



**České
vysoké
učení technické
v Praze**

Katedra počítačové grafiky a interakce

Webový portál pro přípravu podkladů pro rozšířenou realitu

Michal Mráz

**Vedoucí: Ing. David Sedláček, Ph.D.
Studijní program: Otevřená informatika
Studijní obor: Počítačové hry a grafika
Prosinec 2020**

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Mráz** Jméno: **Michal** Osobní číslo: **457055**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra počítačové grafiky a interakce**
Studijní program: **Otevřená informatika**
Specializace: **Počítačové hry a grafika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Webový portál pro přípravu podkladů pro rozšířenou realitu

Název bakalářské práce anglicky:

Web Portal for Augmented Reality Preparation

Pokyny pro vypracování:

Navrhněte a implementujte webový portál pro vytváření podkladů pro rozšířenou realitu. Portál bude umožňovat upload 3D souboru, definice jeho reference v reálném světě (ukotvení), navázání na události definované vedoucím práce, vytvoření anotací a stažení dat klientskou aplikací, která bude řešit samotnou vizualizaci (tvorba této aplikace není součástí zadání). Při práci postupujte podle metodiky User centered design (UCD) s ohledem na dostupnost cílové skupiny (učitelé SŠ a ZŠ, tvůrci výukového obsahu).

Práci otestujte na scénářích vytvořených ve spolupráci s vedoucím práce. Práce předpokládá navázání na již existující části portálu (editor výukových úkolů nevyužívajících rozšířenou realitu, serverová část portálu pro správu dat, klientské aplikace pro prohlížení obsahu na mobilním zařízení).

Primární použité technologie: X3D, X3Dom, JavaScript.

Seznam doporučené literatury:

[1] Behr, Johannes, Peter Eschler, Yvonne Jung and Michael Zöllner. "X3DOM: a DOM-based HTML5/X3D integration model." Web3D (2009).

[2] <http://www.x3dom.org>

[3] Augmented Reality Game Development, Micheal Lanham, Packt Publishing, 2017

[4] Dieter Schmalstieg, and Tobias Hollerer, Augmented Reality: Principles and Practice (Usability), Addison Wesley 2016

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. David Sedláček, Ph.D., katedra počítačové grafiky a interakce FEL

Jméno a pracoviště druhého(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **22.09.2020**

Termín odevzdání bakalářské práce: **05.01.2021**

Platnost zadání bakalářské práce: **30.09.2022**

Ing. David Sedláček, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Rád bych zde poděkoval vedoucímu mé bakalářské práce Ing. Davidu Sedláčkovi Ph.D. za poskytnuté konzultace a cenné rady. Také bych rád poděkoval mé rodině za podporu ve studiu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 7. prosince 2020

Abstrakt

Předmětem této bakalářské práce je návrh a implementace webové aplikace pro vytváření podkladů pro rozšířenou realitu. Aplikace je vyvíjena za pomoci technologií Javascript, HTML a CSS. Pro zobrazování 3D scén používá technologii X3DOM. Práce sestává z analýzy požadavků, průzkumu podobných již existujících řešení, návrhu a porovnání řešení, implementace, ukázky aplikace, testování s uživateli a z popisu možných zlepšení, která z velké části vycházejí ze zpětné vazby uživatelů.

Klíčová slova: rozšířená realita, AR, X3DOM, marker, GPS, editor

Vedoucí: Ing. David Sedláček, Ph.D.

Abstract

The object of this bachelor thesis is the design and implementation of a web application for creation of augmented reality content. The application is developed using Javascript, HTML and CSS. It uses X3DOM technology to display 3D scenes. The thesis consists of requirements analysis, research of similar existing solutions, solution draft, comparison of solutions, implementation, application preview, testing with users and a description of possible improvements, that are largely based on user feedback.

Keywords: augmented reality, AR, X3DOM, marker, GPS, editor

Title translation: Web portal for augmented reality preparation

Obsah

1 Úvod	1
1.1 Rozšířená realita.....	1
2 Analýza	5
2.1 Požadavky	5
2.2 Existující řešení	6
2.3 X3D uzly	7
3 Návrh řešení	11
3.1 Uživatelské rozhraní.....	12
3.2 Porovnání řešení.....	13
4 Implementace	15
4.1 Přehled objektů	15
4.2 Použité technologie	18
4.3 Implementační detaily	20
5 Ukázka aplikace	25
6 Testování	29
6.1 Výsledky testování	29
7 Budoucí vývoj	35
8 Závěr	37
Literatura	39
A Serializovaný scénář	43
B Instrukce k testování	47
C Obsah přiloženého CD	51

Obrázky

Tabulky

1.1 Příklad AR, která využívá marker	2
1.2 Příklad AR, která využívá GPS..	3
2.1 TARTT - vytváření interaktivního katalogu.	7
2.2 Augmara - umístování modelu na marker.....	7
2.3 Augment - nahrávání trackeru. . .	8
2.4 Augment - umístování modelu na tracker.....	8
3.1 Ukázka uživatelského rozhraní.	12
4.1 Objektové schéma scénáře.	16
4.2 Trojrozměrná reprezentace okolí GPS bodu (Karlovo náměstí)	23
5.1 Modely hraček umístěné ve scéně, která je ukotvená pomocí GPS souřadnic.	26
5.2 Přidávání spouštěče události . . .	27
5.3 Přidávání akce události	27

Kapitola 1

Úvod

Tato práce se zabývá návrhem a implementací webové aplikace pro vytváření obsahu pro rozšířenou realitu (augmented reality - AR, překlad slova augmentation - rozšíření, zvětšení, rozhojnění, rozmnožení). Aplikace je součástí projektu EduARd, jehož účelem je zatraktivnění výuky některých témat na základních a středních školách tím, že se do ní zařadí využití mobilních zařízení, jako jsou telefony a tablety. K tomu je potřeba pro tato zařízení vytvořit kvalitní obsah. V rámci projektu se vytvoří systém jak pro tvorbu tohoto obsahu, tak pro jeho prohlížení.

Rozšířená realita se v posledních letech dostala do širokého povědomí díky cenově dostupným mobilním zařízením s vysokým výkonem, která ji dokáží vykreslovat v reálném čase a také díky populárním mobilním aplikacím a hrám. S rozšířenou realitou se můžeme setkat například ve zdravotnictví, vzdělávání, v automobilovém, vojenském, herním, spotřebním a v reklamním průmyslu [24].

