kód, který je uvnitř této metody, se provede ihned po inicializaci skriptu. Druhou metodou je **Update**. Vše co se nachází v této metodě, se provádí na každý snímek hry. Třída MonoBehaviour obsahuje všechny důležité metody, proměnné a zprávy (události). Šablona nově vytvořeného skriptu pro C# vypadá následovně:

```
using UnityEngine;
using System.Collections;

public class Prvni_Skript : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

}
```

Zdrojový kód 1: Šablona pro C#

U tvoření skriptů bylo nutno přihlížet k jejich náročnosti na hardware a znuvupoužitelnost. Je zde spousta případů, kdy je vyžadováno, aby skript prováděl několik řádků kódu každý snímek. Musel jsem si dávat pozor, abych v těchto metodách neinkrementoval zbytečně nějaké proměnné, vypínal skript po

dokončení práce nebo alespoň vyřadil většinu skriptu výjimkou. Znovupoužitelnost vyžadovala napsat skript dostatečně abstraktně tak, aby se nechal použít ve více případech. Bylo třeba naprogramovat nástroj na tvorbu určitých mechanismů, bez nutnosti dalšího programování.

4.9.1 Skripty pro odladění výkonu hry

Z důvodů objemnosti modelu, ve které se hra odehrává, bylo zapotřebí naprogramovat skripty pro odlehčení náročnosti na hardware. Zvolil jsem vypínání vykreslování objektu, na který se hráč v danou chvíli nedívá. Skript jsem vytvořil tak, že na každý objekt, který se má vypnout, pokud se na něho hráč nedívá, se tento skript musí přidat jako komponenta. Tyto instance skriptu čekají, až se s jejich colliderem protne collider hráčovo zorného pole a v tu chvíli se zviditelní. Poté co collider hráčovo zorného pole opustí mnou kontrolovaný objekt, vypne se jeho vykreslování.

4.9.2 Skripty pro dávkovou úpravu objektu

Model města se skládal z mnoha objektů. Upravovat jeden po druhém ručně by zabralo mnoho času. Proto bylo zapotřebí naprogramovat skripty na hromadné úpravy modelu.

Hromadně bylo zapotřebí do modelu přidat můj skript na vypínání objektů, na které se zrovna hráč nedívá. Dále jsem upravoval nastavení shaderu u fasád. Import modelu totiž nepoznal, že textury fasád obsahují alfa vrstvu průhlednosti. Bylo tedy zapotřebí, aby skript našel všechny objekty, na kterých je tato textura nanesená a na shaderu zapnul průhlednost alfa vrstvy. Tento úkol mi velice ulehčilo jednotné pojmenování jednotlivých druhů objektů v celém modelu. Z názvů objektů v modelu bylo zřejmé, zda se jedná o objekt s texturou fasády nebo o jiné části budov. Proto jsem mohl pomocí skriptu tyto objekty najít podle jejich jména a hromadně upravit. Nyní je již celý model upravený ale budovám chybí důležitá komponenta a to *collider*. Collider je komponenta, která zjišťuje doteky, nárazy a průniky s jinými collidery. Jedná se o neviditelný obal 3D objektu, který může mít stejný tvar jako objekt nebo jednodušší z důvodů nižší náročnosti na výkon. V základní nabídce v Unity 3D je collider ve tvaru krychle, koule, kapsle a mesh collider, který se přizpůsobuje tvaru objektu. Pokud by budovy neměli komponentu collider, hráč by mohl procházet skrz ně. Pro model města jsem používal collidery ve tvaru krychle a mesh collider. Pokud se jednalo o průčelí domu, na kterém je

nenanesená textura fasády, použil jsem collider ve tvaru krychle jelikož se jednalo vždy o plochý pravidelný objekt. U ostatních objektů jsem použil mesh collider.

