

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

Popularizační hra Fakulty elektrotechniky a informatiky

Miloslav Moravec

Bakalářská práce

2017

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2016/2017

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Miloslav Moravec**
Osobní číslo: **I13180**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Popularizační hra Fakulty elektrotechniky a informatiky**
Zadávací katedra: **Katedra informačních technologií**

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

Práce je ohraničena tématy herního průmyslu, tvorby aplikací pro stolní počítače a metodami distribuce hotových řešení prostřednictvím oficiálních i alternativních tržních kanálů.

V práci je třeba vymezit prostředí jednotlivých operačních systémů z pohledu podpory Unity3D aplikací, a popsat distribuční kanály pro jednotlivá prostředí, přičemž důraz je kladen na časové, legislativní a finanční aspekty. Dále bude v práci zapracován rozbor postupů při vývoji počítačových her.

Realizovanou případovou studií bude aplikace, která bude odpovídat současným požadavkům na moderní hru. V rámci implementace bude využito 2D nebo 3D scény, práce s modely či texturami a lineárním příběhem. Případová studie musí být vymezena prostřednictvím korektních dokumentů (např. storyline, game design document, konceptuální schémata).

Při své praktické práci student nejprve provede vymodelování budovy FEI (Nám. Čsl. Legií) a to minimálně v rozsahu přízemích prostor a prvních tří nadzemních podlaží. Takto vytvořený 3D model zpracuje vhodnými postupy proto, aby mohlo být provedeno mapování a texturování a vlastní budova mohla sloužit jako herní pole. V herním poli (budově) bude umožněn pohyb hráčům (pohled z 1. osoby), přičemž důraz bude kladen na seznámení se s prostorem a umístěním jednotlivých učeben a lokalizací vyučujících. V rámci popularizace FEI bude u vybraných vyučujících umožněno spustit minihry libovolného typu (např. didaktický test), které provedou hráče předmětem daného vyučujícího.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná**

Seznam odborné literatury:

DILLE, Flint. The ultimate guide to video game writing and design. New York: Watson-Guption Publications, 2007. ISBN 158065066X.

PECINOVSKÝ, Rudolf. OOP: Naučte se myslet a programovat objektově. Brno: Computer Press, a.s., 2010. ISBN 978-80-251-2126-9.

KNUTH, D. E.: Umění programování - Základní algoritmy, Brno, Computer Press 2008, ISBN: 978-80-251-2025-5.

WRÓBLEWSKI, Piotr. Algoritmy: datové struktury a programovací techniky. Vyd. 1. Překlad Marek Michalek, Bogdan Kiszka. Brno: Computer Press, 2004, 351 s. ISBN 80-251-0343-9.

KEOGH, Jim; DAVIDSON, Ken. Datové struktury bez předchozích znalostí : průvodce pro samouky. Vyd 1. Brno : Computer Press, 2006. 223 s. ISBN 80-251-0689-6.

Vedoucí bakalářské práce:

Ing. Josef Brožek

Katedra informačních technologií

Datum zadání bakalářské práce:

31. října 2016

Termín odevzdání bakalářské práce:

12. května 2017

Ing. Zdeněk Němec, Ph.D.
děkan



L.S.

Ing. Zdeněk Šilar, Ph.D.
pověřený vedením katedry

V Pardubicích dne 31. března 2017


Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 21. 04. 2017



Miloslav Moravec

Poděkování

Děkuji Ing. Josefu Brožkovi za skvělé vedení mé bakalářské práce. Dále bych rád poděkoval Bc. Lence Tobiškové za nesmírnou ochotu při sběru dat. Stejně tak děkuji Tomáši Heřmánkovi za pomoc a zpracování při sběru dat. Hlavně bych ale rád poděkoval své rodině, která mě neustále podporovala a snažila se mi ve všem pomoci. V neposlední řadě děkuji své přítelkyni za to, že to se mnou vydržela a byla chápavá v mé práci.

Anotace

Práce popisuje možnost využití herního enginu Unity při tvorbě popularizační hry Fakulty elektrotechniky a informatiky. V teoretické části práce je popsán obecný postup při tvorbě počítačové hry a možnosti dalšího využití herního enginu Unity. Práce se též zabývá tvorbou 3D modelu v CAD softwaru SketchUp a dále implementací tohoto modelu do herního enginu Unity.

Klíčová slova

SketchUp, Unity, 3D model, Fakulta elektrotechniky a informatiky, počítačová hra

Title

Faculty of Electrotechnics and Informatics PR game

Annotation

The thesis describes the option of using the Unity game engine for creating a popularizing game of Faculty of Electrical Engineering and Informatics. The theoretical part of this work is focused on description of the universal process of making a computer game and other options of using the Unity game engine. The thesis also deals with making of 3D model in CAD software SketchUp and furthermore with the implementation of this model to the Unity game engine.

Keywords

SketchUp, Unity, 3D model, Faculty of Electrical Engineering and Informatics, computer game

Obsah

Seznam obrázků	10
Seznam tabulek	11
Úvod	12
1 Vymezení a cíl práce	13
1.1 Základní pojmy.....	13
1.2 Postup vývoje práce.....	20
2 Postup při vývoji hry	21
2.1 Typy her.....	21
2.2 Platforma.....	23
2.3 Engine.....	23
2.4 Etapy vývoje hry.....	24
3 Nástroje	28
3.1 SketchUp.....	28
3.1.1 Prostředí softwaru SketchUp.....	28
3.1.2 Nástroje softwaru SketchUp.....	29
3.1.3 Export & Import dat z nástroje SketchUp.....	30
3.2 Unity.....	31
3.2.1 Prostředí softwaru Unity.....	32
3.2.2 Nástroje softwaru Unity Personal.....	33
3.3 Microsoft Visual Studio 2013.....	34
3.3.1 Prostředí softwaru Microsoft Visual Studio 2013.....	34
3.3.2 Funkcionality softwaru Microsoft Visual Studio 2013.....	35
4 Vlastní práce	37
4.1 Sběr dat.....	37
4.2 Tvorba 3D modelu fakulty.....	40
4.3 Vývoj samotné hry.....	42
4.3.1 Scénář.....	42

4.3.2 Programování.....	43
5 Uživatelské testování.....	49
Závěr	50
Použitá literatura	51

Obsah příloh

1	Příloha A	53
1.1	Model hlavního schodiště	53
1.2	Model druhého schodiště	53
1.3	Model bočního schodiště	54
1.4	Model schodiště s bezbariérovým přístupem.....	54
2	Příloha B	55
2.1	Splash Screen	55
2.2	Úvodní dialog	55
2.3	Prvotní zobrazení prostor školy ve hře	56
2.4	Dialog s informacemi o kantorovi a možnosti spuštění testu	56
2.5	Testový formulář.....	57
2.6	Zobrazení průběžných výsledků zkoušek	57
2.7	Zobrazení informace o výhře	58
2.8	Zobrazení informace o prohře.....	58
3	Příloha C	59
3.1	Statistiky o výkonu hry – bez osvětlení scény	59
3.2	Statistiky o výkonu hry – s osvětlenou scénou	59
4	Příloha D	60
4.1	Textura vstupu do fakulty	60
4.2	Textura oken na hlavním schodišti	60
4.3	Textura oken v respiriu	61
4.4	Textura okna na bočním schodišti	61
5	Příloha E	62
5.1	Přiložené DVD	62

Seznam obrázků

Obrázek 1: 3D model bez textury	14
Obrázek 2: 3D model s texturou	14
Obrázek 3: Porovnání originální nástěnné cedulky s bitmapovým obrázkem.....	17
Obrázek 4: Práce s texturou v softwaru Unity	17
Obrázek 5: Typický průběh práce game engineu	18
Obrázek 6: Princip animace	19
Obrázek 7: Prostředí nástroje SketchUp	29
Obrázek 8: Nastavení exportu modelu v programu SketchUp	31
Obrázek 9: Prostředí softwaru Unity Personal	32
Obrázek 10: Prostředí softwaru Microsoft Visual Studio 2013.....	35
Obrázek 11: Windows Forms Application	36
Obrázek 12: Půdorys přízemí Fakulty elektrotechniky a informatiky Univerzity Pardubice...38	
Obrázek 13: Ukázkový soubor s otázkami z předmětu IWWW.....	39
Obrázek 14: Ukázka XML souboru s předmětem IWWW.....	40
Obrázek 15: Ukázka použití obrázku jako šablony v softwaru SketchUp	41
Obrázek 16: Výsledný 3D model fakulty z programu SketchUp	42
Obrázek 17: FPS Controller v herním engineu Unity	44
Obrázek 18: Capsule Collider před vstupem do kabinetu v Unity	45
Obrázek 19: UML diagram třídy UserConfiguration	46
Obrázek 20: Příloha - Model hlavního schodiště.....	53
Obrázek 21: Příloha - Model druhého schodiště	53
Obrázek 22: Příloha - Model bočního schodiště.....	54
Obrázek 23: Příloha - Model schodiště s bezbariérovým přístupem	54
Obrázek 24: Příloha - Splash Screen	55
Obrázek 25: Příloha - Úvodní dialog.....	55
Obrázek 26: Příloha - Prostory školy ve hře.....	56
Obrázek 27: Příloha - Dialog vybraného kantora	56
Obrázek 28: Příloha - Testový formulář	57
Obrázek 29: Příloha - Zobrazení průběžných výsledků zkoušek	57
Obrázek 30: Příloha - Zobrazení informace o výhře	58
Obrázek 31: Příloha - Zobrazení informace o prohře	58
Obrázek 32: Příloha - Statistiky bez světla	59

Obrázek 33: Příloha - Statistiky se světly	59
Obrázek 34: Příloha - Textura vstupu do fakulty	60
Obrázek 35: Příloha - Textura oken na hlavním schodišti.....	60
Obrázek 36: Příloha - Textura oken v respiriu	61
Obrázek 37: Příloha - Textura okna na bočním schodišti.....	61

Seznam tabulek

Tabulka 1: Soupis předmětů v jednotlivých studijních oborech hry	43
--	----

Úvod

Tato práce pojednává o průběhu tvorby počítačové hry. Jedná se o hru pro platformu webového prohlížeče, která se odehrává v prostředí Fakulty elektrotechniky a informatiky Univerzity Pardubice.

Práci jsem si vymyslel sám po absolvování předmětu Vývoj počítačových her. Hlavní myšlenkou mé práce bylo vytvořit interaktivní prostředí pro budoucí uchazeče o studium na této fakultě. Díky této aplikaci si bude moci každý z pohodlí svého domova prohlédnout a samostatně projít prostory školy a zároveň se seznámit s některými předměty, které se na této fakultě vyučují.

Formální úprava práce byla zvolena na základě [1].

Práce je strukturovaná tak, že nejprve jsou objasněny základní pojmy, týkající se vývoje této počítačové hry (kapitola 1). Čtenář se například dozví co je to 3D model, CAD software či herní engine. Dále se práce věnuje vysvětlení obecného postupu při tvorbě počítačové hry (kapitola 2). V této části jsou vysvětleny pojmy: storyline, level design document a game design document. Následuje kapitola 3, věnující se popisu jednotlivých nástrojů, které byly použity při tvorbě praktické části bakalářské práce. Konkrétně jsou to nástroje SketchUp, Unity a Microsoft Visual Studio. Předposlední kapitola (kapitola 4) detailně popisuje průběh vlastní práce na vývoji aplikace, a to od sběru dat, potřebných pro tuto práci, až po vyhotovení finální aplikace. A konečně poslední kapitola (kapitola 6) informuje čtenáře o průběhu testování funkčnosti výsledné aplikace a ladění případných chyb a nedostatků.

Typografické konvence

V této práci se používají následující typografické konvence:

Kurzíva

Používá se pro vyznačení názvů vlastností game objectů.

Styl písma Courier New

Slouží k vyznačení názvů tříd, metod a datových struktur.

1 Vymezení a cíl práce

Cílem této práce je vytvořit funkční hru pro platformu webového prohlížeče v herním enginu Unity. Tato hra uživateli představí prostředí Fakulty elektrotechniky a informatiky Univerzity Pardubice a dále ho seznámí s některými předměty, které se na této fakultě vyučují.

Tato kapitola představuje základní průběh tvorby této práce a objasnění několika pojmů, které se v této práci objevují.

1.1 Základní pojmy

3D model

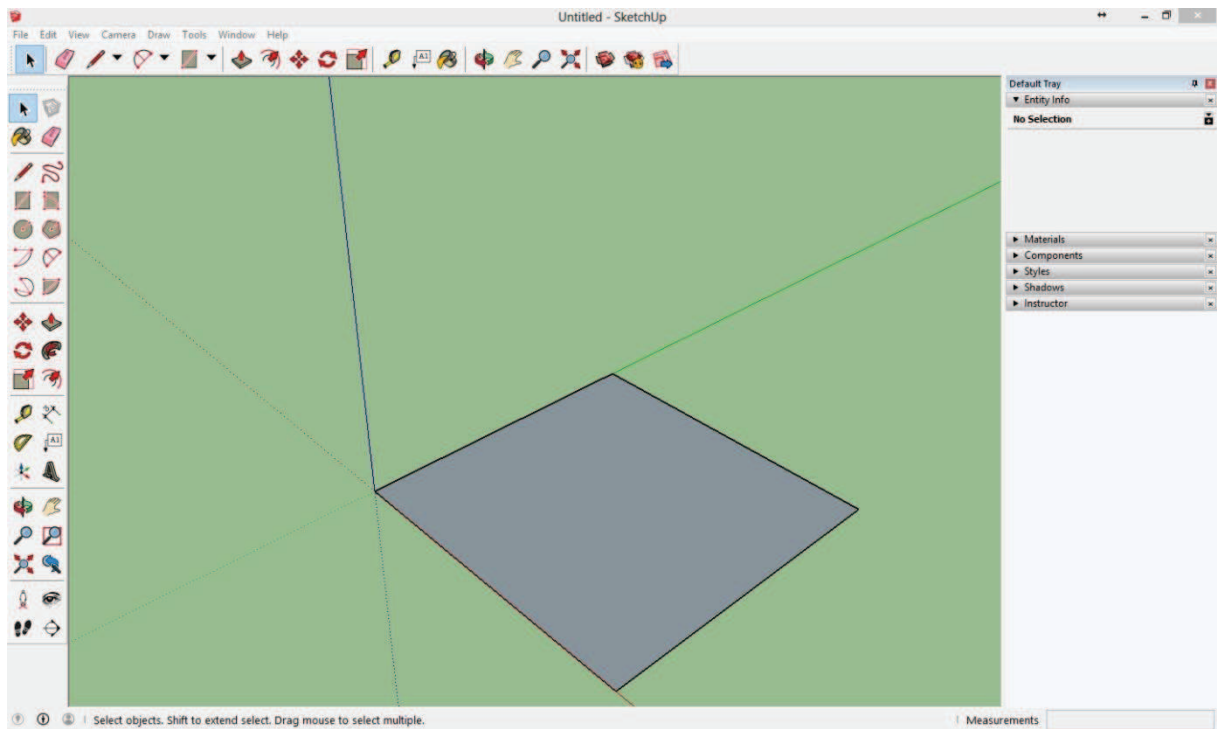
3D model je výsledek 3D modelování. 3D modelování je proces vytváření jakéhokoliv povrchu nebo objektu v trojrozměrné podobě pomocí manipulace s polygony, hranami a vrcholy v simulovaném 3D prostoru. 3D modelování může být dosaženo buď ručně, pomocí specializovaných 3D softwarů, které umožňují vytvářet a deformovat polygonální povrchy, anebo skenováním objektů reálného světa do množiny datových bodů, které mohou být použity k digitální reprezentaci objektu. [2]

3D modelování se používá v celé řadě oblastí, včetně strojírenství, designu, filmu, vizuálních efektů a v neposlední řadě při vývoji počítačových her. [2] Mezi nejlepší softwary pro modelování ve 3D patří například SketchUp, Blender, Maya či 3ds Max Design. [3]

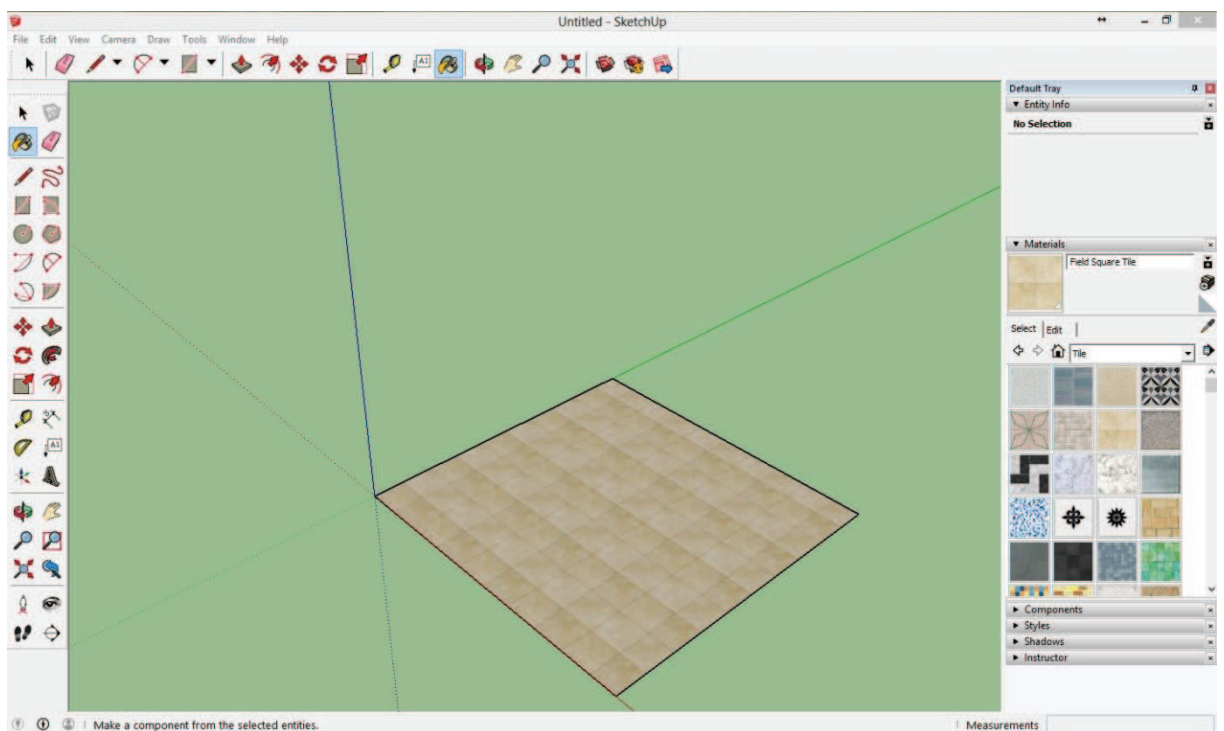
Hotový 3D model ovšem nemá sám o sobě žádný vzhled. Minimálně obsahuje pouze základní barvu povrchu. Finální vzhled modelu se musí dodělat pomocí materiálů, shaderů a textur. Tyto tři prvky mezi sebou úzce souvisí.