1.1 Rozšířená realita

Rozšířená realita propojuje virtuální a reálný svět, vytváří propojení mezi fyzickým světem a elektronickými informacemi. Na rozdíl od virtuální reality (VR), která umísťuje uživatele do kompletně počítačem generovaného prostředí, AR zobrazuje informace přímo propojené s fyzickým prostředím. AR přemostuje propast mezi virtuálním a reálným světem jak prostorově, tak kognitivně. V AR se uživateli digitální informace jeví jako součást reálného světa. Podle nejvíce uznávané definice dle Azuma (1997) [19] musí mít AR tyto tři charakteristiky: 1) kombinuje reálné a virtuální, 2) je interaktivní v reálném čase a 3) je propojená (registered) ve 3D. To znamená, že se AR nemusí projevat jen vizuálně, ale i zvukově, dotykově či dokonce čichově

1. Úvod

nebo chuťově. V naší práci se ale věnujeme projevům vizuálním. Požadavek ovládnutí a propojení s reálným světem v reálném čase implikuje, že uživatel může interaktivně měnit pohled na AR a při tom počítačem generovaná rozšíření zůstanou propojená s odpovídajícími reálnými objekty. Interaktivita implikuje, že interakce člověka s počítačem funguje v těsně provázané zpětnovazební smyčce. Uživatel se v AR kontinuálně pohybuje a systém v reakci na to zaznamenává vstup od uživatele a následně mu zobrazuje vizualizaci, která je propojená s objekty v reálném světě. Tedy pro kompletní AR systém je potřeba mít alespoň tři komponenty: sledovací (tracking), propojovací (registration) a vizualizační. Čtvrtá komponenta - prostorový model - obsahuje model reálného světa a model virtuálního světa. Sledovací komponenta využívá model reálného světa pro určení polohy uživatele v reálném světě. Virtuální se skládá z obsahu, který je využitý pro rozšíření. [18]

Typickým příkladem AR je mobilní aplikace, která snímá okolí prostřednictvím kamery, rozpozná ve videu marker (terč, obrázek) a sleduje změnu jeho polohy v reálném čase. Podle rozpoznané polohy markeru na něj aplikace ve videu umístí virtuální 3D předmět a výsledné video se zobrazí na displeji. Uživateli se tak naskytne iluze, že předmět existuje v reálném světě. Na obrázku 1.1 vidíme virtuální konvici postavenou na markeru.



Obrázek 1.1: Příklad AR, která využívá marker (obrázek převzat z [1])

Druhý typ ukotvení, který je pro nás důležitý, je k GPS souřadnicím. Tento

typ AR je určen pro prohlížení venku, protože pomocí GPS dokážeme určit polohu zařízení s přesností na jednotky metrů. To by mohlo být v budoucnu zlepšeno s využitím navigačního systému Galileo, který má nabídnout přesnost určení polohy do jednoho metru. Na obrázku 1.2 vidíme AR, která za pomoci GPS souřadnic a orientace telefonu zobrazuje virtuální obsah zasazený do prostoru.



Obrázek 1.2: Příklad AR, která využívá GPS (obrázek převzat z [2])

Kapitola 2

Analýza

V této kapitole si uvedeme požadavky na naši aplikaci, které plynou ze zadání a z průběžných konzultací. Dále prozkoumáme některá již existující řešení a pro nás nejdůležitější X3D uzly. V kapitole 3 si představíme náš návrh řešení a všechna uvedená řešení porovnáme.

2.1 Požadavky

Scéna obsahuje AR objekty a je ukotvena k reálnému světu. Scénář je soubor scén. Používáme dva typy ukotvení AR, pomocí markeru nebo pomocí GPS souřadnic. S oběma typy jsme se již seznámili v sekci 1.1. Událost v mobilní aplikaci pro prohlížení AR obsahu představuje soubor podmínek a akcí, kde se všechny akce události vykonají po splnění všech podmínek události. Požadavky pro náš portál jsou následující:

- Aplikace bude umět načíst scénář, uživatel ho bude moci upravit a uložit na server. Následně ho půjde stáhnout a prohlížet v mobilní aplikaci.
- Scénář bude obsahovat scény, které budou moci být ukotvené pomocí GPS nebo markeru. Do scénáře půjde vkládat modely ve formátu X3D, aplikace bude automaticky detekovat animace a podčásti, které v modelu mohou být definované.
- Aplikace bude umožňovat tvorbu událostí a jejich provázání s různými objekty ve scénáři.
- V aplikaci půjde vytvářet anotace, tedy přiřadit určitý text k modelu, části modelu nebo k místu.
- Budou prováděny testy s uživateli a výsledky se při vývoji budou zohledňovat.

- Aplikace navazuje na další části projektu, kterými jsou mobilní aplikace pro prohlížení AR obsahu, webový portál pro tvorbu výukových úkolů. Také bude komunikovat se serverem, na který se budou nahrávat a ukládat data (AR scény, videa, obrázky, modely, textury).

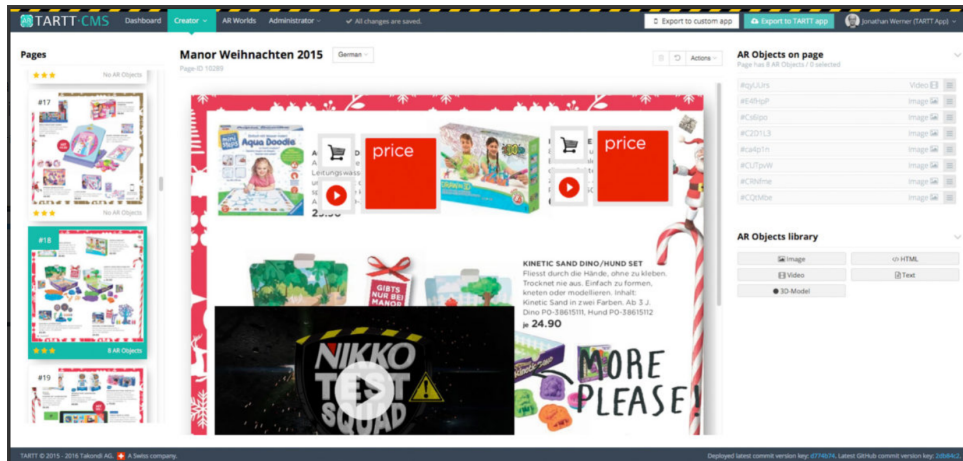
2.2 Existující řešení

Řešení, která se zabývají podobným tématem, jsou například Augment [8] a Augmara [9] (dříve TARTT [10]). Obě služby zahrnují webovou aplikaci pro správu a editaci AR obsahu a mobilní aplikaci pro jeho prohlížení.