4.9.3 Scripty pro ovládání hry

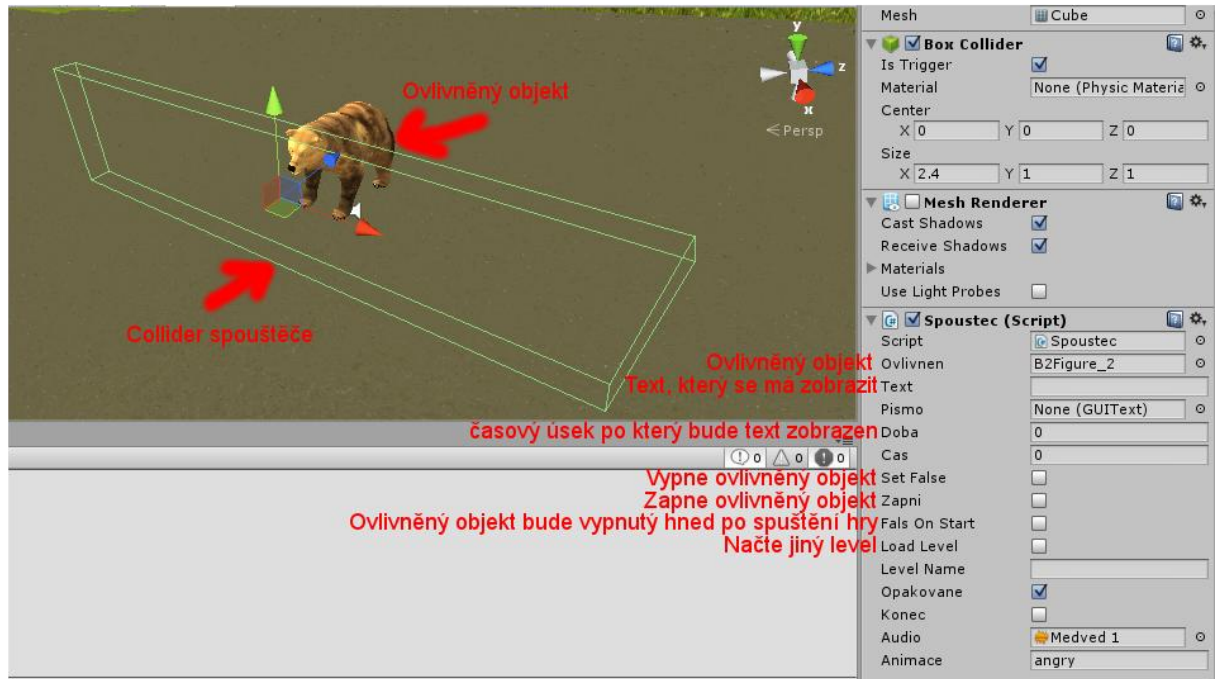
Styly pro ovládání hry využívám hned dva. Jeden pro manipulaci s minihrami, kdy je kamera ve fixní poloze a celá hra se ovládá pouze myší. Poté hráč může tento pohled opustit a volně se pohybovat prostředím. Pro druhý „volný pohyb“ jsem zvolil mnoho lety ověřený způsob ovládání FPS „*First person shooter*“. Jedná se o simulaci pohledu člověka z jeho očí. Chůzi zajišťují tlačítka „W,A,S,D“. Pohyb kamery (otáčení hlavy) a směr chůze udává myš. Tento skript Unity 3D již obsahuje spolu s originální komponentou „*Charakter controller*“, která je k tomuto ovládání hry určená.

Dále bylo nutné vytvořit způsob interakce hráče s okolím. Hráč v naší hře může zapínat hry, nechávat si vypsát informace o objektu, spouštět animace atd.

Zapnutí minihry provede hráč tak, že se k ní přiblíží, namíří na ní pohled a stiskne klávesu „E“. Tím aktivuje skript, který zajistí plynulý přesun kamery do pozice pro hraní hry. Tento skript pracuje ve dvou stavech. Buď má hráč svobodu odchodu z rozehrané hry, nebo musí hru dohrát, aby mohl odejít. Z tohoto důvodu si skript samotné minihry může zavolat metodu ze Skriptu pro ovládání kamery, aby jí vrátil do pozice pro volný pohyb.

Dalším důležitým mechanismem byl multifunkční Skript pro ovládání objektů po aktivaci hráčem. Tento skript opět vyžaduje, aby hráč mířil na ovládaný objekt a stiskl klávesu „E“. Tento skript je ovládaný parametry, které může desinger hry měnit v inspektoru. Zajišťuje základní funkce jako *vypnutí / zapnutí* objektu, fyzikálního chování, animace a zobrazení textu na obrazovce po určitou dobu.

V některých situacích je potřeba aby se animace, zvuky a texty spouštěly, pokud se hráč přesune do určité pozice. Například u postav, které postávají po herní mapě. Hráč se k nim může přiblížit a ony mu poradí, co má dělat. Nebo u animace zlodějky, která se snaží hráči utéci. Model zlodějky je usazen do herní mapy a čeká, dokud se hráč nepřiblíží na určité místo a až poté se spustí animace útěku. Tyto funkce zajišťuje skript, který jsem vytvořil. Nazval jsem ho „*Spouštěč*“. Jedná se o skript, který je zodpovědný za spouštění určitých akcí, jako je animace, zvuky, zobrazení textu apod. Oproti předešlému skriptu, se tento aktivuje automaticky a to pokud se collider hráče dotkne collideru objektu, na kterém je tento skript přidán jako komponenta. V inspektoru je poté možné nastavit, jaký objekt bude ovlivněn jakou animací a zvukem, jaký text se má hráči zobrazit na obrazovce a další, viz obr. 25.



Obr. 25 Spouštěč

```

using UnityEngine;
using System.Collections;

public class Spoustec : MonoBehaviour {

    // Use this for initialization
    bool hit; // pokud je true, hráč se spouštěče dotknul

    public GameObject ovlivnen; //objekt který má být spouštěčem ovlivněn
    public string text; //text který se uživateli zobrazí
    public GUIStyle Pismo; //styl písma
    public float doba; // doba zobrazení textu
    public float cas; //jak dlouho se bude text zobrazovat
    public bool SetFalse; // jestli se má objekt smazat
    public bool Zapni; // pokud je true objekt se zapne
    public bool FalsOnStart; // pokud je true bude objekt vypnut při
inicializaci skriptu
    public bool LoadLevel; // pokud je true načte se level
    public string LevelName; // jmeno levlu který se má načíst
    public bool opakovane = false; // pokud je true, bude se akce opakovat
pokaždé
    bool used = false; // zjišťuje se zdal byl spouštěč již použit
    public bool konec = false; // pokud je true jedná se o ukončovací spouštěč
    public AudioClip audio; // zvuk který se má přehrát

    public string animace; //animace která se má spustit na objektu

    void Start () {
    if(FalsOnStart == true)
        {
            ovlivnen.SetActive(false);
        }
    }
    // metoda pro zjištění doteku s hráčem
    void OnTriggerEnter(Collider other)
    {
        if(other.tag == "Player")