- **Materiály** jsou definice, jak by měl být daný povrch vykreslen. Obsahuje mimo jiné odkazy na použité textury nebo barevné odstíny. [4]
- **Shaders** jsou malé skripty, obsahující matematické výpočty a algoritmy pro výpočet barvy každého vykresleného pixelu. Jsou založené na vstupním osvětlení a konfiguraci materiálu. [4]
- **Textury** jsou rastrové obrázky. Protože materiál může obsahovat odkazy na textury, tak shader daného materiálu dokáže využít tyto textury při výpočtu barvy povrchu daného objektu. [4]

Následující obrázky zobrazují rozdíl vzhledu mezi objektem bez textury (Obrázek 1) a s texturou (Obrázek 2).



Obrázek 1: 3D model bez textury



Obrázek 2: 3D model s texturou

Souborové formáty finálních modelů mohou být různé. V mnoha případech si každý CAD software vyčleňuje svůj vlastní souborový formát. Proto se vymezilo několik univerzálních

formátů, které lze importovat do různých druhů CAD softwarů. Mezi takové formáty patří například:

- IGES,
- DXF,
- STEP,
- VDA-FS.

[5]

Importování 3D modelu do herního prostředí Unity se velmi liší v závislosti na modelovacím softwaru. Například program Maya neexportuje pouze 3D model, ale celou scénu, ve které se model vytvářel. Tato scéna pak obsahuje celý model, včetně textur a animací, které se při importu do Unity přepíše do animací, které nabízí Unity. Naproti tomu softwary SketchUp či AutoCAD Architecture mají podobné exportování modelu ve formě FBX (**F**ilm**B**ox) souboru, který se také dá velmi pohodlně importovat do herního enginu Unity.

Systemy CAD

Systemy CAD (Computer Aided Design – počítačem podporovaný návrh) jsou obvykle vektorové softwary, které se používají v různých oborech architektury a stavebního inženýrství při projektování. Umožňují rýsování technických, ale i krajinných prvků. Kreslicí prkno nahradilo elektronické a počítačová sestava se stala klíčovým prvkem projekčních ateliérů. Zdrojem pro tuto kapitolu byl [6]. Důvody zavádění této techniky jsou zejména:

- přesnost, variabilita a efektivita práce,
- množství a dostupnost digitálních dat,
- modifikovatelnost a výměna.

Rozhodujícím kritériem při výběru CAD systému je způsob použití:

- 2D: základním konstrukčním prvkem je lomená čára a jednoduché plošné geometrické tvary (kružnice, křivka, mnohoúhelník a další).
- 3D klasické modelování: základními prvky jsou těleso a plocha. Lze zde aplikovat booleanovské operace (sjednocení, průnik, rozdíl) a počítat objemy.
- 3D parametrické modelování: základní konstrukční technikou je definice parametrů, které popisují geometrické vazby a vztahy mezi 3D objekty. Změnou parametrů se generují nové sestavy i výkresová dokumentace.

Mezi nejznámější CAD systémy patří například AutoCAD, SketchUp nebo SolidWorks.

Grafický editor

Grafický editor je program, který je určený k tvorbě a úpravě grafiky neboli obrázků. Pokud není uvedeno jinak, zdrojem pro tuto kapitolu byl [7]. Existují dva základní druhy grafických editorů:

- bitmapové (rastrové),
- vektorové (objektové).

Bitmapová grafika je definována barvou jednotlivých bodů. Převede-li se fotografie do počítače (například pomocí skeneru), obrázek je vždy ve formě souboru v bitmapové grafice. Záznam v bitmapové grafice může být v různých formátech. Může to být PCX, GIF, TIF, BMP a jiné. Mezi bitmapové grafické editory patří například Adobe Photoshop či Corel PhotoPaint.

Grafická informace u vektorové grafiky je uložena ve formě matematického zápisu tvarů čar nebo křivek, které jsou základním elementem každého obrázku. S tímto typem grafiky pracují převážně profesionální CAD a kartografické systémy. Na rozdíl od bitmapové grafiky – zvětšování obrázku ve vektorové grafice nebere obrázku na kvalitě. Mezi vektorové grafické editory patří například Corel Draw.

Pro tuto práci byl vybrán bitmapový grafický editor Adobe Photoshop. V tomto editoru byly vytvářeny školní informační tabule a nástěnné cedulky, které lze nalézt u kanceláře každého kantora. Mezi přednosti tohoto editoru patří velký objem funkcí a jednoduché ovládání. Ukládaný obrázek může být uložen v mnoha různých formátech. Mohou to být formáty: Cineon, DICOM, IFF, JPEG, PNG, TIFF a mnoha dalších.

Následující obrázek (Obrázek 3) představuje porovnání mezi originální nástěnnou cedulkou a cedulkou vytvořenou v grafickém editoru Adobe Photoshop.



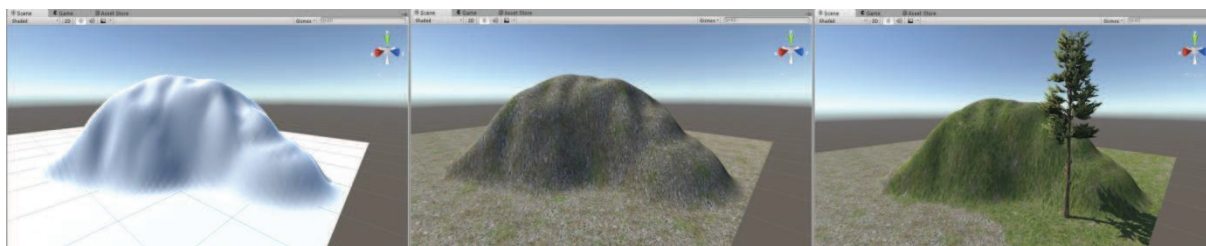
Obrázek 3: Porovnání originální nástěnné cedulky s bitmapovým obrázkem

Textura

Textura je obrazec, který se nanáší na povrch těles. Principem nanášení textur se označuje obarvení povrchu zobrazovacích těles různými obrázky. Geometrické vlastnosti těles se nijak nemění, pouze se jinak zobrazuje jejich povrch. Textury jsou ve většině případů představovány plošnými obrázky (rastrové obrázky vzniklé namalováním, vyfocení apod.), dále lze textury vytvářet pomocí různých algoritmů, založených převážně na fraktálních technikách – tímto způsobem vznikají takzvané procedurální textury. [8]

Procedurální textury lze použít buď pro výpočet rastrových obrázků před vlastním vykreslováním, nebo se může výpočet textur provádět až při vykreslování v reálném čase. Při vykreslování v reálném čase se parametry výpočtu textury nastaví podle aktuální velikosti a orientace plochy, na kterou má být textura nanášena. [8]

Na následujícím obrázku (Obrázek 4) lze vidět průběh práce s texturou v prostředí softwaru Unity. Obrázek vlevo představuje holý model kopce bez textury. Prostřední obrázek je tentýž kopec s již nanesenou texturou hlíny a poslední obrázek je kopec s procedurální texturou trávy, doplněný o otexturovaný model stromu.

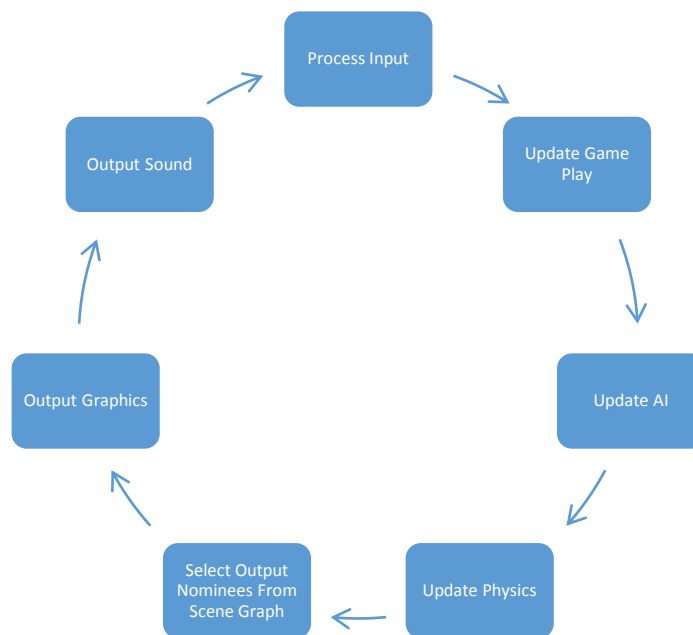


Obrázek 4: Práce s texturou v softwaru Unity

Herní engine

Herní engine (volně přeloženo jako herní motor) se dá přirovnat k motoru auta. Je to něco, co umožňuje hře pohyb. Obecná představa o herním engineu je poměrně jednoduchá: umožňuje vykonávat běžné úkoly, související s hrami, jako jsou rendering (tvorba reálného obrazu na základě počítačového modelu) či práce s fyzikou a vstupními prvky. Vývojáři (umělci, designéři, programátoři apod.) se tak můžou zaměřit na detaily, které dělají jejich hry jedinečné. [9]

Herní enginey nabízejí opakované použití komponentů, které mohou být upraveny tak, aby přivedly hru k životu. Zobrazení a animování modelů, detekce kolizí mezi objekty, práce s fyzikou či grafické uživatelské rozhraní mohou být všechno komponenty, které tvoří motor. Naproti tomu obsah hry, specifikace modelů a textur, myšlenka za kolizemi objektů a způsob reagování objektů s okolním světem jsou součástí, které dělají hru skutečnou. [9]



Obrázek 5: Typický průběh práce game engineu

Typický běh práce game engineu (Obrázek 5) startuje uživatel vstupním zásahem do hry. Samotná hra a logika AI (Artificial Intelligence – umělá inteligence) pak reaguje na vstup odpovídajícím způsobem a mění stavy v prostředí hry. Dále fyzický modul řeší fyzikální stavy a detekce kolizí. V dalším kroku Scene Graph (datová struktura objektů ve hře) vybírá vizuální objekty, které se mají uživateli zobrazit. Nakonec se aktualizuje grafické zobrazení a společně se zvukem tak vytvářejí nový stav hry. [10]

Asset

Pojem asset reprezentuje libovolnou položku, kterou lze použít v prostředí herního enginu při vývoji hry. Assetem může být třeba: [11]

- 3D model,
- zvuk,
- obrázek,
- skript,
- animace,
- textura,
- či rovnou celá herní scéna.

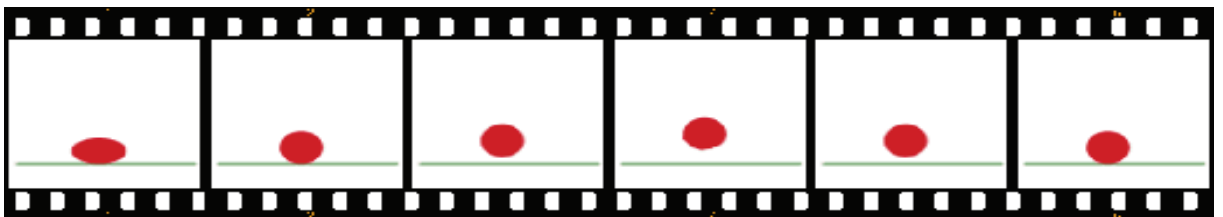
V prostředí herního enginu Unity existuje dokonce tzv. Asset Store, kde mohou vývojáři s těmito assety různě obchodovat.

XML soubory

XML soubor je datový soubor, napsaný ve značkovacím jazyce XML (eXtensible Markup Language). Důležité části dokumentu se označují pomocí značek. V terminologii XML se jednotlivým označeným částem dokumentu říká elementy. Elementy do sebe mohou být navzájem vnořené a tím dle potřeby zachycovat strukturu informací, uložených v dokumentu. [12]

Animace

Animace je způsob, jak zdánlivě rozpohybovat objekty. Většina programů vytváří animace pomocí klíčových snímků – nastaví se pozice objektu v daných časech a program sám dopočítá pohyb mezi těmito pozicemi. Na obrázku (Obrázek 6) se nachází filmový pás, který představuje animovaný film. Jednotlivé záběry zobrazují červenou kuličku v daném čase. Když se tento pás spustí, kulička bude zdánlivě skákat.



Obrázek 6: Princip animace¹

¹ Zdroj: [13].

Trigger

V překladu spoušť, která při vyvolání určité události vykoná danou sekvenci příkazů či operací. Pojem trigger se objevuje v prostředí databázových systémů, herním průmyslu a v mnoha dalších jiných programovacích jazycích. [14]

1.2 Postup vývoje práce

Postup vývoje této práce měl následující průběh:

- 1) Nastudování stavebních plánů fakulty a vyměření všech objektů, ze kterých se tato stavba skládá.
- 2) Nafocení veškerých prostor fakulty pro kompletní přehled o vybavení interiéru.
- 3) Vymodelování jednotlivých pater Fakulty elektrotechniky a informatiky Univerzity Pardubice v CAD softwaru, dle stavebních plánů a fotografií interiéru.
- 4) Vytvoření grafických informačních panelů a cedulek jednotlivých kantorů v grafickém editoru Adobe Photoshop.
- 5) Texturování modelu.
- 6) Kompletování jednotlivých modelů pater do jednoho celistvého modelu fakulty.
- 7) Vložení 3D modelu fakulty do herního engine jako nový asset.
- 8) Sběr testových otázek z jednoho předmětu od každého kantora na fakultě.
- 9) Vytvoření XML souboru pro každý vyučovaný předmět, který v sobě bude uchovávat data o předmětu a testové otázky.
- 10) Vložení vstupních bran před každý kabinet.
- 11) Zavedení animací na tyto vstupní brány.
- 12) Nastavení triggeru na tyto brány, při kterých se spustí test s otázkami z odpovídajícího předmětu.
- 13) Vytvoření dialogových oken ve formě formuláře s testovými otázkami.
- 14) Testování a ladění programu.

2 Postup při vývoji hry

Na začátku každé hry bývá myšlenka. Námět na vylepšení již existující hry, nebo vymyšlení hry zcela nové. Než se ale začne se samotným vývojem hry, je nejprve zapotřebí napsat koncept (game design dokument), ve kterém se vymezí příběh a prostředí hry. Tento dokument obsahuje naprosto všechny prvky hry. Od způsobu ovládání a vlastností postav až po části kódu, seznamu použitých assetů či odhadu náročnosti projektu na čas, finance a zdroje (lidské i materiální). Je velmi důležité, aby tento koncept existoval. Zabrání se tak zbytečným nesrovnalostem při vývoji hry, kdy skupina vývojářů má nějak rozdělenou práci a pokud každý bude pracovat s odlišnou myšlenkou, nikdy se jejich výsledky nesejdou.

Po vytvoření game design dokumentu následuje práce na level design dokumentu. V něm se navrhuje prostředí, ve kterém se hra bude odehrávat a dále dílčí úkoly, po jejichž splnění bude moci hráč postoupit ve hře dále. Jinými slovy se hra rozdělí do určitých úrovní (levelů). Každá úroveň musí být charakteristická svým prostředím a náročností. Právě kvalita jednotlivých úrovní často rozděluje hry na dobré a špatné, i když mají stejný žánr a podobné zpracování.

Existuje-li game design dokument a level design dokument, je pravý čas vhodně zvolit následující aspekty hry:

- typ hry,
- platforma,
- engine.

2.1 Typy her

Následující kapitola byla inspirována ze zdroje [15]. Současný dynamicky se rozvíjející trh počítačových her s sebou přinesl výrazné rozšíření herních žánrů:

- adventure,
- RPG,
- strategie,
- 3D akční,
- 2D akční,
- sportovní,
- simulátory,
- logické.