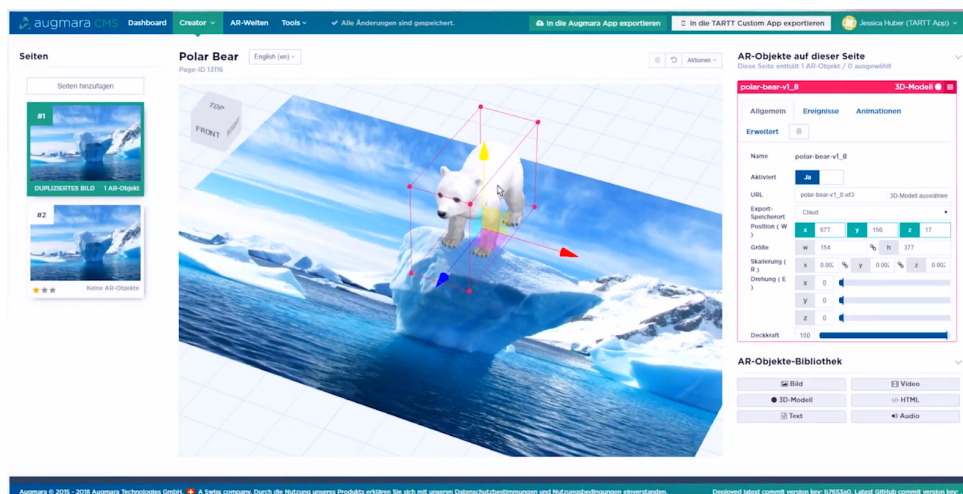
Augmara primárně umožňuje přidávat AR obsah do různých katalogů a propagačních materiálů. Lze přidávat AR objekty jako obrázek, 3D model, text, video, audio. Služba také umožňuje používat události a animace. Služba tedy pracuje jenom s AR ukotvenou k nějakému markeru, ten je vytvořen pomocí nahraného obrázku nebo pdf souboru. Modely lze transformovat jak pomocí polí s konkrétními hodnotami transformací, tak pomocí různých pomocných prvků v zobrazené 3D scéně. Okno aplikace je rozděleno do tří hlavních částí. První částí je seznam stránek (markerů), na které se přidává AR obsah. Druhou částí, největší, je aktuální 3D pohled na scénu. V třetí části je výběr AR objektů, které lze přidat, a přehled objektů, které se aktuálně ve scéně nachází. Když je nějaký konkrétní objekt vybrán, zobrazí se vlastnosti tohoto objektu. Ty lze i přímo upravovat. Aplikace nabízí také možnost zrušit poslední akci a k dispozici je i volba jazyka. Na liště u horního okraje obrazovky je vidět jméno profilu uživatele a vedle něj se nachází tlačítko pro export obsahu do mobilní aplikace. Tvorba obsahu pomocí této aplikace je intuitivní a je do velké míry vzorem pro náš portál.

Augment je určený hlavně pro prezentaci produktů zákazníkům, tedy zobrazování 3D modelů pomocí AR, například si díky němu zákazník může vizualizovat, jak by vypadal kupovaný výrobek u něj doma. Augment nabízí kromě mobilní a webové aplikace také desktopovou aplikaci pro prohlížení a konfiguraci 3D modelů (např. jejich textur a animací). Modely lze prohlížet dvěma způsoby: s markerem (v této službě nazvaný „tracker“) nebo bez. S markerem si lze umístit model na vlastní obrázek. Prohlížení bez markeru je určeno pro větší modely, model se umístí na úroveň podlahy. V obou případech lze následně jednoduše model transformovat pomocí gest, tzn. zvětšovat ho, posouvat a otáčet s ním. Tyto transformace dělá uživatel sám v AR, aby dostal co nejlepší představu o tom, jak bude reálný produkt vypadat. V aplikaci jsou tlačítka pro přemístění modelu tam, kam je aktuálně namířená kamera, a pro vytvoření snímku obrazovky. Ve webové aplikaci lze nahrávat

2.3. X3D uzly



Obrázek 2.1: TARTT - vytváření interaktivního katalogu. (obrázek převzat z [10])



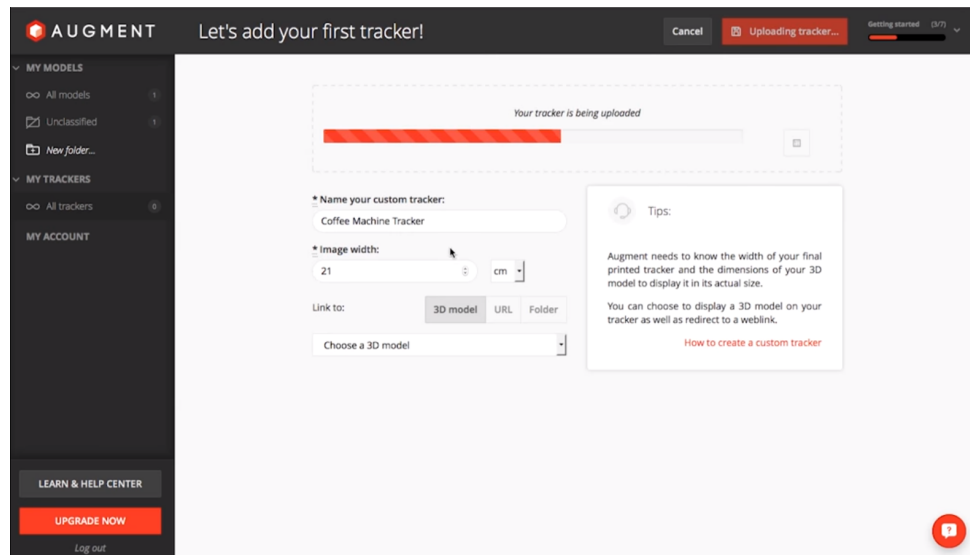
Obrázek 2.2: Augmara - umístování modelu na marker. (obrázek převzat z [11])

markery a 3D modely. Předností služby Augment je její jednoduchost.

2.3 X3D uzly

Naše aplikace má podporovat modely ve formátu X3D, ty jsou jednoduše zobrazitelné na webové stránce pomocí technologie X3DOM. Obě technologie

2. Analýza



Obrázek 2.3: Augment - nahrávání trackeru. (obrázek převzat z [12])



Obrázek 2.4: Augment - umisťování modelu na tracker. (obrázek převzat z [12])

jsou blíže popsány v sekci 4.2. Pro naši aplikaci je důležitý X3D uzel TimeSensor. Uzel s ubíhajícím časem generuje události [23] a na tyto události mohou být v X3D souboru napojené např. interpolace transformací a interpolace

barev. Tedy pomocí uzlu TimeSensor lze vytvářet animace a v naší aplikaci můžeme animace modelu zapnout a vypnout pomocí změny atributu uzlu TimeSensor, který se jmenuje „enabled“. Uzel v X3D může mít atribut id, který tento uzel jednoznačně identifikuje. Dalšími důležitými uzly jsou uzly Transform a MatrixTransform. Za pomocí nich měníme transformace 3D objektů, které uživatel vidí v zobrazené 3D scéně. Uzel Appearance používáme pro zvýraznění vybrané podčásti modelu.