```

```

        {
            hit = true;
        }
    }
}
//metoda která zjišťuje jestli hráč opustil oblast spouštěče
void OnTriggerExit(Collider other)
{
    if(other.tag == "Player")
    {
        hit = false;
    }
}
//metoda zobrazující texty
void Texty()
{
    if(Pismo != null)
    {
        if(cas > 0 && konec == false)
        {
            cas = cas - Time.deltaTime;
        }

        if(cas < 1)
        {
            Pismo.text = "";
            konec = true;
        }
        else
        {
            Pismo.text = text;
        }
    }
}
//Update is called once per frame
void Update () {
    if(konec == false)
    {
        Texty();
    }
    //vše co se má stát po doteku hráče se spouštěčem
    if (hit == true && used == false)
    {
        konec = false;
        if(Zapni == true)
        {
            ovlivnen.SetActive(true);
        }
        if(Pismo != null)
        {
            cas = doba;
        }
        if(ovlivnen != null)
        {
            ovlivnen.SetActive(true);
        }
        if(ovlivnen.animation == null && animace == "")
        {
            //ovlivnen.animation.Play();
        }
        else if(ovlivnen.animation != null && animace != "")
        {
            ovlivnen.animation.Play(animace);
        }

        if(ovlivnen.audio != null)
        {

```

```

        if(ovlivnen.audio.isPlaying == false && ovlivnen.audio !=
null)
        {
            ovlivnen.audio.clip = audio;
            ovlivnen.audio.Play();
        }
        else
        {
            //ovlivnen.audio.Play();
        }
    }

    if(LoadLevel == true)
    {
        Application.LoadLevel(LevelName);
    }

    used = true;
}
//Debug.Log();
if(hit == false && opakovane == true)
{
    used = false;
}

if(ovlivnen != null && ovlivnen.animation.isPlaying == false &&
SetFalse == true && used == true)
{
    ovlivnen.SetActive(false);
}
}
}
}

```

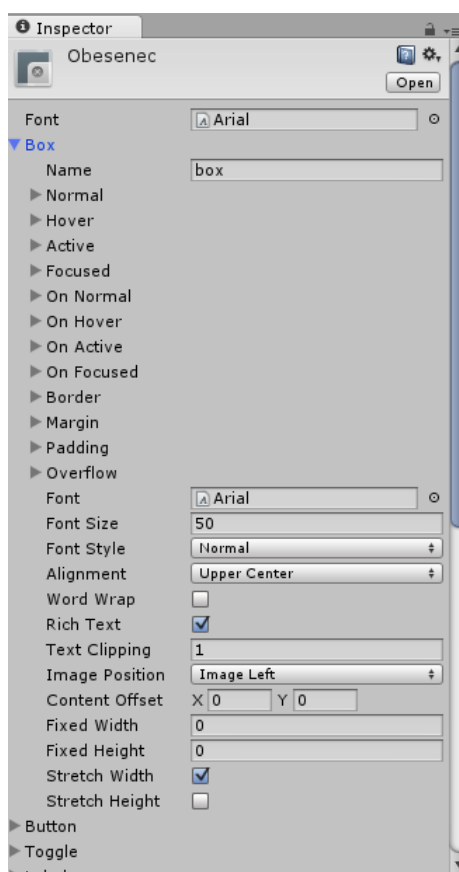
Zdrojový kód 2 : Spouštěč

Všechny tyto mechanismy by se neobešly bez skriptu, který zjišťuje, na co se hráč právě dívá. Za předpokladu že se hráč dívá na interaktivní objekt a stiskne klávesu „E“, musí tento skript ovládanému objektu poslat všechny důležité informace. Jedná se konkrétně o informaci, kdy se hráč na daný objekt dívá, a jelikož je hra připravena na sběr věcí do inventáře, posílá i obsah inventáře. Dále tento skript zajišťuje samotný sběr věcí do inventáře, jejich zobrazování a uchování.

4.9.4 Programování GUI (uživatelského rozhraní)

Uživatelské rozhraní se v Unity 3D může tvořit dvěma způsoby. První je přidání objektu do scény z menu „*GameObject/Create other*“. Zde je na výběr ze dvou druhů prvků uživatelského rozhraní. Prvním je „*GUIText*“. Tento objekt vykresluje na obrazovku text. V inspektoru lze nastavit obsah textu, barvu, polohu, font a velikost. Druhým prvkem je „*GUITexture*“. Tento objekt vykresluje na obrazovku texturu / obrázek. V inspektoru pak lze nastavit polohu obrázku na obrazovce, velikost apod. Druhým způsobem je tvorba GUI přímo ve skriptu. Slouží na to metoda „*OnGUI*“. Tato metoda je volaná každý snímek a lze v ní tvořit prvky GUI. Skriptem lze vytvořit více rozličných prvků jako je tlačítko (Button), rámeček

s textem (Box), text (GUIText), texturu (GUITexture), popisek (Label), zaškrťovací políčko (Checkbox), posuvník (Slider) apod. Formátovat jednotlivé objekty lze dvěma způsoby. První je zasahovat skriptem přímo vlastnost daného objektu a měnit tak jeho hodnoty, nebo použitím „Skinu“. Skin je objekt, ve kterém lze naformátovat jednotlivé objekty uživatelského rozhraní a poté ve skriptu těmto objektům skin přiřadit, viz obr. 26.