Adventure

Adventure (v překladu „dobrodružství“) je herní žánr, kde se hráč ocitá v roli bytosti, která putuje po určité lokalitě a po cestě plní různé úkoly, které mu nakonec pomůžou vyřešit hlavní poslání. Adventure může být akční, hororová i vtipná. Příkladem je česká hra Polda.

RPG – Role Playing Game

Role Playing Game (v překladu „hra v hlavní roli“ nebo „hra na hrdiny“) je typ hry, kde dějištěm bývá často smyšlený svět. Hráč si postaví družinu složenou z dobrodruhů různých ras, vlastností a povolání a vydává se na nebezpečnou pouť do tajemných krajin. Důraz je kladen na využití charakteristických vlastností jednotlivých postav v boji i při řešení logických problémů. Příkladem je Gothic či Witcher.

Strategie

Ve strategiích se hráč stává nejčastěji správcem nějakého území. Jeho úkolem je přidělené území zvětšovat na úkor protivníků. Zvětšování území se většinou dosahuje vojenskými taženími nebo obchodováním. Hrací pole je v podobě 3D mapy, na které se hráč dívá z ptáčích perspektivy. Toto pole se skládá z malých políček, na kterých se provádí veškeré operace. Strategie může být válečná i ekonomická. Příkladem je Age of Empires.

3D akční

3D akční hry jsou asi nejoblíbenějším herním žánrem. Hráč se pohybuje připraveným světem zcela svobodně a dívá se na něj z vlastního pohledu. Cestu hrdinovi znepřijemňuje množství nepřátel, které hráč musí porazit, což obvykle bývá jediným cílem hry. Příkladem může být Counter Strike.

2D akční

2D akční hra je starším kolegou 3D akční hry. Pohled na hru je buď shora, nebo ze strany. Příkladem je ovládání rakety, jež se řítí vesmírem a v cestě jí stojí řada nepřátelských raket, které má hráč za úkol zničit.

Sportovní

Sportovní hry jsou velmi oblíbeným žánrem na herních konzolích. Jedná se o simulaci různých sportů, zahrnující snad všechny známé disciplíny. Hra se vyznačuje zvládnutím pohybů sportovců, čehož se dosahuje pomocí technologie „motion capture“ (snímače

pohybů). Grafika bývá převážně ve 3D, počítána v reálném čase. Příkladem je Need For Speed či FIFA.

Simulátory

V herních simulátorech se dá simulovat takřka vše, co se hýbe. Nejběžnější bývají simulátory aut, letadel, vrtulníků, ale i vesmírných plavidel. Tyto hry se vyznačují technickou přesností a realistickým chováním strojů. Grafika je vzhledem k povaze her výhradně ve 3D a počítána v reálném čase. Příkladem je Microsoft Flight Simulator.

Logické

Logické hry dokážou velmi potrápit mozkové závity hráče. Zvládnutí hry závisí na rychlosti a přesnosti reakcí hráče, dále pak na schopnostech kombinace a logickém úsudku. Obvykle jsou pro jednoho či 2 hráče a reprezentují deskové hry, kde správná logika dovede hráče k vítězství. Příkladem může být Lemmings či Hexcells Infinite.

2.2 Platforma

Dalším důležitým aspektem před vývojem hry je vhodná volba platformy, na kterou bude hra vyvíjena a tomu je pak nutno uzpůsobit další věci při tvorbě hry (např. ovládání, zobrazení apod.).

Nejčastější platformy, na které se vyvíjí počítačové hry, jsou:

- Android,
- iOS,
- Mac OS,
- Play Station,
- PC (s důrazem na operační systém Windows),
- Xbox.

2.3 Engine

V dnešní době existuje obrovské množství herních enginů, ze kterých si mohou vývojáři vybírat a většina z nich je dokonce v základní verzi zadarmo. Výběr správného, vhodného enginu závisí na typu hry a na platformě, na které má být hra vytvořena.

Na velké plnohodnotné hry jsou nejčastěji využívány enginy:

- Unreal Engine,

- Cry Engine,
- či Unity.

Další známé enginy jsou například:

- Xcode – Pro jednoduché aplikace a hry na platformě iOS.
- Flash – Vhodný pro tvorbu her na webových prohlížečích i na mobilních zařízeních. V dnešní době se od tohoto enginu ale už ustupuje.
- Cocos2d – Engine pro vývoj malých 2D her.
- RPG Maker – Jednoduchý engine pro tvorbu vlastní RPG hry.

2.4 Etapy vývoje hry

Po výběru žánru hry, platformy a vhodného enginu přichází na řadu práce na samotné hře. Zdroj této kapitoly je [16].

Postup takové práce se dělí na etapy:

- programování hry,
- grafická stránka hry,
- zvuková stránka hry,
- testování hry,
- vydání hry.

Program

Programování her je stejné jako každé jiné. Avšak na rozdíl od kancelářských aplikací klade vyšší nároky na znalost hardwaru a operačního systému. Hry jsou totiž optimalizované více než většina jiných aplikací.

Grafika

Grafická stránka hry je mnohdy to první, čeho si hráč na hře všimne. Grafika ve hrách má mnoho podob. Dva nejzákladnější směry ovšem jsou:

- grafika pro renderované sekvence,
- grafika pro real-time 3D hry.

Renderované sekvence jsou zastoupeny takřka u všech žánrů her. Některé adventury se odehrávají v kompletně renderovaném prostředí a hráč během hry buď přechází mezi statickými obrazovkami, nebo je při přechodu z jedné scény do druhé spuštěna animace. Pro

předrenderované obrázky je charakteristická preciznost a dokonalost, která mnohdy působí až velmi nepřirozeným dojmem.

Grafika pro real-time 3D hry se převážně počítá v reálném čase, a proto jsou na ní kladena různá omezení. Například je to složitost a typ modelů, použitých v prostředí hry, což můžou být různé krajiny, budovy, dopravní prostředky aj. Další omezení jsou v počtu světelných zdrojů a použitých barev.

Zvuk

Zvuk je nedílnou součástí celkového dojmu hry a neměl by být v žádném směru podceňován. Ne vždy padá ovšem vina za nekvalitní zvuk jen na herní vývojáře. Ti jsou nuceni, stejně jako v dalších směrech hry, podvolit se hardwarovým možnostem platformy, na kterou hru vyvíjejí.

Na konzolích je situace z pohledu vývojáře jasná: k dispozici má celou řadu špičkových 16bitových zvukových procesorů, přičemž systém je optimalizován tak, aby šlo naráz mixovat několik desítek zvukových stop najednou. Například pro současné přehrávání hudby a zvukových efektů.

Na PC je nabídka hudebních formátů rozmanitější. Objevují se zde tři hlavní formáty: MIDI, MODuly a digitální záznamy. Současným trendem v hudebním doprovodu hry je bezpochyby digitální záznam (WAV, MP3, ...), ten je mnohdy tak kvalitní, že se vydává i jako samostatný soundtrack ke hře.

Kvalitní zvukové efekty dodávají na reálnosti každé hry. V dnešní době existuje mnoho specializovaných databází, obsahující i několik desítek tisíc zvuků, ze kterých mohou vývojáři čerpat. Samozřejmě některé databáze jsou placené a některé nikoliv. U těch neplacených musí ovšem vývojář počítat s tím, že zvuky nebudou tak kvalitní a přesvědčivé, jako ty placené. Někdy se vývojář nespokojí s nabízenými zvuky a sáhne po vlastních. Buďto si je ve speciálním hudebním softwaru sám vytvoří, nebo se přes mikrofon nahrají reálné zvuky, které se dále v počítači zpracovávají a upravují.

Testování

Po vyhotovení relativně funkčního úseku hry, zpravidla jedné scény nebo úrovně, se tato část předává dev-testerům. Ti mají za úkol hru opakovaně ozkoušet a zaznamenávat veškeré chyby, na které během hraní narazí. Tyto chyby pak nahlásí vývojovému týmu, který obratem chyby opraví a pošle testerům novou verzi. Poté, co se hra dostane do závěrečné etapy vývoje,

se poskytne alfa-testerům a později beta-testerům. Ti mají stejný úkol jako alfa-testeři, nicméně musí vyzkoušet hru celou, několikrát a na různých typech zařízení s různými konfiguracemi a hardwarovými prostředky. Při tomto testování je šance, že se odchytne většina chyb, ještě před vydáním finální verze.

Vydání hry

Po úspěšném otestování hry nastává vhodný okamžik k jejímu vydání. Vydání hry s sebou nese několik nelehkých úkolů: vytvořit a graficky upravit manuály a obal hry, zajistit vylišování DVD včetně fyzické výroby manuálu a obalu, případně vytvořit trailer ke hře, a nakonec se postarat o to, aby se hra dostala k co možná nejširšímu okruhu zákazníků. Pokud je vývojový tým finančně silný, dá se první část zvládnout ve vlastní režii. Distribuce produktu už je ovšem tvrdším oříškem.

V zásadě se nabízejí následující možnosti, jak hru distribuovat:

- online,
- fyzické médium,
- distribuční sítě e-licencí,
- Steam apod.

Licencí hry pak může být:

- shareware,
- OEM prodej,
- vlastní distribuce,
- profesionální distribuce.

Shareware distribucí nabízejí autoři zákazníkům možnost vyzkoušet program zcela zdarma. Mnoho velkých společností vypouští první úroveň své hry jako shareware a pokud zákazníka hra zaujme, může si objednat plnou verzi. Pro autory her je forma sharewaru finančně výhodná, protože nemusí platit nemalá procenta ze zisku distributorům.

Zkratka OEM (Original Equipment Manufacturer) označuje výrobce zařízení, jehož výrobek je prodáván a propagován jinou obchodní značkou. OEM prodej tedy znamená, že vývojový tým může uzavřít smlouvu s velkým výrobcem počítačů, ve které se výrobce zavazuje, že ke každému svému počítači bude dodávat i vyvíjenou hru. Příkladem může být i kromě her nabídka operačního systému k ceně počítače.

Vlastní distribuce je velice riskantní. Dovolit si ji mohou jen opravdu velké a úspěšné firmy. Rozhodně není vhodná pro malé začínající vývojové týmy a v dnešní době již prakticky neexistuje.

Profesionální distribuce je zpravidla nejlepší cesta pro většinu vývojových týmů. Distributor se dokáže postarat o to, aby se hra dostala i do těch nejmenších obchůdků a jako producent hry je i odpovědný za její propagaci a fyzický vznik. Pokud je hra opravdu vydařená, tak přes profesionálního distributora může i malý vývojový tým najít cestu, jak se nejlépe proslavit. Jakmile se tým proslaví, má mnohem větší šance k dalším projektům, sponzorům a distributorům. V celosvětovém měřítku je toto nejpoužívanější a nejúspěšnější způsob. Mezi nejznámější distributory her patří například:

- Activision,
- Blizzard,
- EA Games,
- Rockstar Games,
- Ubisoft a další.

3 Nástroje

Tato kapitola popisuje použité nástroje a vývojová prostředí při tvorbě praktické části této bakalářské práce.

K vytvoření 3D modelu fakulty byl použit nástroj SketchUp Pro 2016. Samotné sestavení hry probíhalo v programu Unity Personal a zdrojové kódy byly implementovány v prostředí Microsoft Visual Studia 2013.

3.1 SketchUp

SketchUp byl původně produktem firmy @Last Software, později Google Inc. a nyní ho vlastní společnost Trimble. Tento nástroj lze využít pro tvorbu a úpravu 3D modelů. Jeho uplatnění je možné nalézt v tvorbě počítačových her, filmů, v architektuře, modelování bytových prostor apod. [17]

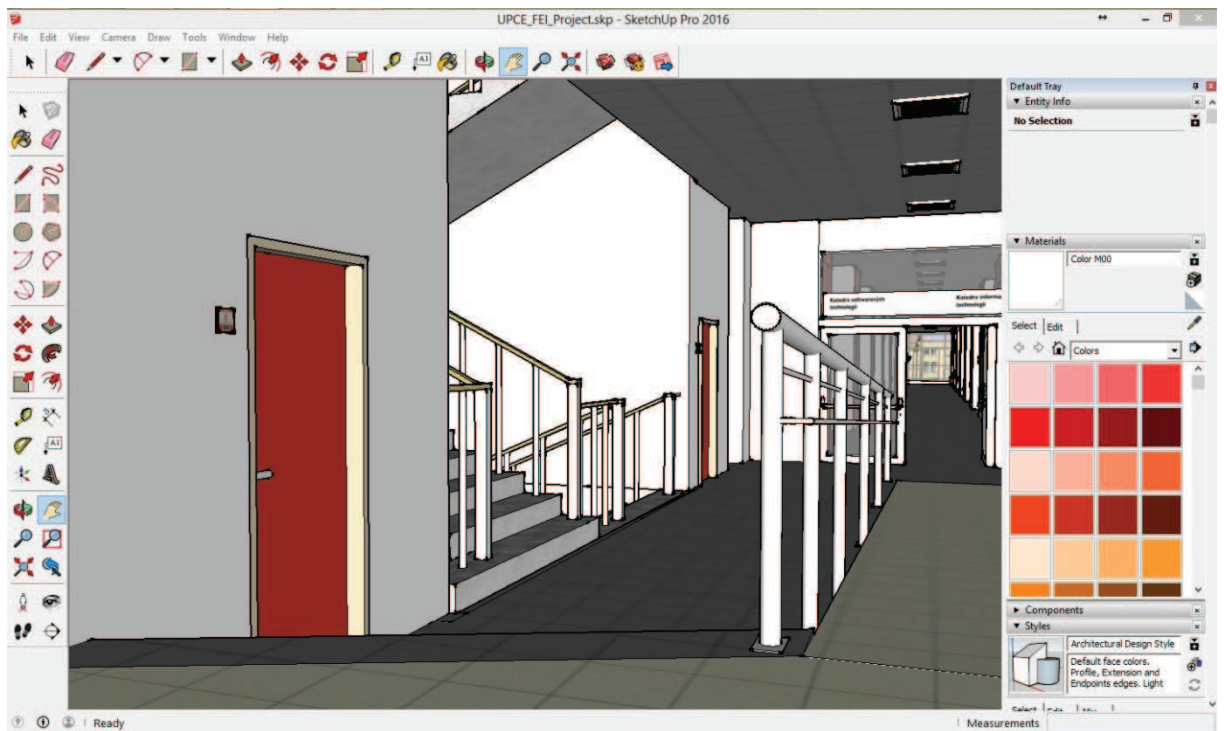
Tento software byl pro tuto práci vybrán proto, že se výborně hodí k tvorbě architektonických objektů a má velmi kvalitně zpracované uživatelské prostředí, ve kterém se nemá problém zorientovat ani úplný začátečník. Další výhodou je přehledná nástrojová lišta, ve které lze nalézt většinu používaných nástrojů. Po vybrání nástroje se zobrazí nápověda. Existují i klávesové zkratky některých základních funkcí, které výrazně urychlují práci. [17]

Pro tuto práci se nabízel i program Maya, který je v rámci studia vyučován. Avšak pro účely této práce je zbytečně moc složitý.

3.1.1 Prostředí softwaru SketchUp

Úvodní obrázek (Obrázek 7) zobrazuje prostředí nástroje SketchUp.

Horní část je Toolbar, zde nazvaný jako Getting Started. Lze v něm nalézt nejzákladnější funkce, které SketchUp nabízí. V levé části se nachází další Toolbar, označený jako Large Tool Set, ve kterém jsou pokročilé funkce programu, včetně těch základních. V pravé části prostředí je Toolbar (Default Tray), kde se nastavují materiály, komponenty, styly a jiné grafické prvky. Dolní lišta pak slouží ke sledování například délky čáry, která je právě vykreslována. Zbylá část prostoru je využita jako hlavní pracovní plocha, ve které se nachází osy x, y, z, podle kterých se může uživatel orientovat při kreslení modelu.



Obrázek 7: Prostředí nástroje SketchUp

3.1.2 Nástroje softwaru SketchUp

Během vytváření modelu fakulty byly použity téměř všechny funkce, které nástroj SketchUp nabízí. Zde je soupis těch nejzákladnějších.

Line

Funkce Line slouží k natažení úsečky z počátečního bodu do koncového. Vytvoří se tak vlastně obyčejná čára, se kterou se může následně pracovat dál.

Eraser

Funkce Eraser je v podstatě guma, kterou se dají mazat označené části modelovaného objektu.

Rectangle

Funkcí Rectangle se v prostředí SketchUp vykresluje obdélník. Patří mezi nezbytné nástroje při tvorbě 3D modelů, protože s pomocí nástroje Push/Pull z něj lze jednoduše vytvořit kvádr.

Circle

Funkce Circle je obdobná jako funkce Rectangle. Rozdíl je pouze v geometrickém tvaru vykresleného objektu, který má zde tvar kružnice.

Push/Pull

Funkcí Push/Pull lze jednoduše vytvořit z 2D modelu 3D model. Například plochu obdélníku lze vytáhnout do výšky a získat tak kvádr. Tažení ven z objektu a jeho prodloužení se nazývá Pull, naopak zatlačení plochy objektu do objektu samého se nazývá Push – při této operaci se objekt zkracuje. O kolik je potřeba objekt zkrátit, či prodloužit se dá přímo nastavit uvedením hodnoty na klávesnici a stisknutím klávesy Enter.

Scale

Funkce Scale dokáže daný objekt zvětšit či zmenšit přesně v daném poměru jeho rozměrů.