Kapitola 3

Návrh řešení

V této kapitole si uvedeme náš návrh řešení, uživatelské rozhraní a porovnáme naše řešení s již existujícími řešeními ze sekce 2.2).

Základem našeho řešení je strom objektů, jehož kořenem je objekt Scenario. Motivací pro strom objektů byla přehlednost a jednotný přístup k výběru, editaci, vytváření a odstraňování objektů, které dohromady představují vytvářený AR obsah.

Naše aplikace stahuje pomocí API serializovaný scénář ze serveru, deserializuje ho do stromu objektů a ty zobrazuje. Uživatel může mezi těmito objekty procházet, vytvářet nové, mazat stávající a upravovat vlastnosti objektů. Uživatel může nahrávat soubory jako modely, obrázky, videa a textury na server a ve scénáři na ně, kromě textur, vytvářet reference. Textury uživatel nahraje zároveň s 3D modelem na server, tyto textury aplikace před zobrazením modelu automaticky detekuje v souboru modelu a podle jména je najde na serveru. Scénář se uloží pomocí serializace a nahrání na server. Ze serveru bude moci scénář stáhnout i mobilní aplikace a bude předpokládat stejnou strukturu scénáře jako naše aplikace.

Do scénáře můžeme přidávat scény a vytvořit k nim ukotvení. Ukotvení pomocí GPS uživatel vytvoří pomocí dynamické mapy. Na mapě vybere bod a po vytvoření ukotvení se zobrazí 3D reprezentace okolí tohoto bodu včetně terénu a 3D budov. Ukotvení pomocí markeru se provede nahráním obrázku na server a vybráním reference na něj. Objekt Model se vytvoří pomocí nahrání souboru na server a následným vytvořením reference na něj ve scénáři. Aplikace automaticky detekuje části modelu a animace v souboru modelu a vytvoří pomocí nich nové objekty, které jsou potomky objektu Model.

Ve scéně může uživatel vytvořit událost, událost může mít spouštěče události a akce události. Ty odkazují na nějaký již existující objekt ve scénáři a zároveň jsou nějakého typu a mohou mít i parametry. Seznam typů a jejich parametrů

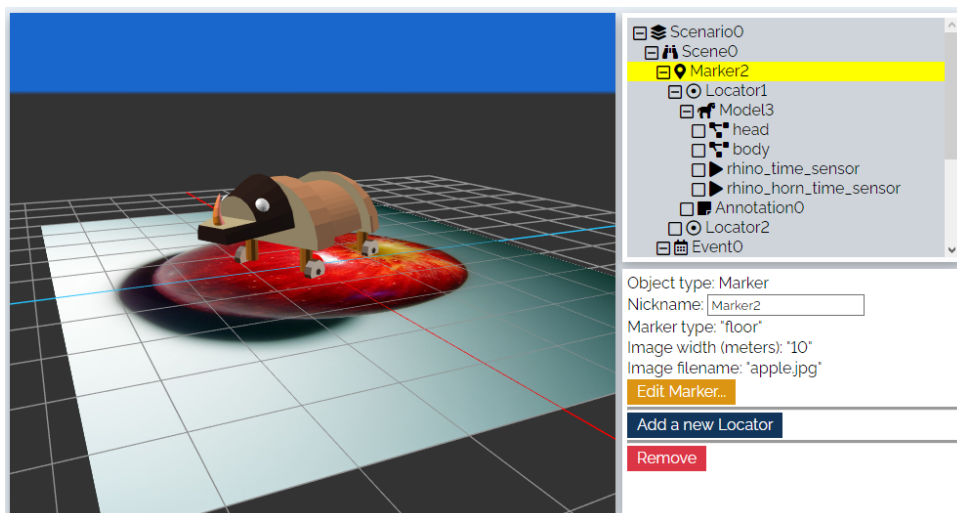
je definovaný v samostatném souboru a lze ho jednoduše rozšířit. V mobilní aplikaci je však rozšíření složitější, protože vývojář bude muset tyto změny systému událostí naimplementovat.

Anotace jsou v aplikaci reprezentované objektem Annotation, lze je připojit k objektu Model, ModelPart nebo Locator. V mobilní aplikaci půjde tyto anotace zobrazit pomocí událostí.

Aplikace je vložena do portálu pro vytváření výukových úkolů a ten s ní komunikuje pomocí zpráv, ve kterých zasílá tokeny nutné pro autentizaci při komunikaci s API serveru.

3.1 Uživatelské rozhraní

Uživatelské rozhraní (na obrázku 3.1) se skládá ze tří sekcí. V sekci **outliner** (vpravo nahoře) je strom objektů a právě vybraný objekt je zvýrazněný. V sekci **properties** (vpravo dole) jsou vlastnosti vybraného objektu (některé z nich lze upravovat) a akce dostupné pro vybraný objekt. V sekci **3D view** (vlevo) je trojrozměrný pohled na scénu, která obsahuje aktuálně vybraný objekt. V ní jsou zobrazeny objekty Marker, GPS, Locator, Model, ModelPart, Animation a 3D prvky pro změnu transformací (gizmo). Jednotlivým částem uživatelského rozhraní se dále věnujeme v kapitole 4.



Obrázek 3.1: Ukázka uživatelského rozhraní. (Obrázek na markeru převzat z [13], model hračky není původní prací autora.)

3.2 Porovnání řešení

Naše řešení je inspirováno službou Augmara, panel pro výběr scény nahrazuje stromem objektů, kde jsou mimo jiné i scény. Augmara umožňuje pouze ukotvení k markeru, naše řešení umožňuje navíc ukotvení k GPS souřadnicím. Transformace 3D objektů lze v obou řešeních měnit jak pomocí manipulačních prvků ve 3D scéně (gizmo), tak pomocí ručního zadání hodnot transformací. Ve službě Augmara se každý objekt vytváří přímo z hlavní obrazovky. Oproti tomu v našem řešení se nové objekty přidávají jako potomci již existujících objektů. Ke každému objektu lze přidat pouze objekty určitých typů.

Kvůli větším vzdálenostem, které se běžně vyskytují při ukotvení pomocí GPS, jsme přidali objekt lokátor, který slouží jako jakýsi „podpočátek“. Poloha jeho potomků je vůči němu relativní a nově vytvořený objekt Model má polohu shodnou s polohou lokátoru. Pokud změníme polohu lokátoru, změní se zároveň poloha všech jeho potomků.