Obr. 26 Inspektor

```
void OnGUI ()
{
    GUI.skin = mujSkin; //proměnná mujSkin je inicializovaná z inspektoru
    //kam jsem vložil mnou upravený skin
    //přidání funkčního tlačítka ve kterém bude pouze písmeno „A“
    if(GUI.Button(new Rect(Screen.width/2-80,Screen.height-100,50,50), "A"))
    {
        Dopln("A");//Metoda kterou tlačítka vykoná
        this.gameObject.audio.PlayOneShot(klik);//Při kliku tlačítka se ozve zvuk
    }
}
```

Zdrojový kód 3: Formátování tlačítka a funkce

4.9.5 Tvorba jednotlivých mini her

Podle scénáře má být hra založena na řešení jednoduchých mini her. Tyto herní mechanismy brání hráči v postupu hrou a tvoří ve hře výzvu k pokoření. Minihry korespondují s historií i současností města Český Krumlov. Měly by hráči předat základní informace i Českém Krumlově povětšinou vizuální formou.

Základní vlastností většiny miniher je jejich použití jako samostatného prvku ve scéně. Jednoduché kopírování hry na různá místa ve scéně vyžaduje naprogramování hry tak, aby se mohla kdykoliv načíst do scény z assetů a hned fungovala bez propojování s dalšími skripty. Další vlastností hry je získání bodů za její výhru či za dílčí splnění úkolu. Tento úkol jsem vyřešil pomocí objektu, který obsahoval komponentu obhospodařující počet bodů, jejich přidávání a podobně. Skript mini hry si tento objekt sám musí najít, pokud na scéně existuje, a začít ho používat.

Hra skořápky

Jedná se o známou adaptaci hry skořápek. Hráči je ukázána odměna, která je schována pod skořápkou. Všechny skořápky se následně zamíchají a hráč musí určit, kde se odměna nachází.

Hra skořápky byla programovaná tak aby se později nechala samostatně hrát. Bylo nutné tuto hru založit na náhodě, aby se její hratelnost neopakovala. Dále v průběhu hry přibývají na scéně skořápky, které svým počtem zvyšují obtížnost hry. Ovládání hry je založené pouze na ovládání pomocí myši. Hráč myší klikne na skořápkou, pod kterou si myslí, že se odměna nachází. Hru spouští kamera, respektive pozice, na které se kamera nachází. Skript na míchání skořápek zjistí, kdy se kamera nastaví do polohy pro její hraní. První úkon, který vykoná je ukázka, kde se nachází odměna. Poté skořápky 3X zamíchá a čeká na tah hráče. Pokud se hráč netrefí, hra mu ukáže, kde se nacházela odměna a znovu se skořápky zamíchají. Pokud hráč uhodne, kde odměna je, přidá se mu bod a zvýší se počet skořápek, se kterými se bude skořápka s odměnou míchat. Zároveň se zvýší počet míchání. To znamená, kolikrát se jednotlivé skořápky mezi sebou prohodí. Takhle to pokračuje až do doby, která je nastavená v podmínce pro hru. V případě mého použití ve hře, je konec nastaven na 5 správných uhodnutí a poté se změní pozice kamery, což zastaví funkci celého skriptu Skořápek.

Skořápky jsou v naší hře jarmulky a pod nimi se jako odměna schovává vltavín. Minihra vizuálně, i textem předává hráči informace o prostředí. Po kliknutí na jakoukoliv jarmulku se na obrazovce vypíše popis jarmulky a její účel.

Hra oběšenec

Tato hra je založená na doplňování samohlásek do textu za určitý časový úsek. Hráči se zobrazí celé věta na 5 sekund. Poté se z ní smaží veškeré samohlásky a začne běžet časomíra. Pokud hráč nestihne větu doplnit do časového limitu, všechny již vyplněné políčka se odstraní a hra začíná od začátku. Pokud hráč stihne do časového limitu vyplnit celou větu, celý skript se zastaví a přidá hráči body.

Hra oběšenec by měla v hráči rozvíjet schopnosti postřehu a zároveň by měla povzbudit k rychlému doplnění jednotlivých písmen za účelu nasbírání co nejvíce bodů (vltavínů). Texty, které jsou ve hře použity, přímo souvisejí s prostředím hry, např: anglický nápis na zdi Českokrumlovské synagogy.

Hra házení na cíl

U této hry se hráč snaží vrhat objekty a trefovat terče. Terče jsou zde ve dvou stavech, špatný a správný terč. Hráč musí trefovat pouze správné terče a za ně získává body. Hra je podmíněná 5-ti zásahy za 40 sekund. Pokud se tak nestane, všechny již nasbírané body se hráči seberou a hra začíná od začátku.

Celá tato minihra funguje ve dvou obtížnostech. V první obtížnosti jsou před hráče postavené cvičné cíle, aby si vyzkoušel, jak se hra ovládá. Poté se generují terče náhodně po prostoru. Čekají 3 sekundy, zda li je hráč trefí a pak mizí. Na scéně se vždy vygeneruje maximálně 5 terčů naráz. V druhé obtížnosti je hra ztížena pohybem terčů do stran. Při generování terče se vždy určí jeho směr pohybu a tím se pak po dobu svého života pohybuje.

Ovládání hry je založeno na tahu myši a mělo by simulovat vrh objektu kupředu. Hráč stiskne levé tlačítko myši na objektu, který chce hodit, poté táhne myši směrem, kterým by chtěl objekt vrhnout. Po puštění levého tlačítka myši se vypočítá směr letu objektu a je vyvržen. Celá hra se opět ve scéně spouští pomocí pozice kamery.