Offset

Funkcí Offset se dá vytvořit větší nebo menší verze původního objektu v ně daného objektu.

Tape Measure Tool

Tape Measure Tool je v podstatě metr, kterým se dají měřit vzdálenosti, ale i vytyčovat pomocné osy, které napomáhají při tvorbě složitějších objektů.

Paint Bucket

Paint Bucket vyplňuje části modelovaného objektu barvou či jiným materiálem, který se dá naimportovat, ale i vytvořit, a to přímo v prostředí nástroje SketchUp.

Follow Me

Funkce Follow Me patří mezi velmi pokročilé. Jedná se o natažení jednoho objektu na objekt druhý. Příkladem může být válec. Na konec obyčejné úsečky se pomocí funkce Circle umístí kružnice. Označí se úsečka, zvolí se funkce Follow Me a tahem myši se natáhne kružnice po celé délce úsečky. Tím vznikne vytoužený válec.

3.1.3 Export & Import dat z nástroje SketchUp

Import dat do prostředí softwaru SketchUp se provádí přes File/Import, kde se nachází i výběr formátu dat pro import.

Podporované formáty pro import jsou například:

- SketchUp Files (*.skp),
- AutoCAD Files (*.dwg, *.dxf),
- All Supported Image Types (*.bmp, *.jpg, *.png, *.psd, *.tif, *.tga).

Při výběru vstupních dat obrázkového formátu se musí dále zvolit, zda obrázek bude sloužit pouze jako obrázek nebo jako textura.

Před exportováním modelu je zapotřebí označit všechny části modelu, stisknout pravé tlačítko myši a zvolit možnost Make Component. Tím se všechny části modelu spojí v jeden velký celek.

Pro export pak stačí zvolit File/Export → 3D model a před uložením nastavit volby exportování dle Obrázku 8. Exportovaný model je ve formátu FBX.



Obrázek 8: Nastavení exportu modelu v programu SketchUp

3.2 Unity

Unity je výkonný grafický engine a plně vybavený editor, který poskytuje platformu pro vytváření 2D nebo 3D her a aplikací. V neposlední řadě podporuje i tvorbu her a aplikací pro nástroje virtuální reality.

Tento software byl vybrán proto, že je to jediný herní engine, který je v rámci studia vyučován. Zároveň poskytuje profesionální prostředí pro tvorbu plnohodnotných her.

Tento software vyrábí společnost Unity Technologies, která ho prodává v několika verzích, dle licence. Jsou to:

- Unity Personal (bezplatná verze pro začátečníky),
- Unity Plus (placená verze pro pokročilé),

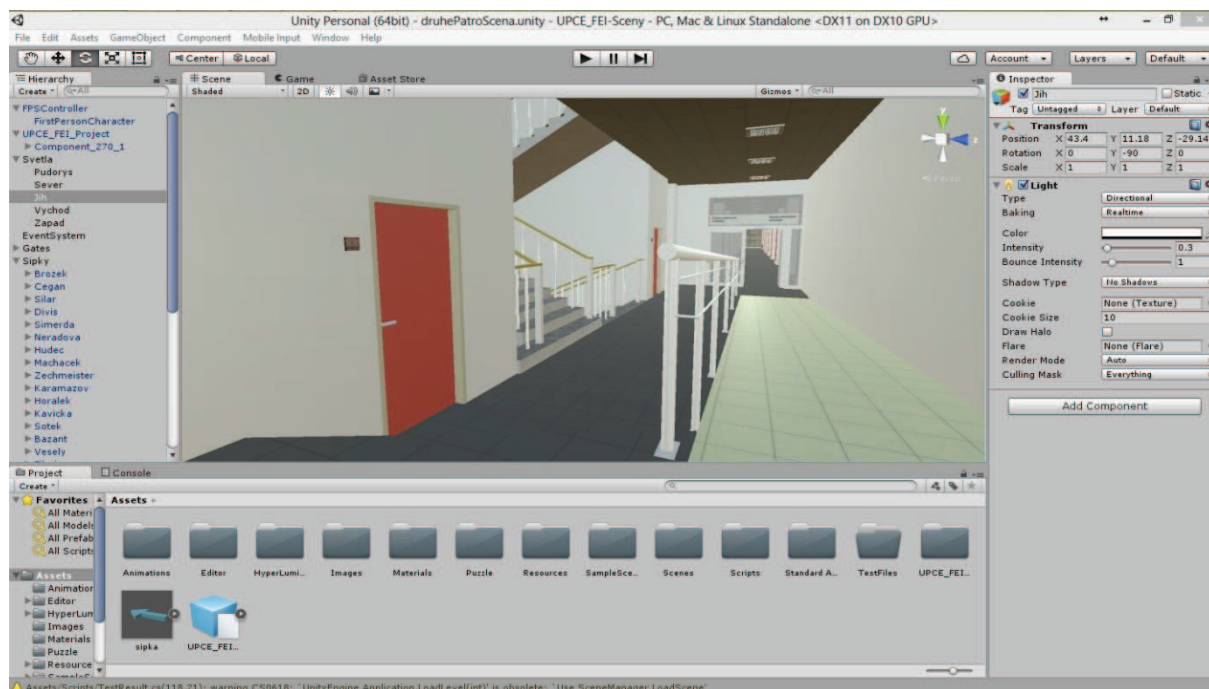
- Unity Pro (placená verze pro profesionály).

Nicméně platí, že pokud hra, vytvořená v bezplatném programu Unity Personal, začne vývojářům vydělávat větší peníze, musí se i tak společnosti Unity Technologies zaplatit jednorázový poplatek.

3.2.1 Prostředí softwaru Unity

Následující obrázek (Obrázek 9) zobrazuje výchozí prostředí softwaru Unity Personal.

Levá část okna se jmenuje Hierarchy a představuje seznam komponentů, ze kterých se skládá daná projekt. Po kliknutí na komponentu se v pravé části obrazovky zobrazí Inspector, ve kterém se upravují a nastavují vlastnosti zvolené komponenty. Dolní část okna je zobrazením složek a souborů, ze kterých se daný projekt skládá. Zde se například mohou importovat hotové komponenty či vytvářet skripty. V záhlaví okna se nachází několik tlačítek pro nastavení práce se zvolenou komponentou. Zda je potřeba ji otočit, rozšířit apod. Další tlačítka uprostřed záhlaví slouží ke spuštění, pozastavení či úplnému zastavení hry. Pracovní plocha uprostřed okna se skládá ze tří záložek: Scene, Game a Asset Store. Scene je prostor, ve kterém se hra vytváří. Nastavují se zde pozice, velikosti či animace jednotlivých komponentů. V záložce Game se nachází pohled, jak bude hra vypadat z pozice hráče. Vlastně je to pohled z kamery, která se jako komponenta vloží do Scene. Poslední záložka je obchod Asset Store.



Obrázek 9: Prostředí softwaru Unity Personal

3.2.2 Nástroje softwaru Unity Personal

Vytvořená komponenta v prostředí Unity se označuje jako GameObject. GameObject může nabývat mnoha stavů a charakteristik, které Unity nabízí již jako defaultní. GameObject se v Unity dělí do několika kategorií: 3D Object, 2D Object, Light, Audio, UI, Particle System a Camera. Samozřejmě lze vytvořit i prázdný GameObject, který si uživatel od základu může nadefinovat sám. Následující text zobrazuje seznam několika takových předdefinovaných objektů.

- 3D Object
 - Cube – vytvoří objekt ve tvaru krychle.
 - Sphere – vytvoří objekt ve tvaru koule.
 - Capsule – vytvoří objekt ve tvaru kapsle.
 - Cylinder – vytvoří objekt ve tvaru válce.
 - Plane – vytvoří plochu čtvercového tvaru.
 - Ragdoll – umožňuje vytvořit si vlastní postavu z jiných komponentů (částí těla).
 - Terrain – vytvoří plochu (stejně jako Plane), kterou je ale možné různě upravovat. Jako třeba pokrýt trávou, vymodelovat kopce atd.
 - Tree – vytvoří jednoduchý strom, který lze dále graficky upravit a například vložit do objektu Terrain.
 - Windzone – vytvoří objekt, který simuluje vítr. Dokáže pak například ohýbat stromy.
- 2D Object
 - Sprite – jedna z možností, jak ve hře zobrazit prostý obrázek. Využívá se při činnostech s obrázkem jako objektem. Například při stolních 2D hrách.
- Light²
 - Directional Light – využívá se ve hrách při simulaci denního světla (nelze poznat z jakého místa světlo přesně vychází).
 - Point Light – zde je poznat odkud světlo přichází. Využívá se například pro simulaci světla z žárovky.
 - Spotlight – obdoba Point Light. Lze zde nastavit dosah svitu. Využívá se pro svítílny.
 - Area Light – základem je obdélník v prostoru, který dokola svítí.

² Zdroj [18].

- Reflection Probe.
- Light Probe Group.
- Audio
 - Audio Source – vytvoří objekt pro přehrávání zvuku.
 - Audio Reverb Zone – objekt pro úpravu zvukové stopy.
- UI
 - Text – zobrazí na obrazovce text.
 - Image – další možnost, jak zobrazit obrázek. Tato možnost se nehodí pro práci s obrázkem.
 - Raw Image – obdoba Image. Dokáže ovšem zobrazit i obrázky ve formě textur, na rozdíl od objektu Image.
 - Button – zobrazí na obrazovce tlačítko.
 - Toggle – zobrazí na obrazovce přepínač.
 - Slider – zobrazí na obrazovce jezdce.
 - Scrollbar – zobrazí na obrazovce posuvník.
 - Dropdown – vytvoří rozbalovací menu.
 - Input Field – zobrazí řádek pro zadání textu.
 - Canvas – 2D oblast, do které se všechny objekty z kategorie UI umísťují. Samotný Canvas lze dále nastavit, zda se má například zobrazovat na celou obrazovku apod.
 - Panel.
 - Scroll View.

3.3 Microsoft Visual Studio 2013

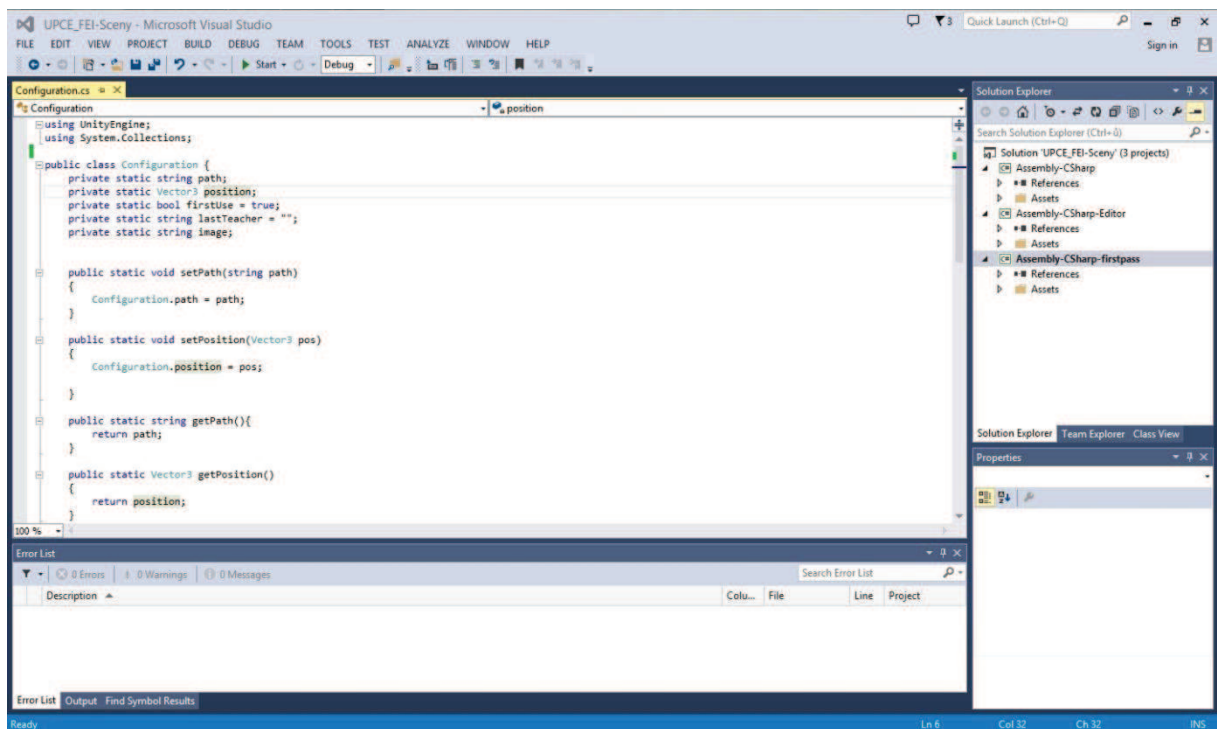
Microsoft Visual Studio 2013 je vývojové prostředí, umožňující vývoj aplikací pro osobní počítače v jazycích C#, Visual Basic, C a C++. Zároveň je tento program úzce spojen se softwarem Unity, jelikož skripty, které se zde píšou, mohou být ve formě JavaScriptů či právě v jazyce C#, a tak vzájemná integrita obou softwarů je zde zcela na místě.

Jak název napovídá – Visual Studio je počín společnosti Microsoft, která tento software vydává již od roku 1997. Aktuální verze má označení Microsoft Visual Studio 2017.

3.3.1 Prostředí softwaru Microsoft Visual Studio 2013

Jak lze vidět na následujícím obrázku (Obrázek 10), uživatelské prostředí Visual Studia je vcelku jednoduché.

V pravé části okna se nachází Solution Explorer. Ten vyobrazuje strukturu právě otevřeného projektu. Zejména jeho složky a soubory, ze kterých se skládá. Pod oknem Solution Explorer se nachází Properties. Zde se zobrazují informace o právě zvoleném objektu v kódu. Ve spodní liště lze přepínat mezi třemi záložkami. Jsou to záložky: Error List, Output a Find Symbol Results. Error List představuje seznam chyb v kódu. Jedna taková položka v seznamu uživatele informuje o tom, zda se jedná o závažnou chybu, či jen varování. Dále popíše, o jakou chybu se jedná, na kterém řádku se chyba nachází, v jakém souboru a samozřejmě i v jakém projektu. Po kompilaci projektu se v záložce Output zobrazí informace, zda kompilace proběhla úspěšně. V opačném případě se rázem zobrazí záložka Error List. Při hledání určitého výrazu v kódu se v záložce Find Symbol Results vypíše seznam míst, ve kterých se daný výraz nachází. Hlavní okno je samotný editor, kde se zobrazí text právě otevřeného souboru, připravený k editaci.



Obrázek 10: Prostředí softwaru Microsoft Visual Studio 2013

3.3.2 Funkcionality softwaru Microsoft Visual Studio 2013

Microsoft Visual Studio není pouze jen prostý editor pro psaní kódu. Umí toho daleko víc. Zde je soupis několika dovedností tohoto programu.

Editor kódu

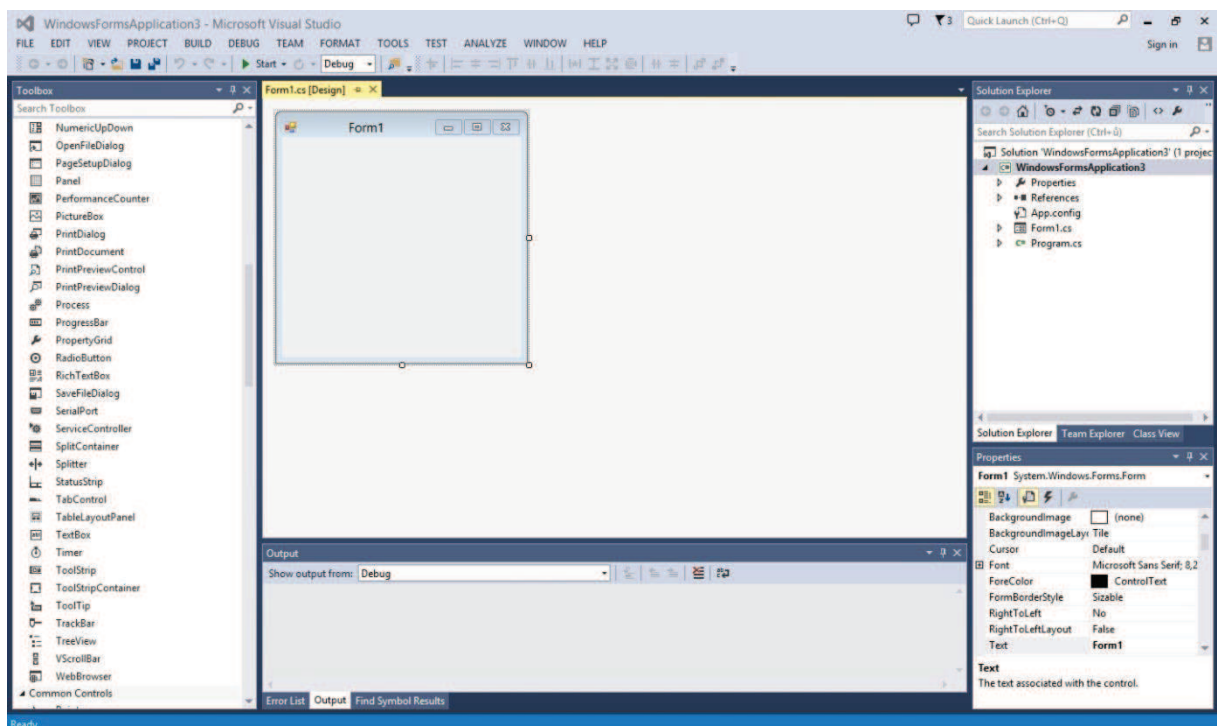
Jako každé vývojové prostředí i Visual Studio obsahuje svůj editor kódu. Ten dokáže kontrolovat a zbarvovat syntaxi, napovídat při výběru metod či dokončovat určité příkazy. Nastavení editoru lze libovolně měnit v Tools/Options/Text Editor.