Oproti službě Augmara náš systém podporuje automatickou detekci podčástí modelu a jejich navázání na události. Také podporuje zadávání anotací, tedy textu přiřazeného buď k modelu, k jeho podčásti nebo k lokátoru. Augmara oproti našemu řešení umožňuje vrátit poslední provedenou akci.

Služba Augment je oproti našemu systému daleko jednodušší a umožňuje uživateli si řídit, jak a kde chce AR obsah zobrazovat. V tomto ohledu je náš systém méně flexibilní, ale je to způsobeno tím, že naším cílem je vytvářet komplexnější scény, které jsou ukotvené.

Kapitola 4

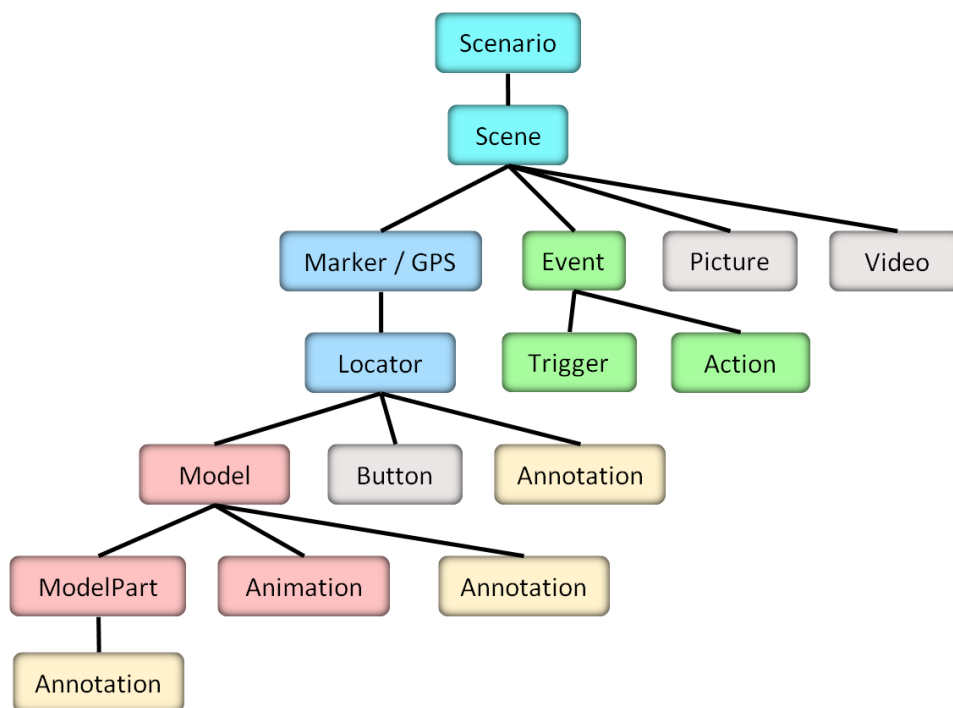
Implementace

Uložení / načtení stromu objektů se provádí pomocí serializace / deserializace stromu objektů a stáhnutí / nahrání na server. Každý objekt má pevně dáno, které typy objektů mohou být vytvořeny jako jeho potomci. Tyto vztahy jsou znázorněny ve schématu 4.1.

4.1 Přehled objektů

Následuje popis jednotlivých objektů, jejich možností a vlastností. Uváděné možnosti objektů jsou možnosti, které má uživatel dostupné v sekci properties pro konkrétní vybraný objekt v sekci outliner. Vlastnosti objektů, které uvádíme, platí pro serializovaný strom objektů scénáře do formátu JSON, který slouží pro nahrání scénáře na server.

- Většina **objektů** má možnost odebrat sám sebe a možnost vytvořit nový objekt jako svého potomka. Pokud konkrétní možnost objektu vyžaduje nějaký další vstup od uživatele, zobrazí se pro ni vyskakovací okno, které od uživatele získá další potřebný vstup. Každý objekt má svoje automaticky generované jméno a uživatelsky editovatelnou přezdívku. Pokud objektu byla zadána přezdívka, zobrazuje se místo jeho automaticky vygenerovaného jména.
- Pokud **neexistuje objekt Scenario**, uživatel má na výběr z možností vytvořit nový scénář, otevřít již existující, správa scénářů. Možnost vytvořit nový scénář vytvoří nový scénář na serveru a ten se načte v aplikaci. Možnost otevřít již existující scénář načte scénář ze serveru. Možnost správa scénářů umožňuje přejmenovávat a mazat scénáře.
- U objektu **Scenario** jsou na výběr možnosti vytvořit scénu, uložit scénář na server, smazat scénář na serveru, správa souborů na serveru, zavřít



Obrázek 4.1: Objektové schéma scénáře. Typy objektů jsou pro snadnější orientaci odlišeny barevně. Světle modré uzly mají pouze organizační význam, tmavě modré mají navíc svoji reprezentaci v 3D pohledu na scénu. Červené uzly jsou 3D modely, jejich části a animace. Žlutě jsou zvýrazněné anotace, zeleně jsou události a jejich součásti. Šedě jsou podbarvené uzly, které mají význam jen ve vztahu k událostem.

scénář. Možnost vytvořit scénu vytvoří novou scénu jako potomka tohoto scénáře. Pomocí možnosti správa souborů se dají mazat soubory (náležící otevřenému scénáři) na serveru. Možnost zavřít scénář lokálně smaže objekt scénář a vrátí stav aplikace do stavu, kdy nebyl vytvořen žádný objekt. Objekt Scenario si uchovává ve vlastnosti `namingCounters` pro každý typ objektu informaci o posledním použitém id při pojmenování a po vytvoření objektu daného typu id inkrementuje.

- Scénář může obsahovat **scény**, mezi kterými se přepíná v sekci outliner. Obsah scén je navzájem oddělený, v sekci outliner jich může být zobrazeno více, ale v 3D view je zobrazena maximálně jedna. Možnosti scény jsou přidat ukotvení, událost, obrázek, video a odebrat scénu. Každá scéna může mít maximálně jedno ukotvení, pokud již ukotvení dané scény

existuje, možnost vytvoření scény se v této scéně zakáže.