Celý tento model hry byl aplikován na házení jarmulek na siluety dětských hlav. Hráč musí trefovat pouze chlapecké siluety, dívky jarmulky nenesí.

Hra vybarvování polí

Princip hry je založen na vybarvování určitých částí objektu podle předlohy. V našem případě se jedná o znak pětilisté růže (erb). Hráči se vygeneruje jeden z pěti znaků, který musí v časovém úseku 20 vteřin barevně zkopírovat. Generování předlohy je naprogramováno jako losování. Pokud hráči budou hru hrát vícekrát, je

založena na náhodném losování. To znamená, že pokaždé je posloupnost předloh jiná.

Minihra je založena na erbech jednotlivých pánu z jižních Čech [1]. Hráč poté dostává vzory jednotlivých erbů a podle nich musí vybarvit černobílou předlohu. Na obrazovce se generuje obrázek erbu a jeho název.



Obr. 27 Legenda o dělení růží

Spojování bodů

Tato hra je založená na spojování bodů do obrázku. Hráč kliká na jednotlivé body, které jsou očíslované. Musí dodržet správné pořadí bodů, aby se po spojení všech bodů vykreslil obrázek. Body znázorňující obrázek jsem vytvořil ručně, a to jejich umístěním do rohových míst tvaru, který má být vykreslen. Každému bodu poté bylo nutné přiřadit pořadí za pomoci skriptu. Poté bylo potřeba vytvořit další skript pro vykreslování čáry. Pro tento úkol jsem využil již zabudované komponenty LineRender. Tato komponenta v Unity 3D vykresluje čáry různých barev a tvarů. Dále také umožňuje zalomení čáry na určitých souřadnicích. Můj skript tedy ovládá komponentu LineRender a po kliknutí na správný bod v obrázku přiřadí komponentě souřadnice zalomení čáry v tomto bodě. Dále již jen kontroluji, zda byly propojeny všechny body z důvodů dokončení hry.

V minihře se hráč opět seznamuje s tvary pětilisté růže, která se spojením bodů vykreslí.

Posuvná skládačka

U této hry je cílem složit obrázek, který je rozdělen na čtvercové dílky. V obrázku je jeden dílek prázdný a ostatní dílky si s ním mohou vyměňovat pozici. Postupným prohazováním dílků hráč musí složit obrázek podle předlohy. Hra se při aktivaci sama zamíchá a poté zvukovým znamením dá hráči najevo, že může začít skládat. Celá hra je omezena dvou minutovým časovým limitem, za který hráč musí obrázek složit. Jinak se obrázek opět zamíchá.

V této hře jsem naprogramoval více možných variant použití. Je možné vypnout časový limit, hra se může aktivovat správnou pozicí kamery nebo může být již aktivovaná a zamíchaná, aby si jí hráč při průchodu kolem všiml. Možnost aktivace hry před příchodem hráče využívám u fasád na domech. Hráč při průchodu městem může zaznamenat špatně poskládanou fasádu na budově. Správným složením této fasády získává body.

Hráč skládá pouze obrázky související s Českým Krumlovem. Jedná se nejčastěji o historické pohlednice a staré fasády domů.

Hra stírání omítky

Jedná se o malování myši na 3D objekt. Hráč může ve městě zaznamenat rozdílnost fasád na některých budovách. Jedná se o historické nákresy fasád, nebo staré fotografie upravené jako textura pro budovu. Hráč poté může tahem myši tuto omítku setřít a odkrýt tak její současný stav.

Mnou upravený skript místo kreslení barvou na 3D objekt označuje pixely textury jako neviditelné. Poté s použitím shaderu pro zneviditelnění alpha vrstvy (Transparent/CutOut/Diffuse) se kreslení na fasádu jeví jako její odstraňování. Druhý mnou vytvořený skript kontroluje počet neviditelných pixelů v textuře. A pokud je splněna mnou zadaná podmínka, je hra splněna. Jako podmínku jsem určoval zhruba 80% pixelů z celé textury. Pokud tedy hráč smaže více jak 80% textury je hra splněna.

Tančení do rytmu hudby

Tato minihra je založená na mačkání tlačítka v určité době (rytmu). Z pravé části obrazovky přilétají textury jednotlivých tlačítek. Uprostřed obrazovky je malý kruh, do kterého se tato textura dostane přesně v době, kdy má být stisknuto určené tlačítko. Hráč tedy vidí řadu přilétávajících textur a tlačítko na klávesnici stiskne pouze tehdy, kdy se textura dostane na místo označené kruhem. Jelikož ze scénáře vyplývalo, že se musí tlačítka v kruhu zobrazovat do rytmu zpěvu, bylo nutné do

skriptu nastavit jednotlivé časování textur ručně. Použil jsem k tomu program Audacity ,kde jsem vyhledával nejvhodnější rytmus zpěvu. Tyto jednotlivé časy jsem zadal do Skriptu do pole typu float. Skript pak toto pole postupně prochází a zároveň měří čas skladby. Pokud se shoduje čas skladby s prvním časem v poli, vypustí první texturu a hlídá, kdy se čas písničky bude rovnat času v poli na druhém místě. Od jednotlivých časů v poli je ještě nutné odečíst čas, který textuře tlačítka zabere k dosažení středu obrazovky, kde se nachází kruh pro stisk tlačítka. Aby tento čas byl na různých rozměrech obrazovek stejný, bylo nutné, aby se textury tlačítek pohybovali relativně. Při použití metody standardní metody pro pohyb GUI textur jsem zjistil, že se textury přesouvají relativní rychlostí vůči velikosti obrazovky. Proto nebylo nutné dále do pohybu zasahovat.