Debugger

Debugger je výborná věc pro kontrolu funkčnosti napsaného kódu. Umožňuje rozmístění tzv. breakpointů na určitá místa v kódu, na kterých je potřeba podrobněji sledovat průběh dění. Běh programu se na breakpointu pozastaví. Od tohoto místa uživatel může sám krokovat program a pozorovat v dolní liště, jak se mění hodnoty jednotlivých proměnných v programu.

Designér

V prostředí Visual Studia lze vytvářet programy nejen jako konzolové aplikace (do příkazového řádku), ale hlavně jako aplikace s uživatelským rozhraním. Zde nazvané jako Windows Forms Application. Obdobné těmto aplikacím jsou pak WPF Aplikace, kde si lze mnohem více pohrát s designem aplikace. Dále lze vytvářet ASP.NET Webové aplikace, Silverlight aplikace a dokonce i aplikace pro Windows Phone. Obrázek 11 vyobrazuje prostředí Visual Studia během vývoje Windows Forms Application.



Obrázek 11: Windows Forms Application

4 Vlastní práce

V této kapitole bude detailně popsán vývoj aplikace, která je nedílnou součástí této bakalářské práce. Postup práce měl následující průběh:

- 1) Nastudování stavebních plánů fakulty a vyměření všech objektů, ze kterých se tato stavba skládá.
- 2) Nafování veškerých prostor fakulty pro kompletní přehled o vybavení interiéru.
- 3) Vymodelování jednotlivých pater Fakulty elektrotechniky a informatiky Univerzity Pardubice v CAD softwaru, dle stavebních plánů a fotografií interiéru.
- 4) Vytvoření grafických informačních panelů a cedulek jednotlivých kantorů v grafickém editoru Adobe Photoshop.
- 5) Texturování modelu.
- 6) Kompletování jednotlivých modelů pater do jednoho celistvého modelu fakulty.
- 7) Vložení 3D modelu fakulty do herního enginu jako nový asset.
- 8) Sběr testových otázek z jednoho předmětu od každého kantora na fakultě.
- 9) Vytvoření XML souboru pro každý vyučovaný předmět, který v sobě bude uchovávat data o předmětu a testové otázky.
- 10) Vložení vstupních bran před každý kabinet.
- 11) Zavedení animací na tyto vstupní brány.
- 12) Nastavení triggeru na tyto brány, při kterých se spustí test s otázkami z odpovídajícího předmětu.
- 13) Vytvoření dialogových oken ve formě formuláře s testovými otázkami.
- 14) Testování programu.
- 15) Ladění programu.

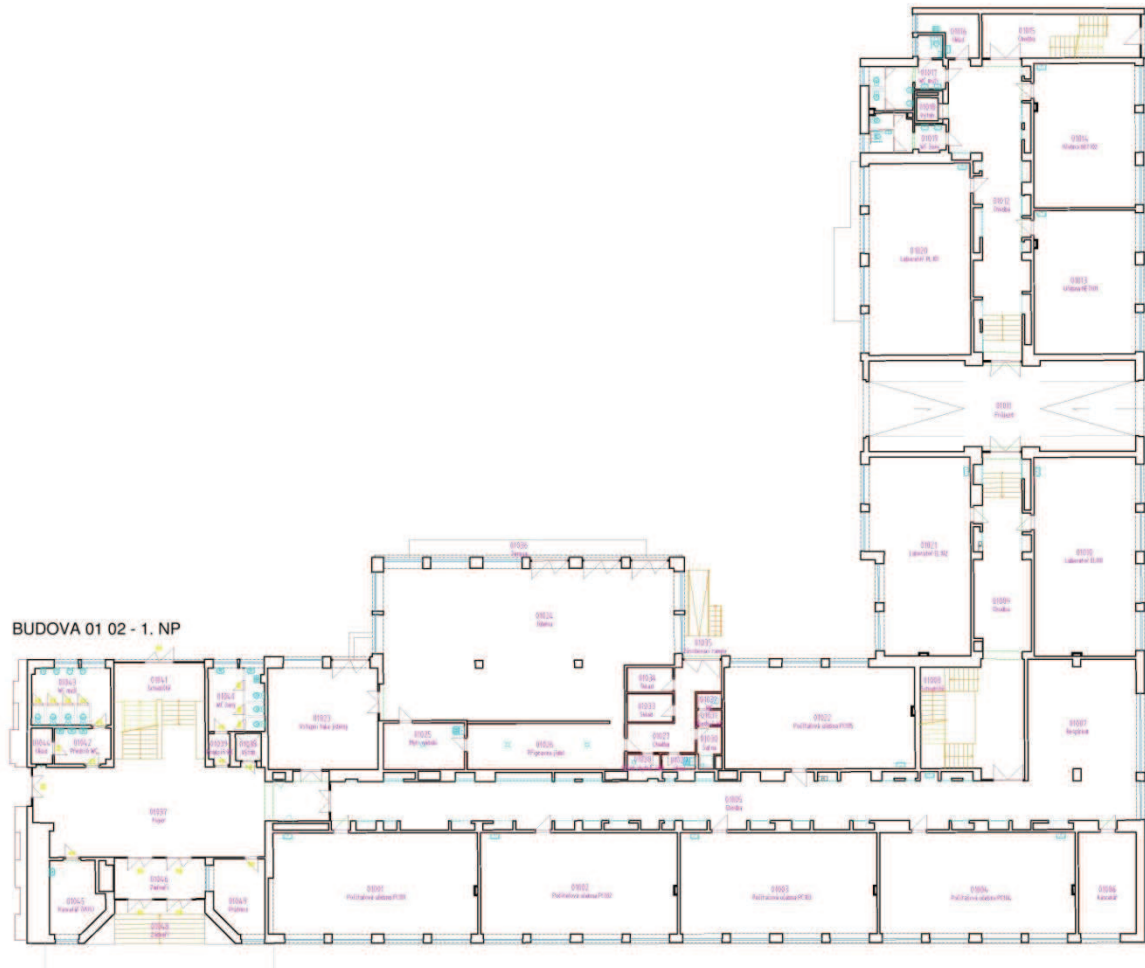
Z tohoto postupu si zaslouží detailní pozornost především následující témata, která se budou důkladněji zabývat daným problémem. Konkrétně jsou to:

- Sběr dat (zahrnuje body 1, 2, 8, 9),
- Tvorba 3D modelu fakulty (zahrnuje body 3, 4, 5, 6),
- Vývoj samotné hry (zahrnuje body 7, 10, 11, 12, 13).

4.1 Sběr dat

Prvotními informacemi, které bylo nutno získat, byly stavební plány fakulty a rozměry jednotlivých objektů ve škole. Stavební plány byly dodány sekretariátem Fakulty elektrotechniky a informatiky Univerzity Pardubice a poskytnuty ve formátech PDF a DWG

(AutoCad). Jak bylo zjištěno, půdorysy všech podlaží jsou, vyjma přízemí, stejné. Liší se pak pouze vybavením, nábytkem či umístěním dveří. Následující obrázek (Obrázek 12) zobrazuje půdorys přízemí fakulty.



Obrázek 12: Půdorys přízemí Fakulty elektrotechniky a informatiky Univerzity Pardubice

Po získání plánů bylo potřeba určit, v jakém měřítku bude modelovaný objekt vytvořen. Nejlepší a nejrychlejší cestou je samozřejmě poměr 1:1, proto byl tento poměr vybrán i pro tuto práci. Dále bylo třeba odměřit, kolika centimetrům ve skutečnosti odpovídá jeden centimetr na stavebním výkresu. Tím byl dokončen sběr dat pro hrubou stavbu školy. Následovalo přesné měření nábytku, schodišť a dalšího vybavení ve škole. Nejdůležitější veličinou byla výška objektů, protože ta ze stavebních plánů vyčíst nelze. Dále pak barvy interiéru či fotografie informačních cedulí a jiných tiskovin. To by byl sběr dat pro účely vytvoření 3D modelu fakulty. Ke správné funkčnosti a hlavní pointě hry následoval sběr dat, týkající se testových otázek z předmětů fakulty.

Pomocí emailové formy (a později i ústní) byli požádáni všichni kantoři na Fakultě elektrotechniky a informatiky Univerzity Pardubice k zaslání několika otázek k jednomu z předmětů, které učí v bakalářském studiu. Nutno dodat, že žádost se týkala pouze těch kantorů, kteří jsou zároveň garantem daného předmětu – to z důvodu zabránění duplicity předmětů. Všechny otázky z jednoho předmětu k sobě musely mít zároveň čtyři možné odpovědi, z nichž právě jedna byla správná. Tyto otázky byly dále zpracovány do XML souborů. Každý soubor byl vytvořen tak, aby odpovídal právě jednomu předmětu a aby uchovával název předmětu, otázky z předmětu a odpovědi k otázkám. Na Obrázku 13 lze vidět ukázkový test, tak jak byl zaslán k předmětu Návrh a tvorba WWW stránek. Obrázek 14 dále vyobrazuje jeho přenesení do XML souboru.

1. Co znamená akronym HTML?

- a) Home Tool Markup Language
- b) Hyperlinks and Text Markup Language
- c) Hyper Text Markup Language
- d) Hyper Translation Markup Language

2. Jaká organizace vytváří, udržuje a rozvíjí webové standardy?

- a) The World Wide Web Consortium
- b) Microsoft
- c) Mozilla
- d) Google

3. Vyberte správný HTML element vyjadřující nadpis nejvyšší úrovně.

- a) <heading>
- b) <head>
- c) <h6>
- d) <h1>

4. Vyberte správný HTML element vyjadřující řádkový zlom.

- a)

- b) <lb>
- c) <break>
- d) <lr>

Obrázek 13: Ukázkový soubor s otázkami z předmětu IWWW

```

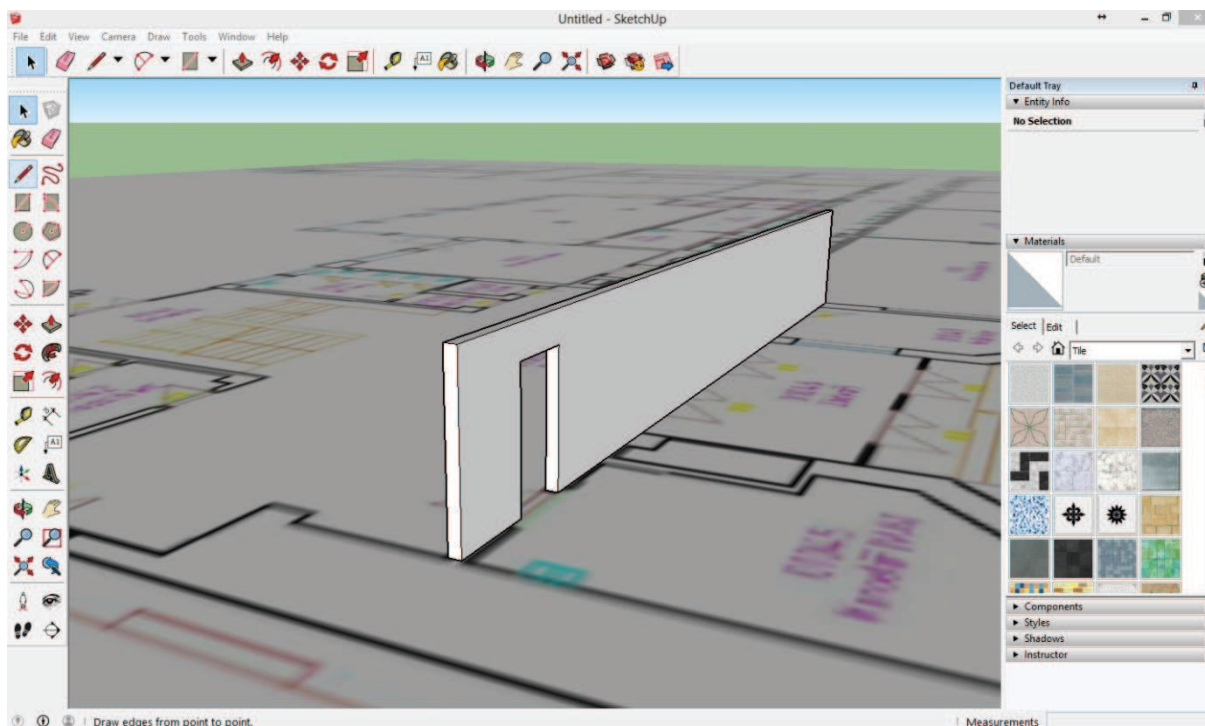
<?xml version="1.0" encoding="UTF-8"?>
<SeznamOtazek>
  <Otazky>
    <Otazka name="Co znamená akronym HTML?">
      <OdpovedA>Hyper Text Markup Language</OdpovedA>
      <OdpovedB>Home Tool Markup Language</OdpovedB>
      <OdpovedC>Hyper Translation Markup Language</OdpovedC>
      <OdpovedD>Hyperlinks and Text Markup Language</OdpovedD>
      <Predmet>Návrh a tvorba WWW stránek</Predmet>
    </Otazka>
    <Otazka name="Jaká organizace vytváří, udržuje a rozvíjí webové standardy?">
      <OdpovedA>The World Wide Web Consortium</OdpovedA>
      <OdpovedB>Mozilla</OdpovedB>
      <OdpovedC>Google</OdpovedC>
      <OdpovedD>Microsoft</OdpovedD>
      <Predmet>Návrh a tvorba WWW stránek</Predmet>
    </Otazka>
    <Otazka name="Vyberte správný HTML element vyjadřující řádkový zlom.">
      <OdpovedA>br</OdpovedA>
      <OdpovedB>break</OdpovedB>
      <OdpovedC>lr</OdpovedC>
      <OdpovedD>lb</OdpovedD>
      <Predmet>Návrh a tvorba WWW stránek</Predmet>
    </Otazka>
    <Otazka name="Vyberte správný HTML element vyjadřující nadpis nejvyšší úrovně.">
      <OdpovedA>h1</OdpovedA>
      <OdpovedB>heading</OdpovedB>
      <OdpovedC>h6</OdpovedC>
      <OdpovedD>head</OdpovedD>
      <Predmet>Návrh a tvorba WWW stránek</Predmet>
    </Otazka>
  </Otazky>
</SeznamOtazek>

```

Obrázek 14: Ukázka XML souboru s předmětem IWWW

4.2 Tvorba 3D modelu fakulty

Po sběru dat nastal přechod na tvorbu samotného modelu. Hlavním nástrojem v této části práce byl program SketchUp. Tato volba usnadnila práci hlavně tím, že software SketchUp podporuje importování souborů ve formě obrázků. Stačilo tedy stavební plány, obdržené ze sekretariátu fakulty, převést do JPG souboru a nainportovat do prostředí softwaru SketchUp. Takto vložený obrázek posloužil jako šablona, podle které mohly být vykresleny zdi fakulty a následně vytaženy do potřebné výšky. Názorně tento proces zobrazuje Obrázek 15.

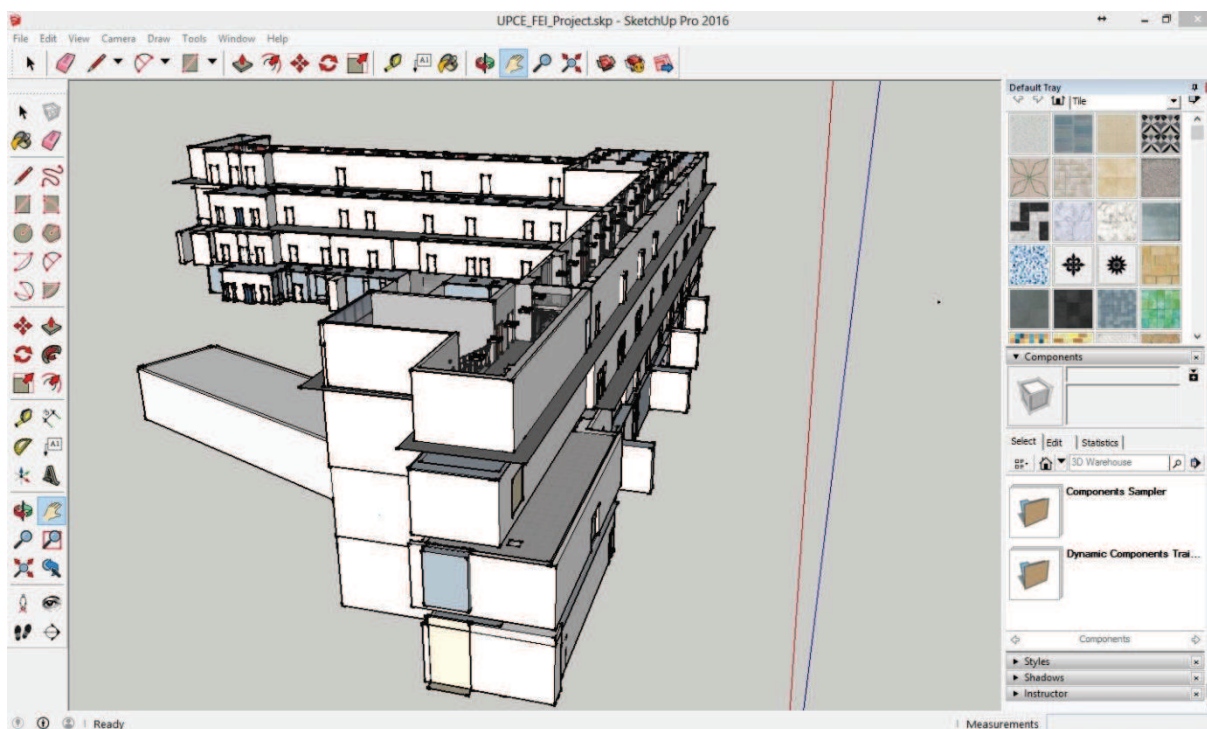


Obrázek 15: Ukázka použití obrázku jako šablony v softwaru SketchUp

Podobným způsobem následovala práce na všech čtyřech podlažích fakulty. Jediný rozdíl byl ve schodištích. Schodiště jsou na této škole celkem 3 a každé úplně jiné. Každé má jiný počet schodů, každé má jinou výšku schodu a každé má jinou šířku schodu. Proto bylo třeba každé schodiště modelovat jednotlivě (pouze v rámci prvního podlaží, dále byly tyto schodiště použity i pro další podlaží, kam byly vloženy jako hotové komponenty). Při tvorbě schodišť bylo použito ručně dělaných nákresů a detailního měření schodišť na fakultě (vizte Příloha A).