- Scénu lze **ukotvit** pomocí GPS souřadnic a pomocí markeru. V případě ukotvení pomocí GPS se od uživatele pomocí mapové aplikace získají GPS souřadnice. Trojrozměrná reprezentace okolí těchto souřadnic (terén, budovy, satelitní snímek) se následně zobrazí ve 3D view. V případě ukotvení pomocí markeru uživatel zvolí typ jeho orientace (na zdi, na stropě, na zemi), šířku v metrech v reálném světě a obrázek markeru. Následně se ve 3D view zobrazí plocha se zvoleným obrázkem. Střed ukotvení je v počátku souřadnicového systému. Objekty GPS a Marker mají možnost editovat (ta uživatele znovu provede oknem vytvoření ukotvení, uživatel zde může změnit vlastnosti ukotvení), přidat lokátor, odebrat ukotvení. Ukotvení se ve scéně nachází v počátku souřadnicového systému. Objekt Marker má vlastnosti orientace, název souboru obrázku, id souboru na serveru, šířka markeru. Objekt GPS má vlastnosti GPS souřadnice a elevace.
- **Lokátor** má ve scéně relativní polohu vůči ukotvení. Ve 3D view je reprezentovaný pomocí červené koule, která má stálou velikost vůči kameře. Jeho polohu lze měnit jak pomocí gizma v 3D view, tak pomocí manuální úpravy hodnot v odpovídajících polích v properties, tak pomocí tažení myši za tato pole. Lokátor má možnosti resetovat transformace, vytvořit objekty anotace, tlačítko, model a možnost odebrat lokátor. Objekt Locator má vlastnost `translace`.
- Objekt **Model** má přiřazený 3D model, který se v rámci scény zobrazuje v 3D view. Model může také obsahovat části modelu a animace, tyto jsou automaticky detekovány a s vytvořením objektu model jsou vytvořené i objekty odpovídající detekovaným animacím a částem. Modelu lze upravit transformace posunutí, velikost a otočení pomocí gizma a polí v properties. Jeho možnostmi jsou resetování transformací, vytvoření objektu anotace a odebrání. Model má vlastnosti `translace`, `rotace`, `škálování`, název souboru modelu, id souboru na serveru.
- U objektu **ModelPart** lze vytvořit anotaci. Nemá možnost odebrat, odebere se až s odebráním rodičovského modelu. Po vybrání v sekci `outliner` se za pomoci shaderu zvýrazní odpovídající část modelu v sekci 3D view. ModelPart má vlastnost `x3dId`, která obsahuje id odkazované části souboru modelu.
- Objekt **Animation** umožňuje přímo ve scéně spustit a zastavit odpovídající animaci modelu. Také nemá přímo možnost odebrání. Animation

obsahuje vlastnost `x3dId`, která obsahuje id odkazovaného uzlu `timeSensor` v souboru modelu.

- Objekt **Annotation** obsahuje uživatelem zadaný text. Má možnosti text upravit a odebrat objekt anotace. Annotation má vlastnost `obsah`, která obsahuje text anotace.
- Objekt **Event** má možnosti vytvořit objekt `Trigger`, `Action` a odebrat tento objekt. V mobilní aplikaci se událost aktivuje, pokud jsou současně splněny podmínky všech spouštěčů v této události. Po aktivaci se provedou všechny akce obsažené v události.
- Objekt **Trigger** je určitého typu. Ke každému typu náleží seznam parametrů. Spouštěč má „cílový objekt“, kterým je některý z již existujících objektů ve stromu objektů.

Soubor ve formátu JSON definuje možné typy spouštěčů, jim odpovídající povolené typy cílových objektů, seznam názvů parametrů a jejich výchozí hodnoty. Tento soubor je lehce rozšiřitelný, aplikace ho zpracovává automaticky.

Spouštěč má možnosti editovat spouštěč a odebrat. Pokud je některý objekt ve stromě zároveň cílovým objektem některého spouštěče či akce, nelze ho odstranit, dokud je cílovým objektem. Trigger obsahuje vlastnosti `triggerType`, `parameters` (seznam, který obsahuje střídavě vždy název parametru a jeho hodnotu), `targetName` (jméno cílového objektu).

- Objekt **Action** funguje obdobně jako spouštěč. Má obdobné vlastnosti.
- Objekt **Button** se v mobilní aplikaci zobrazí jako překrytí obsahu obrazovky, pokud bude uživatel v jeho blízkosti. Slouží jako cílový objekt spouštěče události. Má možnost odebrat.
- Objekty **Picture** a **Video** obsahují odkaz na soubor obrázek resp. video na serveru, který se po vyvolání akcí události zobrazí v okně v mobilní aplikaci. Má možnost odebrat. `Picture` a `Video` mají vlastnosti `assetId` (id souboru na serveru) a název souboru.

4.2 Použité technologie

Naše aplikace používá technologie X3D, X3DOM, JSON a standardní webové technologie HTML, JavaScript, CSS. Dále OpenStreetMap a Google Maps API. Pro komunikaci se serverem projektu EduARd používá REST API.

- **X3D** je otevřený formát pro reprezentaci 3D scén a objektů. X3D byl původně vyvíjen jako VRML (modelovací jazyk pro virtuální realitu), avšak dnes již dospěl do fáze, kdy je ISO X3D standardem. [3]
- **X3DOM** je framework a runtime pro 3D grafiku na webu, integruje HTML5 a 3D obsah a dovoluje zahrnout X3D elementy jako součást HTML5 DOM¹ stromu. [4] Byl vyvinut, protože chybělo jednoduché řešení, které by bez pomoci pluginů umožňovalo zobrazovat 3D grafiku na webu. Svůj vzor má v integraci SVG (formát pro vektorovou grafiku) přímo do HTML. Využívá již existující webové technologie např. pro skriptování, tedy neuvádí technologie nové, nýbrž používá ty stávající. Popisuje, jak může vypadat propojení X3D a HTML5, obě tyto specifikace na sebe sice navzájem odkazovaly, ale model pro jejich integraci stále chyběl. X3DOM model nerenderuje obsah DOMu přímo, ale automaticky synchronizuje a aktualizuje X3D strom. [5]

X3DOM je experimentální technologie, občas se chování výrazně mění mezi verzemi, dokumentace je stručná, informací o X3DOM je na internetu málo, některé informace jsou zaměřené pouze na starší jazyk VRML, i tak mohou být užitečné.

- **JSON** neboli JavaScript Object Notation je souborový formát nezávislý na počítačové platformě, jehož výhodou podle Lanham (2017) [20] je, že je dobře srozumitelný a zabírá málo místa.
- **HTML** neboli Hypertext Markup Language je standardní jazyk pro vytváření webových stránek a aplikací.
- **JavaScript** je programovací jazyk, který umožňuje interaktivitu webových stránek.
- **CSS** neboli Cascading Style Sheets je jazyk pro popis způsobu zobrazení HTML dokumentu.
- **SASS** je jazyk, který rozšiřuje syntaxi CSS a lze ho zkompilovat do CSS.
- **OpenStreetMap** [6] je projekt, který vytváří a distribuuje svobodná geografická data pro celý svět.
- **Google Maps API** [7] je služba, pomocí které mohou vývojáři integrovat Google Maps do své webové stránky. Od společnosti Google používáme Static Maps API, Elevation Service a Maps JavaScript API.