Hráč netančí sám ale s dalšími postavami, se kterými dotváří kruh, ve kterém tancují. Postavy reagují na hráčovo mačkání tlačítek do rytmu. Pokud se hráč strefí do rytmu hudby, spustí se na postavách animace tlesknutí. Těmto postavám byla dále přiřazena klidová animace („idle“), aby působili více živě např.: otáčení hlavou, a mrkání očima.

Tato minihra je důležitá pro příběh hry. Hráče během tance okrade zlodějka o všechny vltavíny, které má u sebe. Postupné odebírání vltavínů je řešené skriptem, který čeká na určitou dobu skladby (určenou v sekundách). Poté postupně odebírá jeden vltavín za druhým, dokud hráči nezůstane pouze jeden vltavín. Po okradení zlodějkou se tanec zastaví a před hráčem se zjeví tajemná zlodějka. Pokud hráč taneční kolo opustí, zlodějka uteče nyní již otevřenými dveřmi.

Během tance se dále mění čas ve hře. Z noci se stane den. Tento jev jsem vyřešil pomocí shaderu „*Skybox blended*“, který sice není základní součástí Unity 3D ale je volně vystaven na internetu. [15] Tento shader se může ovládat přímo ze skriptu pomocí parametru, který určuje sílu prolnutí dvou textur oblohy. Pokud je parametr shaderu roven 0, je zobrazována pouze první textura oblohy a pokud je roven 100, je zobrazovaná pouze ta druhá.

4.10 Export hry pro různé platformy

Hra byla vytvářena pro dvě různé platformy a to pro internetový prohlížeč a pro mobilní zařízení s operačním systémem Android. Pro webový prohlížeč bylo nutné dbát ohledy na velikost hry z důvodů načítání. Před spuštěním hry v internetovém prohlížeči je nutné nejdříve stáhnout do počítače, na kterém se hra spouští, veškerý herní obsah. Hra byla rozdělena na dvě úrovně. První úroveň je Českokrumlovská synagoga, která po vyexportování do spustitelného souboru

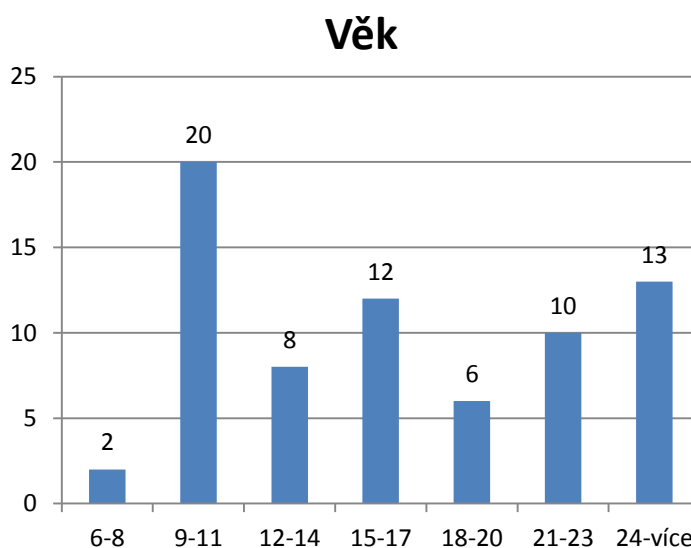
zabírala 50MB a druhou je historická část města Český Krumlov, která měla velikost 120MB. Pro efektivnější načítání herního obsahu do počítače se první stáhne a spustí úroveň se synagogou. Během hraní této úrovně se poté stahuje úroveň města Český Krumlov. Hráč mezi přechody z jedné úrovně do druhé, už nemusí čekat na stáhnutí dalšího obsahu.

U verze pro mobilní zařízení jsem musel úroveň města rozdělit na několik menších úrovní, jelikož hra zabírala velkou část operační paměti zařízení a neběžela plynule. Do jednotlivých ulic jsem umístil načítací místa. Pokud hráč dojde na konec nějaké úrovně, automaticky se načte ta, která s ní sousedí.

4.11 Testování hry

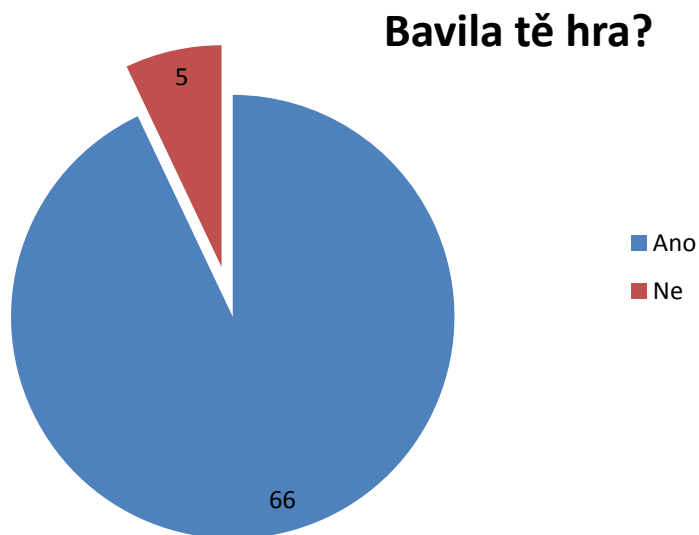
Testování hry probíhalo několika rozličnými způsoby. Ze začátku bylo nutné vyladit obtížnost her a ovládání pro děti. Testoval jsem hru osobně s dětmi a pozoroval, jak hru hrají. V některých hrách bylo nutné například vymazat časové omezení nebo přidat nápovědu pro ovládání hry. Poté co hra byla připravena pro spuštění na internetu, bylo nutné vyrobit jednoduchou webovou stránku (spirity.krajinak.cz). Na této stránce se nacházejí informace o hře a odkaz na hru samotnou. Dále se zde nachází odkaz na jednoduchý dotazník, který hráči mohou po odehrání hry vyplnit. Z dotazníku jsem zjišťoval, kolik času hráčům hra zabírala, jejich věk, jestli je hra naučila něčemu novému a jejich nápady co by chtěli do hry přidat. Do dotazníku přispělo 71 respondentů.