Po dokončení hrubé stavby modelu následovalo plnění stavby interiérovým zařízením. Různé dveře, lavičky, stoly, automaty na kávu či jen obyčejné odpadkové koše musely být opět zvlášť vytvořeny. Či staženy z obchodu WareHouse. WareHouse je oficiální místo, kde mohou vývojáři sdílet své 3D modely, vytvořené v programu SketchUp, s ostatními vývojáři. Po naplnění fakulty nábytkem přišla na řadu práce na informačních cedulích. Zde opět bylo třeba vycházet z reality a přesně navrhnout obdobný objekt. Zejména bylo zapotřebí dodržet styl písma, který patří pod jednotný styl univerzity. Konkrétně je to písmo zvané Gill Sans. Manuál pro dodržení grafického stylu univerzity lze nalézt na webových stránkách Univerzity Pardubice. Všechny tyto grafické prvky, jako informační cedule a jiné nápisy, musely být do modelu vloženy jako textury, aby korektně korespondovaly s povrchem modelu.

Po vytvoření jednotlivých podlaží následovalo zkompletování všech čtyř pater do jednoho celku. Největší problém nastal opět při zpracování schodišť, aby jednotlivá patra korektně navazovala a působila jednotným, celistvým dojmem, bylo třeba dodělat ještě pár stavebních úprav a dále vytvořit stropy se stropním osvětlením. Jak vypadá výsledný model fakulty lze vidět na následujícím obrázku (Obrázek 16).



Obrázek 16: Výsledný 3D model fakulty z programu SketchUp

4.3 Vývoj samotné hry

Prerekvizity (3D model, testy) jsou splněny, zbývá vymyslet základní scénář, jak bude samotná hra probíhat.

4.3.1 Scénář

Cílem hry bude absolvovat všechny zkoušky z vybraného studijního oboru. Z každého vyplněného testu hráč získá hodnocení v rozsahu 1–4. Přičemž 4 znamená nedostatečně a 1 výborně. Na každý předmět bude mít hráč 3 pokusy k úspěšnému zvládnutí testu. Pokud ani po těchto třech pokusech nebude jeho známka z daného předmětu alespoň 3, hra končí. Naopak, když se hráči podaří splnit všechny předměty se známkou 3 a lépe, tak zvítězil. Jak tedy bude celá hra probíhat: po spuštění hry se hráči zobrazí dialogové okno, které ho přivítá ve hře a seznámí s průběhem hry. Hráč by měl zadat své celé jméno a vybrat si studijní obor, který chce absolvovat. Po vybrání oboru se ocitne v prostorách školy, kde se bude moci volně pohybovat a navštěvovat kabiny jednotlivých kantorů. V každém kabinetu na něj bude čekat

teoretický test z vyučovaného předmětu kantora. Při vstupu do kabinetu se hráči zobrazí dialogové okno s informační cedulkou (obsahující jméno, email, telefon a fotografii učitele) a možností absolvování testu. Po vybrání této možnosti se zobrazí formulář s testem, do kterého se nahrají data z XML souboru daného předmětu. Hráč bude moci u každé otázky vybrat vždy pouze jednu odpověď. Po ukončení testu se dozví svůj výsledek a správnou odpověď každé otázky. Následně se vrátí zpět do školy, kde bude pokračovat k dalšímu kabinetu. Během prohlídky školy se klávesou Escape umožní kontrolovat aktuální stav zkoušek, společně s hodnocením jednotlivých předmětů a počtem uplynulých pokusů. Po každém testu program zkontroluje stav zkoušek hráče a zhodnotí, zda již byly všechny zkoušky splněny, nebo naopak vypršel limit počtu pokusů u nějaké nesplněné zkoušky a je tedy čas hru ukončit (vizte Příloha B).

Dle získaných testů, které byly obdrženy od garantů jednotlivých předmětů, obsahují studijní obory v této hře předměty, které lze nalézt v následující tabulce (Tabulka 1).

Tabulka 1: Soupis předmětů v jednotlivých studijních oborech hry

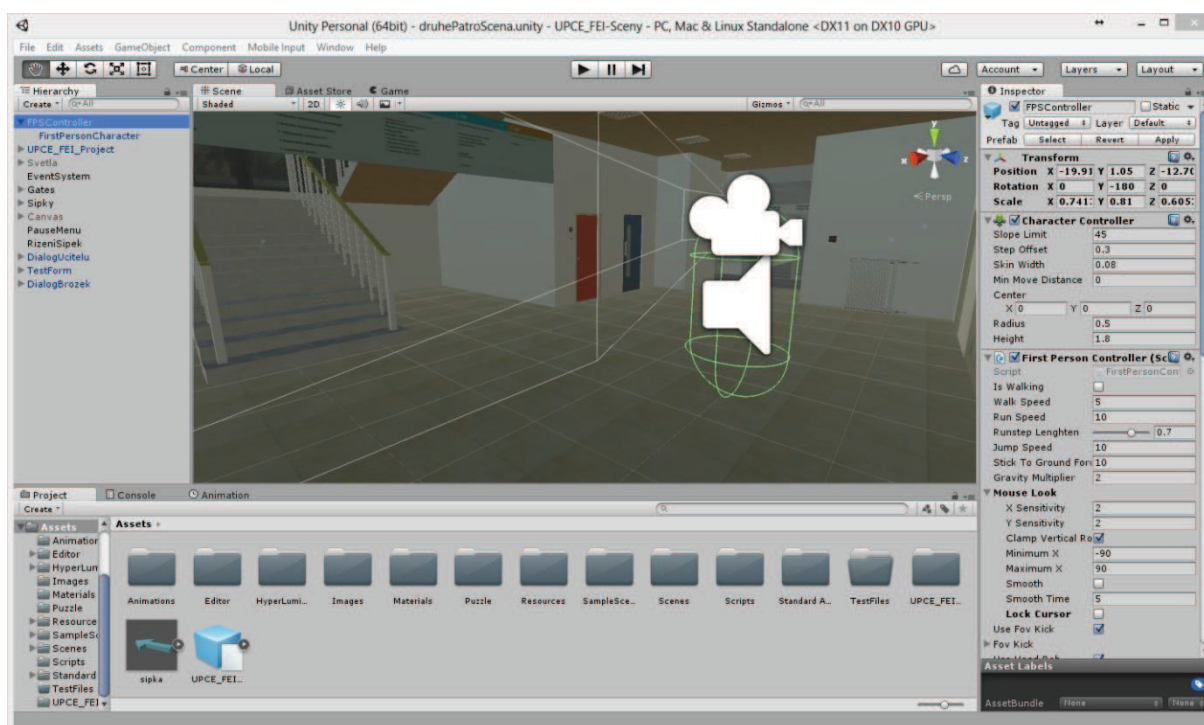
Povinné předměty		Nepovinné předměty
Informační technologie	Řízení procesů	IT + Řízení procesů
Návrh a tvorba WWW stránek	Spolehlivost a bezpečnost elektronických systémů	Programovací techniky v jazyce Java
Modelování ve výpočtových SW	Modelování ve výpočtových SW	Úvod do nových médií a komunikace
Architektura počítačů	Architektura počítačů	Grafické editory
Databázové systémy I	Automatizace	Historie techniky 1
Jazyk C++ I	Sdělovací technika	Vývoj počítačových her
Lineární algebra	Lineární algebra	
Modelování a simulace		
Počítačové sítě I		

Povinné předměty z Tabulky 1 hráč musí ve svém zvoleném oboru splnit. Nepovinné předměty nezasahují do celkové klasifikace hráče, nicméně si je může alespoň vyzkoušet.

4.3.2 Programování

3D model fakulty ve formátu FBX byl importován do herního enginu Unity jako nový asset. Po vložení nového modelu do prostředí Unity je důležité u daného objektu zaškrtnout volbu *Generate Colliders*. Tím se zabrání například procházení zdí a zajistí se, aby model

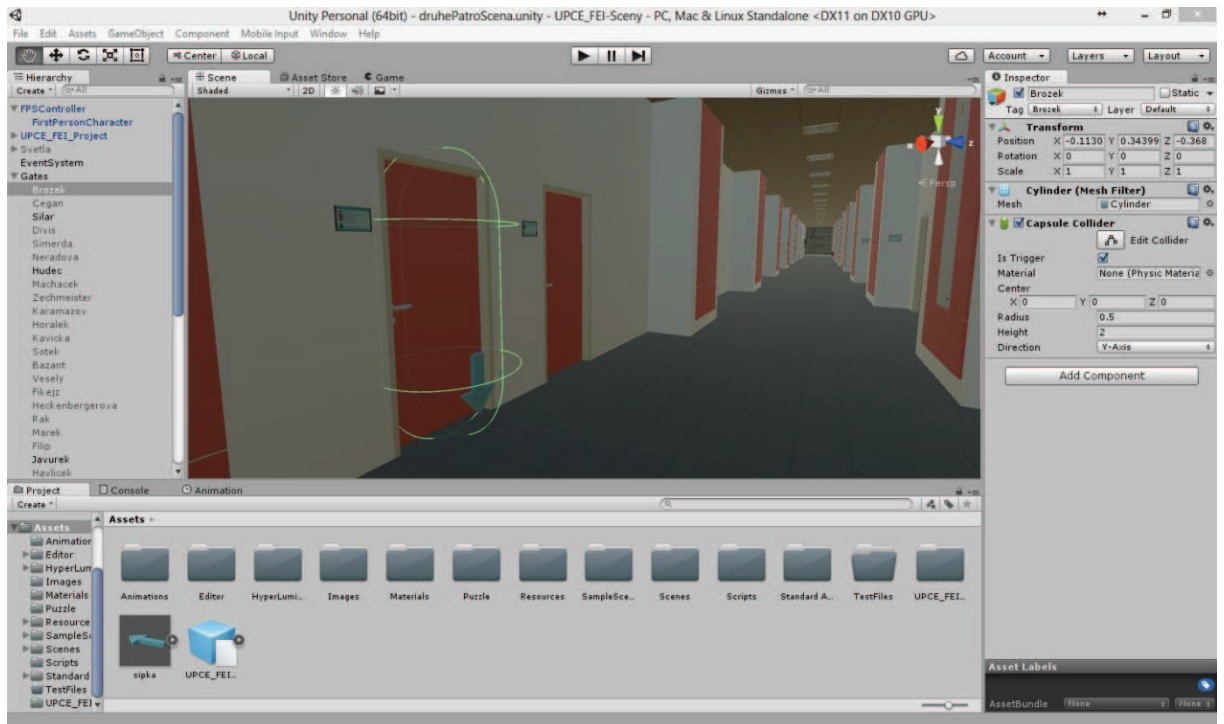
nefungoval jen jako grafický objekt, ale hlavně jako fyzický a mohlo se tak po něm pohodlně pohybovat. Pohyb po modelu byl zajištěn pomocí First Person Character Controlleru (dále jen FPS Controller), který je součástí balíčku Standard Assets. Balíček Standard Assets vydává sama společnost Unity Technologies a je volně dostupný v obchodě Asset Store. FPS Controller je vlastně Game Object, obsahující Capsule Collider a kameru. Capsule Collider zajišťuje možnost kolizí (kontaktů) s ostatními fyzickými objekty ve hře. Hráč tak nebude ve hře „létat“ a procházet zdmi. Kamera, nastavená na vrcholku Capsule Collideru, přenáší obraz hráči a dodává efekt vidění hry z první osoby. FPS Controller dále obsahuje skripty k ovládání celého objektu. Stará se o kameru a o pohyb. Pohyb je zde ovládán šipkami a myší. Dále pak klávesa Shift zrychluje objekt, mezerník pak umožňuje hráči skákat. Celý controller je navíc doplněn o zvuky chůze, které dodávají celému objektu velmi realistický dojem. Jak FPS Controller vypadá v prostředí herního engine Unity je vidět na následujícím obrázku (Obrázek 17).



Obrázek 17: FPS Controller v herním engine Unity

Po zajištění pohybu ve hře následovala práce na zpřístupnění kabinetů jednotlivých kanceláří. Toho bylo dosaženo opět pomocí Capsule Collideru, pracovně označeného jako Gate. Tento Gate byl zbaven veškerých textur a postaven přímo před dveře každého kabinetu. Dále byl nastaven tak, aby fungoval jako trigger. Tím bylo dosaženo toho, že se mohlo kontrolovat, zda se FPS Controller (hráč) ocitl v Gatu. Pokud se tomu tak stalo, mohla se vyvolat

požadovaná akce. Jak bylo zmíněno, Gate byl zbaven textur, čili byl neviditelný – což je pro komponentu nahrazující GUI³ nepřijatelné. Tento problém byl vyřešen návrhem šipky, lákající hráče ke vstupu do kabinetu. Na tuto šipku byla navíc přiřazena jednoduchá animace k zefektivnění vizuálnosti hry. Ke každému Gateu byl navíc přiřazen *Tag* se jménem kantora, vlastníci daný kabinet. *Tag* je jedinečné označení Game Objectu, díky kterému lze rozlišovat více stejných Game Objectů. Vyobrazení Gateu a šipky v prostředí Unity lze vidět na Obrázku 18.



Obrázek 18: Capsule Collider před vstupem do kabinetu v Unity

Existuje tedy hráč, volně se pohybující po budově a dále vstupní brány, které při střetu s hráčem mohou vyvolat nějakou akci. K tomu, aby se celková hra rozpochovala a korektně fungovala podle předchozího scénáře, bylo potřeba vytvořit několik tříd, řídících běh celé hry. Hra tedy obsahuje následující vlastní třídy:

- Introduction,
- UserConfiguration,
- GatesController,
- Configuration,
- PlayerController,
- DialogImage,

³ Graphical User Interface

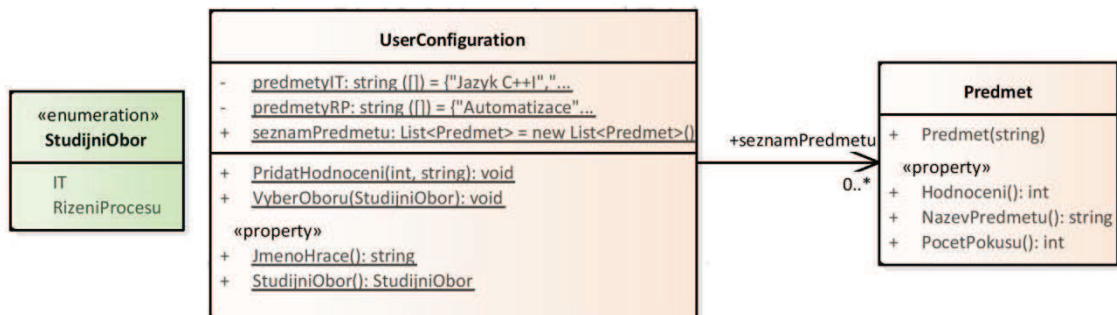
- ItemLoader,
- TestResult,
- SubjectResults.

Introduction

Třída `Introduction` slouží ke zpracování vstupních dat úvodního dialogového okna při spuštění aplikace. Poté co hráč zadá své jméno a vybere studijní obor, který chce absolvovat, zadaná data se nastaví do třídy `UserConfiguration`.

UserConfiguration

Třída `UserConfiguration` uchovává informace o hráči. Nachází se zde jméno hráče, zvolený studijní obor hráče a seznam předmětů, které hráč ve svém oboru musí absolvovat. Seznam předmětů je ve formě datové struktury `List`, která v sobě drží objekty typu `Predmet`. Třída `Predmet` dále obsahuje informace o názvu předmětu, hodnocení předmětu a počtu uplynulých pokusů splnění předmětu. Po zvolení studijního oboru ve třídě `Introduction` se seznam předmětů ve třídě `UserConfiguration` naplní předměty daného oboru. Soupis názvů předmětů každého oboru je uchováván v poli. Obrázek 19 představuje UML diagram této třídy.