¹DOM je API, které vidí HTML dokument jako strom, kde každý HTML uzel je objektem. Metody DOMu umožňují programátorský přístup ke stromu.

- **REST API** je architektura rozhraní navržená pro distribuované prostředí. REST implementuje čtyři základní metody, které jsou známé pod označením CRUD, tedy vytvoření dat (Create), získání požadovaných dat (Retrieve), změnu (Update) a smazání (Delete). Tyto metody jsou implementovány pomocí odpovídajících metod HTTP protokolu. [22]

4.3 Implementační detaily

Sekce outliner. Outliner reprezentuje aktuální stav stromu objektů a pokud dojde ke změně stromu, outliner překreslíme. Každý řádek sekce outliner reprezentuje objekt. V sekci outliner se dá přesouvat mezi jednotlivými prvky jak myší, tak klávesnicí. Pro pohyb klávesnicí je potřeba, aby byl outliner zaostřen (focused). Jednotlivé prvky sekce outliner se dají rozvinout nebo zabalit, tím se zobrazí nebo skryjí jejich potomci. Rozvinutí nebo zabalení prvku lze provést kliknutím myši na příslušné tlačítko nebo šipkami doprava / doleva na klávesnici. Klávesy se šipkou nahoru / dolu slouží pro navigaci mezi viditelnými prvky sekce outliner. Každý objektový typ má přiřazen svůj vlastní symbol ze sady ikon Font Awesome [14], který je dostupný zdarma v omezené verzi. Na některé objekty (Locator, GPS, Marker, Model, ModelPart) se po dvojitém kliknutí myši na jejich položku v sekci outliner zaměří kamera v 3D view.

Sekce properties. Sekce je rozdělena vizuálně na tři části. První část obsahuje výpis vlastností objektu, z nichž některé mohou být editovatelné, a tlačítka akcí, kterými se dají tyto vlastnosti měnit. Druhá část obsahuje akce, kterými se tvoří nové objekty a ve třetí části je většinou možnost odebrat (tuto možnost nemají objekty Scenario, ModelPart, Animation). Odebrání objektu odebere rekurzivně i všechny jeho potomky.

Sekce 3D view. V sekci 3D view se pomocí X3DOM zobrazuje 3D obsah. Toho je docíleno tak, že sekce 3D view obsahuje uzel x3d, v jehož stromu přidáváme, odebíráme nebo upravujeme X3DOM uzly. V sekci je trvale viditelná mřížka, kde jsou barevně naznačené osy x, z. Mřížka se nachází v počátku souřadnicového systému. 3D obsahem lze otáčet pomocí levého tlačítka myši, přibližovat pohled lze pomocí kolečka nebo pravého tlačítka myši. Dvojitě kliknutí myši změní aktuální střed otáčení kamery. Využíváme navigační mód turntable, protože pro naše účely bylo jeho ovládání nejpřirozenější. Po kliknutí na model se vybere odpovídající objekt Model v sekci outliner.

Přibližování kamery jsme změnili tak, že jeho rychlost je závislá na vzdálenosti kamery od středu otáčení kamery. To funguje celkem dobře, ale je

velký rozdíl v rychlosti přibližování v závislosti na tom, kolik 3D obsahu je zobrazováno v 3D view. Je možné, že by tento problém vymizel s použitím výkonnějšího PC.

Vyskakovací okno. Při zobrazeném vyskakovacím okně se ztmaví ostatní obsah stránky a lze interagovat pouze s obsahem okna.

Serializace stromu objektů. Strom se serializuje do souboru ve formátu JSON. Každý objekt má svoji funkci pro serializaci a deserializaci sebe sama. Funkce voláme pro objekt Scenario a volání poté proběhne rekurzivně pro všechny objekty ve stromě. Serializovaný objekt má většinou podobné atributy jako neserializovaný objekt, mohou se lišit názvy. Následuje příklad serializovaného objektu Marker, který představuje půl metru široký plakát jablka umístěný na stěně.

```
{
  "objectType": "Marker",
  "nickname": "apple marker",
  "name": "Marker0",
  "markerOrientation": "wall",
  "imageSrc": "apple.jpg",
  "width": 0.5,
  "_comment": "apple poster"
  "locators": [
    ...
  ]
}
```

Zápis všech objektů vypadá podobně, v kompletním souboru by tento objekt marker obsahoval ještě objekty Locator, které jsou zde vynechány, a byl by zanořen do objektu Scene, který by byl zanořen do objektu Scenario. (Vlastnost `_comment` byla zamýšlena jako možnost pro tvůrce AR obsahu okomentovat objekt. Vlastnost ale zatím není nijak využita.) V objektu Scenario je navíc uchováván seznam posledních id, který je potřeba pro automatické pojmenovávání nových objektů. Příklad kompletního serializovaného scénáře je v příloze A.

Model. Před nahráním souboru modelu na server z něj aplikace odstraní nechtěné uzly (světla, pozadí, skripty, pohledy, navigationInfo, senzory). Adresy textur se nahradí názvy souborů textur. Po vytvoření objektu Model se stáhne odpovídající x3d soubor ze serveru a názvy souborů textur se nahradí odkazy na textury na serveru.

Při nahrávání modelu s texturami aplikace tyto textury v souboru detekuje a zobrazí uživateli seznam názvů detekovaných textur a u nich možnost nahrání jednotlivých textur na server.

Animace. V souboru modelu se automaticky detekují animace, těm odpovídají X3D uzly `timeSensor`, a vytvoří se pro ně nové objekty `Animation`. Objektu `Animation` se nastaví přezdívka stejná, jako je id jejího uzlu `timeSensor`. Animaci lze spustit a zastavit, to se vnitřně dělá pomocí upravení uzlu `timeSensor`, který jsme již zmiňovali v sekci 2.3.