Hra byla vyvíjena pro cílovou skupinu 6 – 17 let. V tomto rozmezí se pohybovalo 60% respondentů, viz graf 1.



Graf 1 Věk

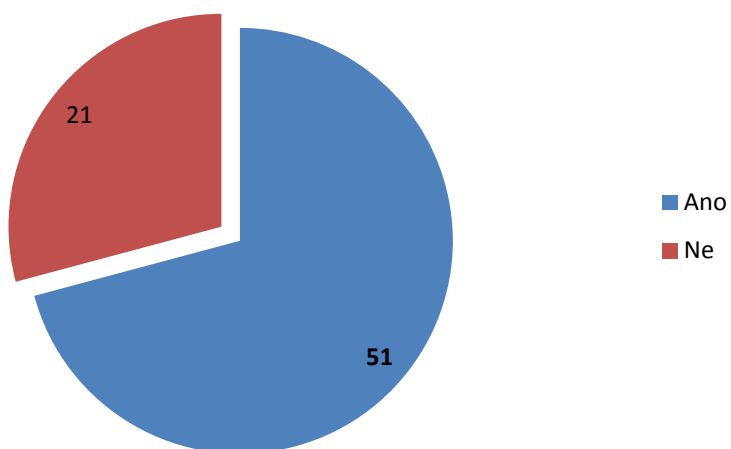
Na tomto grafu (graf 2) je znázorněné, zda hra hráče bavila. Z celkových 71 respondentů kladně odpovědělo 66 a záporně jen 5. Hra se tedy dá považovat za zábavnou a dobře hratelnou.



Graf 2 Bavila tě hra?

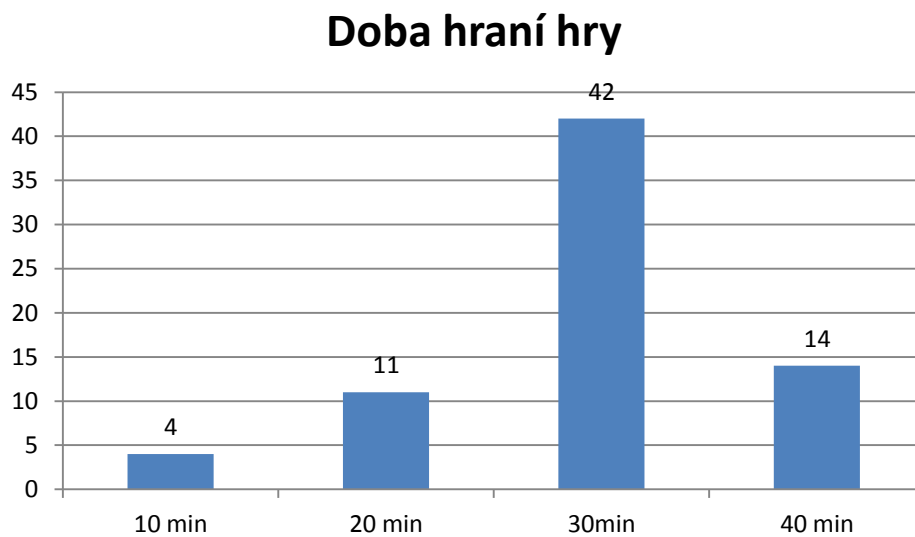
Na otázku, zda hráče hra naučila něco nového, odpovědělo 70% respondentů, že ano. Hra tedy nenásilnou formou předává informace o prostředí, ve kterém se odehrává. Viz graf 3.

Naučila tě hra něco nového?



Graf 3 Naučila tě hra něco nového?

Většina respondentů strávila hraním hry alespoň 30 minut, což odpovídá požadované době hratelnosti. Doba hratelnosti byla stanovena v průběhu tvorby této práce. Viz graf 4.



Graf 4 Doba hraní hry

5 Závěr

Aktuální verze hry obsahuje 2 prostředí a to prostředí Českokrumlovské synagogy a prostředí vnitřního města Český Krumlov. Odehrání celé této ukázky hry zabere hráčům zhruba 30 minut. Celou hrou je hráč doprovázen zlodějkou, kterou se hráč snaží dostihnout. Kladné postavy ve hře tvoří všudy přítomné děti, které hráči rádi, jak má dále postupovat, nebo kde jsou další minihry k odehrání. Hra je dále připravena na sbírání předmětů do inventáře, přidání dabingu jednotlivým postavám i s titulky.