Obrázek 19: UML diagram třídy `UserConfiguration`

GatesController

`GatesController` je třída, která se aktivuje po výběru studijního oboru. V tomto momentě se zkontroluje studijní obor hráče a podle něj se zobrazí vstupní brány (Gates) a šipky před kabinety kantorů. Je tomu tak z toho důvodu, aby hráč, který si například vybral obor Informační technologie, nebyl zbytečně lákán k plnění předmětů oboru Řízení procesů, protože ty mu do klasifikace započítány stejně nebudou. Dále se vhodně zbarví textury šipek

dle předmětů. Šipky, lákající ke splnění povinných předmětů, mají červenou barvu, zatímco šipky u kabinetů nepovinných předmětů mají modrou barvu.

Configuration

`Configuration` je statická třída, udržující informace o aktuálním chodu hry. Nastavuje se zde například jméno zvoleného předmětu, cesta k XML souboru zvoleného předmětu, cesta k obrázku kantora, vyučující daný předmět, jméno posledního navštíveného kantora atd. Jednotlivé třídy si pak mezi sebou tyto informace půjčují a to stylem, že jedna nastaví, druhá použije.

PlayerController

Třída `PlayerController` obsahuje klíčové metody a využívá všech, zde zmíněných, tříd. Je přiřazena pod `FPS Controller`, aby mohla využívat jeho `Capsule Collider`. V této třídě se nachází hlavně metoda `OnTriggerEnter (Collider other)`. Tato metoda kontroluje, zda došlo ke kolizi s jiným objektem, který je nastaven jako trigger. V případě této hry jsou takovými objekty, sloužící jako triggery, vstupní brány před kabinety (`Gates`). Pokud k něčemu takovému dojde, tak se pomocí parametru metody (v tomto případě `Collider other`) zjistí, jaký `Gate` to byl a podle toho se vykoná daná akce. Konkrétně se porovná `tag` daného `Gatu`. Podle tohoto `tagu` se do třídy `Configuration` nastaví cesta k XML souboru daného předmětu, dále jméno kantora, cesta k obrázku kantora a spustí se dialogové okno, tážající se na spuštění testu. O to už se ale stará třída `DialogImage`.

DialogImage

Třída `DialogImage` slouží k zobrazení dialogového okna, tážajícího se na spuštění testu a hlavně k zobrazení informační cedulky daného kantora. Využívá cesty k obrázku, která je uložena ve třídě `Configuration` a obrázek dále zobrazí do `Canvasu`. Také se zde nachází tlačítka na spuštění testu, či na odchod zpět do prostředí školy. Při volbě spuštění testu se volá instance třídy `ItemLoader`.

ItemLoader

`ItemLoader` využívá ke své práci třídy `Item` a `ItemContainer`. Ve třídě `ItemContainer` se podle zadané cesty k XML souboru (která je udržována ve třídě `Configuration`) načtou data o daném předmětu, zejména jeho testové otázky a název předmětu. Všechny tyto informace se uloží do datové struktury `List`, uchovávající objekty

typu `Item`, kterou zapouzdřuje třída `ItemContainer`. Ve třídě `ItemLoader` se pak náhodně vyberou čtyři prvky (otázky) z tohoto `Listu` a jimi se pak vyplní formulář s teoretickým testem. Po vyplnění teoretického testu se test zkontroluje pomocí třídy `TestResult`. Dále se objeví tlačítko s možností návratu do prostor školy.

TestResult

Ve třídě `TestResult` se zkontrolují zaškrtnuté odpovědi hráče, sečtou se správné odpovědi a vyhodnotí se výsledek testu. Dále se zbarví zeleně odpovědi, které měly být správně a případně červeně odpovědi, které byly zaškrtnuté chybně. Hodnocení se odešle do třídy `UserConfiguration`, ve které je připravena metoda `PridatHodnoceni`, kde parametrem je daná známka a název předmětu. V této metodě se dále v seznamu předmětů nalezne daný předmět dle jeho názvu, přiřadí se hodnocení a počet pokusů přibude o 1. Dále dojde ke kontrole všech předmětů, jestli nejsou již splněny, nebo zda aktuálně provedený test nebyl již čtvrtým pokusem daného předmětu a nemá dojít k ukončení hry.

SubjectResults

Během hry lze pomocí klávesy `Escape` kontrolovat průběžné výsledky zkoušek. O to se stará právě třída `SubjectResults`. Ta vypisuje do dialogového okna informace ze seznamu předmětů hráče, které jsou uchovávány ve třídě `UserConfiguration`.

Takto nastavené třídy byly implementovány do prostředí Unity v podobě `Game Objectů`. Celková hra se již dále exportovala do spustitelné aplikace. Unity nabízí export hry do více než dvaceti platforem. Platformy, které Unity nabízí, jsou například:

- webový prohlížeč,
- osobní počítač s operačním systémem Windows, Linux či Mac,
- iOS,
- Android,
- Xbox 360,
- Flash Player,
- PlayStation a mnoho dalších.

V případě této práce byla hra exportována na platformě webového prohlížeče. Testování běhu hry je více popsáno v následující kapitole.

5 Uživatelské testování

Po exportování hry z herního enginu Unity následovalo testování její funkčnosti. Během testování bylo zjištěno, že hra má velmi velké grafické požadavky na počítač. To z důvodu velkého obsazení textur v 3D modelu fakulty a vůbec celkové velikosti modelu. Dále pak díky nasvícení scény, které velmi zpomalovalo průběh hry (vizte Příloha C). Toto nasvícení bylo nakonec ze hry odstraněno a v postupu vývoje hry se již neuvádí. Po této opravě se již běh hry značně zlepšil, avšak i tak lze spouštět na plné detaily pouze na počítačích s vysokým výkonem grafické karty. Celkový problém s během hry by mohla vyřešit funkce zvaná Occlusion Culling. Ta funguje na principu toho, že se objekty, které jsou součástí scény, vykreslují pouze v momentě, kdy se na ně hráč dívá (když jsou zaměřeny kamerou). Do té doby jsou neaktivní a engine se tak nemusí o ně starat, což přispívá ke zrychlení běhu celé hry. Tato bakalářská práce byla vytvořena v bezplatné verzi Unity Personal, která tuto funkci nepodporuje. Funkce Occlusion Culling je dostupná až v placené verzi Unity Pro.

Další problém, který nastal, bylo velice zdlouhavé načítání scény s modelem fakulty. V první verzi hry byly veškeré dialogy a formulář s teoretickým testem umístěny do samostatných scén. Při návratu do scény s fakultou, tak musel hráč pokaždé zdlouhavě čekat, než se mu scéna znovu načte a hra tak byla sama o sobě neefektivní. Tento problém byl vyřešen tak, že jednotlivé scény s dialogy byly nahrazeny Canvasy, které se staly součástí hlavní scény. Celá hra se tedy odehrává pouze v jedné scéně a opakované zdlouhavé načítání tak bylo vypuštěno. Respektive nastane pouze jednou a to při prvotním spuštění hry.

Poslední problém, který bylo třeba vyřešit, byl skrytý kurzor myši. Jelikož k pohybu po fakultě je využíván FPS Controller, není kurzor myši vidět, protože k pohybu, kdy se chůze ovládá šipkami a pohled hráče myší, není kurzoru třeba. Avšak k vyplňování testů a potvrzování dialogů viditelnosti kurzoru třeba bylo. Musel tedy být zákaz kurzoru, který je implementován ve skriptu FPS Controlleru, zrušen a zakrývání či zobrazení kurzoru ovládat samostatným skriptem, dle momentální potřeby kurzoru ve hře.

Závěr

K vyhotovení praktické části bakalářské práce bylo zapotřebí vytvoření 3D modelu fakulty a jeho implementování do herního enginu Unity. Tento proces byl pro mě něčím novým, protože zkušenosti s 3D modelováním jsem do té doby žádné neměl, a tak byla tato práce pro mě v tomto směru určitě přínosná. Taktéž jsem si osvojil práci s herním enginem Unity a vyzkoušel si mnoho nových metod a funkcí, které tento software nabízí.

Celkový pocit z mé práce mám dobrý. Dopadla nad moje očekávání, protože původním úmyslem bylo pouze to, že hráč se bude moci volně pohybovat po fakultě a nic víc. Scénář v té podobě, v jaké je teď, se postupně dotvářel během vývoje práce.

Ačkoli se na začátku zdálo, že v této práci použiji pouze 3D modelování a znalosti z předmětu Vývoj počítačových her, tak nakonec jsem využil i znalostí z Datových struktur a Objektově orientovaného programování, které velmi usnadnily vývoj a zefektivnily práci.

Výsledná aplikace se dá určitě dále využít k jejímu původnímu záměru, tj. k propagaci Fakulty elektrotechniky a informatiky. Jelikož je práce v podobě webové aplikace, nebyl by zajisté problém umístit ji na webové stránky fakulty, kterou navštěvují budoucí uchazeči o studium, kde by si mohl každý volně projít prostory fakulty a vyzkoušet své znalosti z vybraných předmětů.

Aplikace je ovšem omezena statickou podobou modelu fakulty. Jelikož reálná Fakulta elektrotechniky a informatiky Univerzity Pardubice se neustále mění a modernizuje, musel by se s ní měnit i model ve hře, aby si zachoval aktuálnost a důvěryhodnost. K tomu by domohlo neustálého spravování modelu, či vytvoření externí aplikace pro správu tohoto modelu. Nestálé prvky, jako cedulky a informační panely, by měly své texty uchovávané v databázi a ta by se průběžně aktualizovala.

Dalším vylepšením by mohla být GPS navigace. Hráč by zadal jméno hledaného kantora, načež by se mu zobrazila trasa, která by ho dovedla k požadovanému kabinetu.

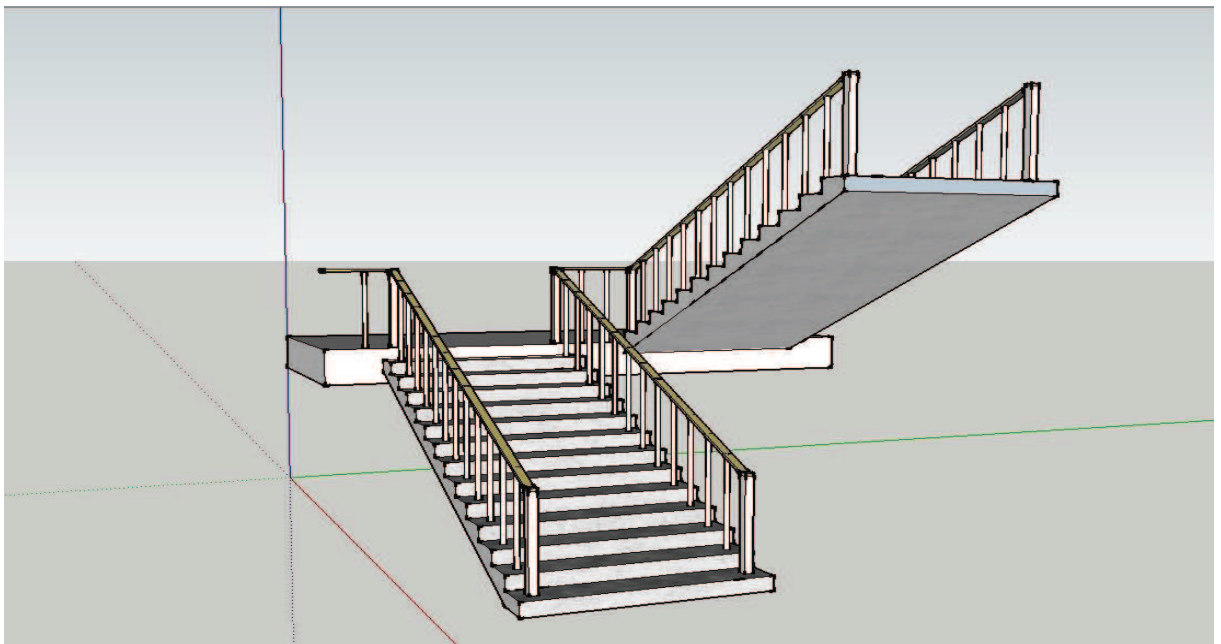
Použitá literatura

- [1] TAUFER, Ivan, Josef KOTYK a Milan JAVŮREK. *Jak psát a obhajovat závěrečnou práci: bakalářskou, diplomovou, rigorózní, disertační, habilitační*. Pardubice: Univerzita Pardubice, 2009. ISBN 978-80-7395-157-3.
- [2] SLICK, Justin. What Is 3D Modeling? *Lifewire* [online]. 2016 [cit. 2017-04-21]. Dostupné z: <http://www.lifewire.com/what-is-3d-modeling-2164>
- [3] Best 3D Modeling Software. *G2 Crowd* [online]. 2017 [cit. 2017-04-21]. Dostupné z: <http://www.g2crowd.com/categories/3d-modeling>
- [4] Materials, Shaders & Textures. *Unity User Manual* [online]. Unity Technologies, 2017 [cit. 2017-04-21]. Dostupné z: <https://docs.unity3d.com/Manual/Shaders.html>
- [5] ČEVELA, Lubomír. Výměna dat mezi CAD systémy. *Root.cz* [online]. 2000 [cit. 2017-04-21]. ISSN 1212-8309. Dostupné z: <https://www.root.cz/clanky/vymena-dat-mezi-cad-systemy/>
- [6] SKOUPIL, Jaromír. Obecný úvod o CAD systémech. *Systémy CAD* [online]. Brno: Mendelova zemědělská a lesnická univerzita v Brně, 2008 [cit. 2017-04-21]. Dostupné z: <https://is.mendelu.cz/eknihovna/opory/index.pl?opora=1371>
- [7] BRDIČKA, Bořivoj. Grafické editory. *Učení s počítačem* [online]. Praha, 1995 [cit. 2017-04-22]. Dostupné z: <http://it.pedf.cuni.cz/~bobr/ucspoc/>
- [8] TIŠNOVSKÝ, Pavel. Grafická knihovna OpenGL (22): texturování. *Root.cz* [online]. 2003 [cit. 2017-04-21]. ISSN 1212-8309. Dostupné z: <https://www.root.cz/clanky/opengl-22-texturovani/>
- [9] WARD, Jeff. What is a Game Engine? *GameCareerGuide* [online]. 2008 [cit. 2017-04-21]. Dostupné z: http://www.gamecareerguide.com/features/529/what_is_a_game_.php
- [10] MOHEBALI, Ali a Thiam Kian CHIEW. Redefining Game Engine Architecture through Concurrency. *NEW TRENDS IN SOFTWARE METHODOLOGIES, TOOLS AND TECHNIQUES*. Amsterdam: IOS Press, 2014, s. 767 – 782. ISBN 978-1-61499-433-6.
- [11] Asset Workflow. *Unity User Manual* [online]. Unity Technologies, 2017 [cit. 2017-04-21]. Dostupné z: <https://docs.unity3d.com/Manual/AssetWorkflow.html>
- [12] KOSEK, Jiří. *XML pro každého*. 1. Vydání. Praha: Grada Publishing, 2000. 164 s. ISBN 80-7169-860-1. Podpora dostupná z <http://www.kosek.cz/xml/>

- [13] Animace. *Wikipedia, otevřená encyklopedie* [online]. 2016 [cit. 2017-04-21]. Dostupné z: <https://cs.wikipedia.org/wiki/Animace>
- [14] SKŘIVAN, Jaromír. SQL – jak na triggeru. *Interval.cz* [online]. 2000 [cit. 2017-04-21]. Dostupné z: <https://www.interval.cz/clanky/sql-jak-na-triggeru/>
- [15] Vývoj her. *PCWorld* [online]. 1998 [cit. 2017-04-21]. Dostupné z: <http://pcworld.cz/archiv/vyvoj-her-19475>
- [16] KREJČÍ, Jan. Jak vznikají hry II. *PCWorld* [online]. 1998 [cit. 2017-04-21]. Dostupné z: <http://pcworld.cz/archiv/jak-vznikaji-hry-ii-19607>
- [17] VOJÍŘ, Milan. 3D modelování v Google SketchUp. *Metodický portál inspirace a zkušenosti učitelů* [online]. 2009 [cit. 2017-04-21]. ISSN 1802-4785. Dostupné z: <http://spomocnik.rvp.cz/clanek/12365/3D-MODELOVANI-V-GOOGLE-SKETCHUP.html>
- [18] Types of light. *Unity User Manual* [online]. Unity Technologies, 2017 [cit. 2017-04-21]. Dostupné z: <https://docs.unity3d.com/Manual/Lighting.html>