Hra běží stabilně na středně výkonných počítačích v internetovém prohlížeči. U slabších počítačů jsem občas zaznamenal zhroucení prohlížeče při načítání úrovně města Český Krumlov. Nejčastěji to způsobilo vysoké vytížení paměti RAM procesem internetového prohlížeče, který byl následně ukončen operačním systémem. Co se týče zařízení pro Android, hra byla testována na tabletu „Lenovo yoga“ na kterém jsem naměřil 10 – 16 FPS. Unity 3D ve verzi zdarma neobsahuje žádné sofistikované nástroje na odladění výkonu hry. Z tohoto důvodu bylo velice obtížné dosáhnout alespoň takových to výsledků.

Práce na této bakalářské práci byla velice zajímavá. Porozuměl jsem více problematice tvorby her a získal plno zkušeností spjatých s touto problematikou.

6 Použitá literatura

- [1]. *Genre and game studies: Toward a critical approach to video game genres. Simulation & Gaming. 2006, roč. 37, č. 1, s. 6-23. DOI: 10.1177/1046878105282278. Dostupné z: <https://www.zotero.org/zachwhalen/items/F4SIEPDA>*
- [2]. *PEGI.info: Pan european game foundation. PEGI S.A. Fakta a čísla [online]. 2013. vyd. 2013 [cit. 2014-05-31]. Dostupné z: <http://www.pegi.info/cs/index/id/126/>*
- [3]. *BLOW, Jonathan. Queue: Game Development - Harder Than You Think. 1. vyd. New York: ACM New York, NY, USA, 2004. ISBN 1542-7730 EISSN.*
- [4]. *What is a Game Engine?. WARD, Jeff. Gamecareerguide.com [online]. 2008 [cit. 2014-06-24]. Dostupné z: http://www.gamecareerguide.com/features/529/what_is_a_game.php*
- [5]. *Unity documentation: Unity manual. UNITY TECHNOLOGIES. How do I Use Normal Maps? [online]. 1. vyd. 2014 [cit. 2014-05-31]. Dostupné z: <http://docs.unity3d.com/Documentation/Manual/HOWTO-bumpmap.html>*
- [6]. *Unreal engine. Epic games. UDK Licensing [online]. 2004-2014 [cit. 2014-05-31]. Dostupné z: <https://unity3d.com/unity/licenses>*
- [7]. *Unity 3D. Technologies. License Comparisons [online]. 2012. [cit. 2014-05-31]. Dostupné z: <https://unity3d.com/unity/licenses>*
- [8]. *Unreal engine. Epic Games. UDK LICENSING [online]. 2004-2014 [cit. 2014-05-31]. Dostupné z: <https://unity3d.com/unity/licenses>*
- [9]. *Epic Games, INC. Unreal Engine [online]. 2004-2014 [cit. 2014-05-31]. Dostupné z: <https://www.unrealengine.com/>*
- [10]. *Shiva Technologies SAS. Shiva 3D: 3D Game Engine with Development Tools [online]. 2012 [cit. 2014-05-31]. Dostupné z: <http://www.stonetrip.com/>*
- [11]. *Unity technologies. Unity 3D [online]. 2014 [cit. 2014-05-31]. Dostupné z: <http://unity3d.com/>*
- [12]. *Unity technologies. Unity 3D Comunity [online]. 2014 [cit. 2014-05-31]. Dostupné z: <http://forum.unity3d.com/forum.php>*
- [13]. *Cihla, Radek - Synagoga v Českém Krumlově. Cihla, Radek. In: Naše dny se naplnily. Z historie Židů v jižních Čechách /České Budějovice : Klub přátel Izraele, 2002 s. 188-196.*
- [14]. *Animace, osvětlení a výpočet v reálném čase 3D prostoru. Wikisofia [online]. 2005 [cit. 2014-06-01]. Dostupné z:*

http://http://wikisofia.cz/index.php/Animace,_osv%C4%9Btlen%C3%AD_a_v%C3%BDpo%C4%8Det_v_re%C3%A1ln%C3%A9m_%C4%8Dase_3D_prostoru

- [15]. BLOCH, Jiří. Vývoj fasád historických objektů ve městě Český Krumlov. [online]. [cit. 2014-06-21]. Dostupné z: http://www.encyklopedie.ckrumlov.cz/docs/cz/mesto_histor_vyvfas.xml
- [16]. Mobile game developer survey leans heavily toward iOS, Unity. Gamasutra.com [online]. 2012 [cit. 2014-06-24]. Dostupné z: http://www.gamasutra.com/view/news/169846/Mobile_game_developer_survey_leans_heavily_toward_iOS_Unity.php
- [17]. Dostál, J. Výukový software a didaktické počítačové hry - nástroje moderního vzdělávání. *Journal of Technology and Information Education*. 2009, Olomouc, Vydala Univerzita Palackého, Ročník 1, Číslo 1, s. 24–28. ISSN 1803-537X (print). ISSN 1803-6805 (on-line).

7 Seznam obrázků

- [1]. *Historie města Český Krumlov do roku 1622. Vízitceskýkrumlov.cz: On-line tourist guide [online]. [cit. 2014-06-24]. Dostupné z: <http://www.visitceskykrumlov.cz/cz/historie-mesta-cesky-krumlov-do-roku-1622/23/>*

8 Přílohy

CD: spustitelná verze hry pro internetový prohlížeč, instalační soubor typu „apk“ pro systém Android.