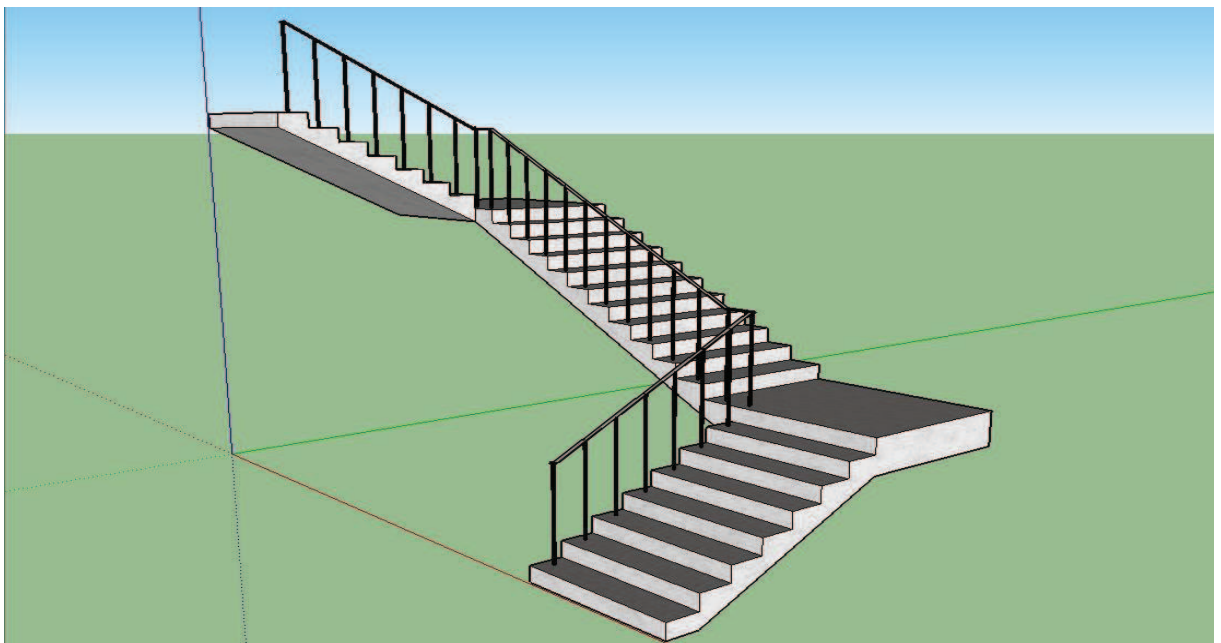
1 Příloha A

1.1 Model hlavního schodiště



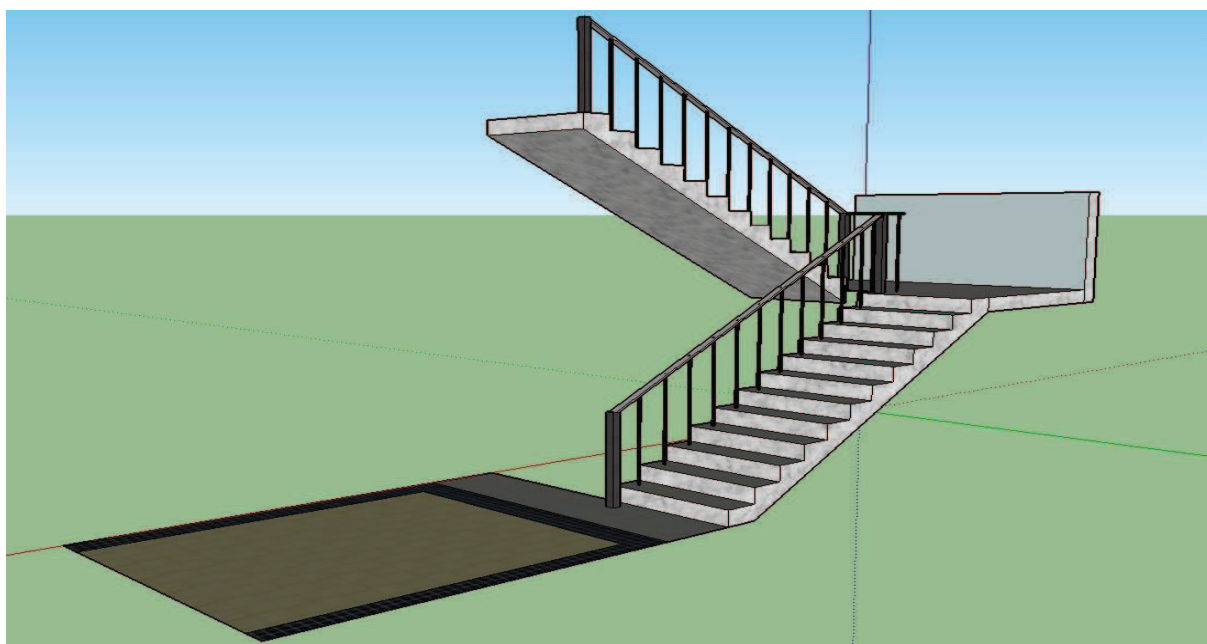
Obrázek 20: Příloha - Model hlavního schodiště

1.2 Model druhého schodiště



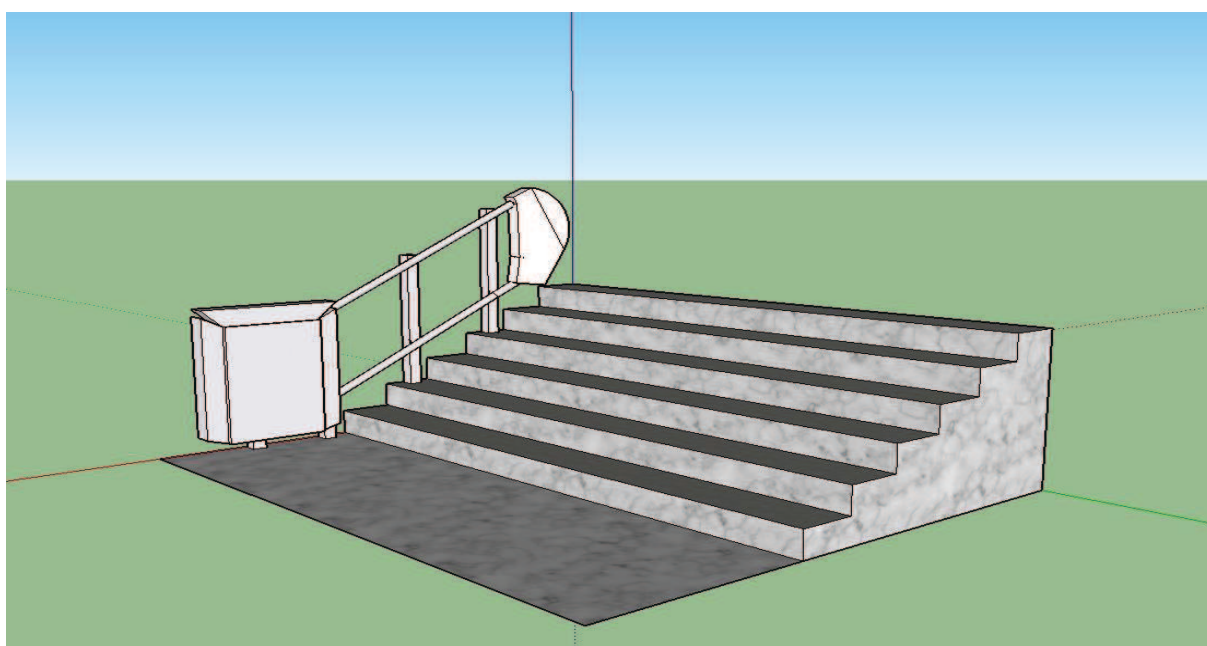
Obrázek 21: Příloha - Model druhého schodiště

1.3 Model bočního schodiště



Obrázek 22: Příloha - Model bočního schodiště

1.4 Model schodiště s bezbariérovým přístupem



Obrázek 23: Příloha - Model schodiště s bezbariérovým přístupem

2 Příloha B

2.1 Splash Screen



Obrázek 24: Příloha - Splash Screen

2.2 Úvodní dialog

Vítejte na Fakultě elektrotechniky a informatiky Univerzity Pardubice

K tomu, abyste úspěšně prošli studiem na této fakultě, musíte splnit několik zkoušek z Vámi vybraného oboru. Na každou zkoušku máte pouze 3 pokusy. Hodnocení každé zkoušky jsou 1, 2, 3 a 4. Přičemž známka 4 znamená Nedostatečně. Pokud se Vám nepodaří dosáhnout lepší známky než 3 ani na 3. pokus - hra končí! Během hry můžete své výsledky sledovat stisknutím klávesy Escape. Níže zadejte své celé jméno a vyberte obor, který chcete splnit.

Zadejte své celé jméno:

Výběr oboru:

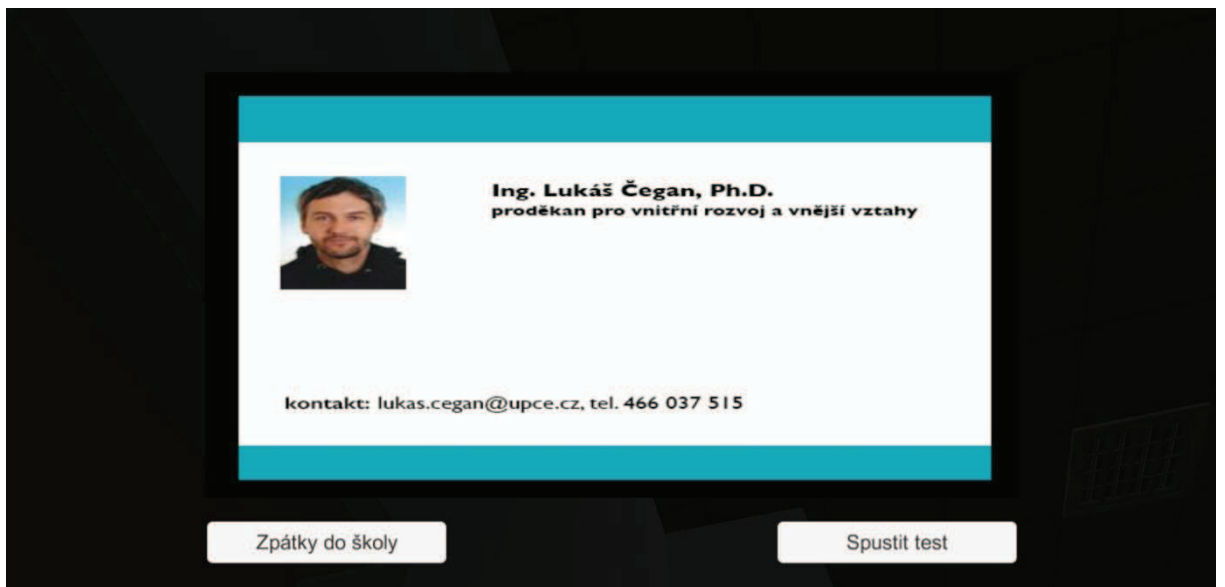
Obrázek 25: Příloha - Úvodní dialog

2.3 Prvotní zobrazení prostor školy ve hře



Obrázek 26: Příloha - Prostory školy ve hře

2.4 Dialog s informacemi o kantorovi a možností spuštění testu



Obrázek 27: Příloha - Dialog vybraného kantora

2.5 Testový formulář

Návrh a tvorba WWW stránek

Vyberte správné označení HTML elementu vyjadřujícího řádkový zlom.

- lb
- break
- lr
- br

Jaká organizace vytváří, udržuje a rozvíjí webové standardy?

- Mozilla
- The World Wide Web Consortium
- Google
- Microsoft

Co znamená akronym HTML?

- Hyper Text Markup Language
- Home Tool Markup Language
- Hyper Translation Markup Language
- Hyperlinks and Text Markup Language

Vyberte správné označení HTML elementu vyjadřujícího nadpis nejvyšší úrovně.

- h6
- heading
- h1
- head

Dokončit a odeslat Zpátky do školy

Úspěšnost vašeho testu je 100%

Obrázek 28: Příloha - Testový formulář

2.6 Zobrazení průběžných výsledků zkoušek

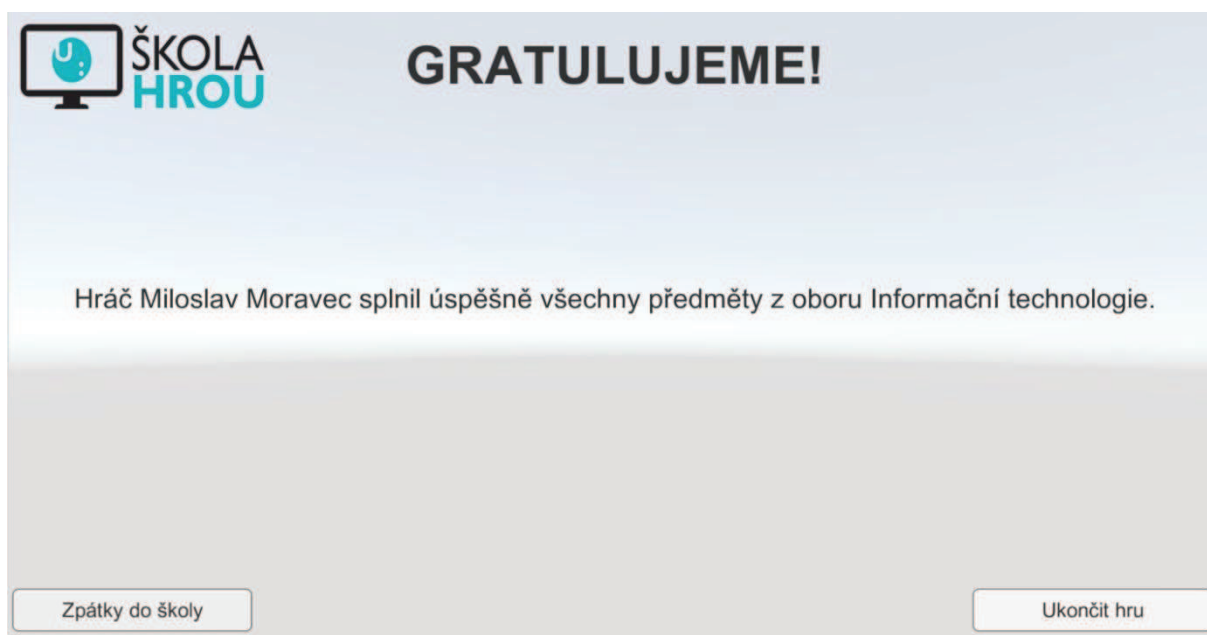
Průběh zkoušek

Název předmětu	Hodnocení	Pokus
Jazyk C++I	0	0
Databázové systémy I	0	0
Lineární algebra	0	0
Modelování a simulace	0	0
Modelování ve výpočtových SW	0	0
Počítačové sítě I	0	0
Statistika	0	0
Architektura počítačů	0	0
Úvod do jazyka C	0	0
Návrh a tvorba WWW stránek	1	1
Základy programování	0	0

Ukončit hru

Obrázek 29: Příloha - Zobrazení průběžných výsledků zkoušek

2.7 Zobrazení informace o výhře



Obrázek 30: Příloha - Zobrazení informace o výhře

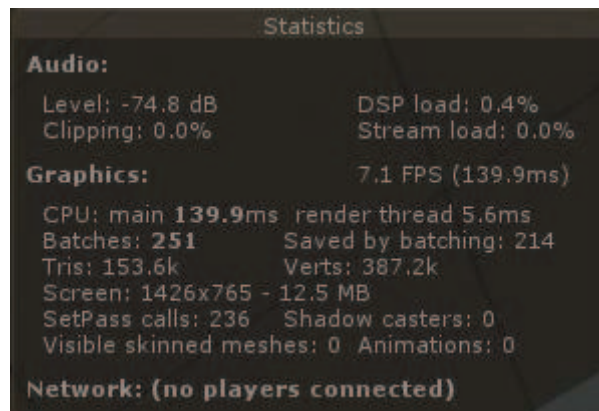
2.8 Zobrazení informace o prohře



Obrázek 31: Příloha - Zobrazení informace o prohře

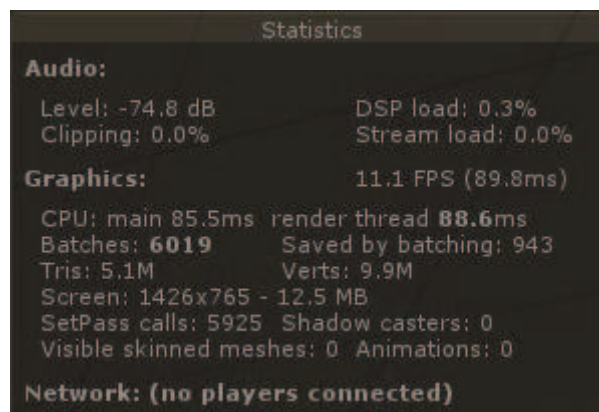
3 Příloha C

3.1 Statistika o výkonu hry – bez osvětlení scény



Obrázek 32: Příloha - Statistika bez světla

3.2 Statistika o výkonu hry – s osvětlenou scénou



Obrázek 33: Příloha - Statistika se světly

4 Příloha D

4.1 Textura vstupu do fakulty



Obrázek 34: Příloha - Textura vstupu do fakulty

4.2 Textura oken na hlavním schodišti



Obrázek 35: Příloha - Textura oken na hlavním schodišti

4.3 Textura oken v respiriu



Obrázek 36: Příloha - Textura oken v respiriu

4.4 Textura okna na bočním schodišti



Obrázek 37: Příloha - Textura okna na bočním schodišti

5 Příloha E

5.1 Přiložené DVD

Obsah přiloženého DVD:

- zdrojové kódy aplikace (spustitelný projekt v prostředí Unity),
- spustitelná aplikace pro platformu webového prohlížeče a
- elektronická podoba teoretické části práce ve formátu PDF.