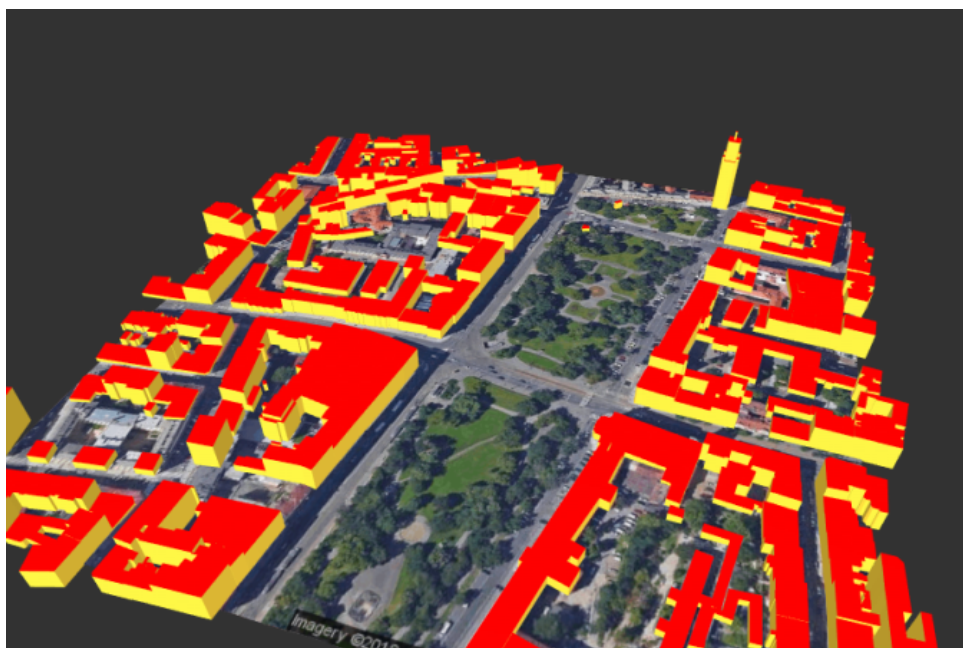
Část modelu. Po přidání Modelu se automaticky v přidávaném X3D souboru najdou uzly, jejichž id začíná předponou „`modelPart_`“, a vytvoří se podle nich objekty `ModelPart`. Objekty dostanou přezdívku podle id odpovídajícího X3D uzlu (kromě předpony). Objekty mají využití jako cíle objektů `Trigger` a `Action`. Pokud je objekt `ModelPart` vybraný v sekci `outliner`, v 3D view je jemu odpovídající část modelu zvýrazněná pomocí shaderu.

Ukotvení scény pomocí GPS. Nejdříve získáme od uživatele pomocí Google Maps API [7] GPS souřadnice. Uživatel interaguje s dynamickou mapou a na ní vybírá bod. K dispozici má i pole pro vyhledání místa na mapě. Zároveň vidí GPS souřadnice vybraného bodu.

Potom získáme satelitní statickou mapu okolí bodu z Google Maps Static API [15], protože podle Lanham (2017) [21] pro podobný účel funguje dobře. Mapu okolí získáváme ve výchozím nastavení při úrovni přiblížení Google Maps (zoom level) nastavené na hodnotu 17. Úroveň přiblížení lze snadno programátorsky změnit. Tato úroveň byla vybrána proto, že dává uživateli alespoň nějaký pojem o okolí a zároveň satelitní mapa není příliš rozmazaná. Tato úroveň přiblížení a maximální rozlišení (640x640 px) statické mapy definují rozměry výřezu světa, který uživateli zobrazíme. Rozměry výřezu světa v zeměpisné šířce a délce získáme pomocí přepočtu z pixelů statické mapy [16].

Dále si vytvoříme seznam rovnoměrně do mřížky rozmístěných bodů, které se nacházejí v našem výřezu světa. Zjistíme elevaci všech bodů ze seznamu pomocí `Elevation Service` [17]. Do nového X3DOM uzlu `ElevationGrid` vložíme údaje o elevaci a texturu statické mapy, tím získáme model terénu. Z `OpenStreetMaps` [6] získáme `.osm` soubor popisující náš výřez, v něm najdeme polygony popisující půdorysy budov a bud z hodnoty výška budovy nebo z počtu pater odhadneme výšky budov. Pomocí X3DOM uzlů `Extrusion` a `IndexedFaceSet` vytvoříme stěny a střechy budov. Pro každou budovu najdeme bod jejího půdorysu, který je po promítnutí na model terénu nejnižší. Tento bod určí, v jaké výšce bude budova umístěna.

Na obrázku 4.2 (satelitní mapa pochází z [15], modely byly vytvořeny podle dat z [6]) můžeme vidět příklad 3D okolí GPS bodu.



Obrázek 4.2: Trojrozměrná reprezentace okolí GPS bodu (Karlovo náměstí)

Ukotvení scény pomocí markeru. Zde využíváme X3DOM uzel `Plane`, kterému nastavíme šířku v metrech (kterou uživatel zvolil pro marker) a uživatelem poskytnutý obrázek. Marker může mít jednu ze tří poloh, na zemi, na stěně nebo na stropě. Poloha markeru je pouze pro pohodlí uživatele při editaci, při výsledném prohlížení v mobilní aplikaci nehraje roli. Ze zadní strany markeru jsme přidali druhý uzel `Plane`, který je vyplněn jednou barvou.

Transformace pomocí gizma. Gizmo je vlastně 3D model, který obsahuje šipky pro translační pohyb, prstence pro rotační pohyb a kvádry pro zvětšování. Díky sensorům v X3D souboru gizma lze jeho transformace měnit pomocí chycení a táhnutí myši. Po vybrání objektu `Model` nebo objektu `Locator` se gizmo ve 3D view zviditelní a jeho proměnné s maticemi transformací se nastaví na ty samé, jako má právě vybraný objekt. Při manipulaci se senzory gizma se mění zároveň transformace gizma a to mění transformace vybraného modelu. Pokud je vybraný objekt lokátor, z gizma jsou viditelné pouze šipky pro translační pohyb, součásti potřebné pro jiné transformace se schovají.

Transformace pomocí pole DragInput. DragInput je v základu HTML uzel input, který je rozšířen o možnost chycení a tažení myší pro změnu jeho hodnoty. Změna hodnoty pole DragInput mění transformaci gizma ve 3D view a to mění transformaci vybraného objektu. Stisknutím klávesy Shift lze zpomalit krok změny hodnoty pole DragInput. Do tohoto pole lze také po kliknutí myší manuálně zadávat hodnoty.

Barvy tlačítek. Tlačítka v aplikaci mají pro přehlednost různé barvy. Modrá se používá pro hlavní akci, oranžová pro vedlejší / pokročilou akci, šedá pro zavření, zelená pro uložení, červená pro odstranění. Styly aplikace byly inspirovány styly v portálu pro vytváření edukačních úloh, do kterého je naše aplikace vložena.

Kapitola 5

Ukázka aplikace

V této kapitole si ukážeme příklad použití aplikace. Používání aplikace je ve své podstatě jednoduché. Jde o sestavování stromu scénáře pomocí přidávání nových objektů. Při vytváření objektu je potřeba zvolit hodnoty jeho vlastností nebo nějaký potřebný soubor (ten je potřeba volit pro objekty představující modely, obrázky nebo videa). V případě vytváření spouštěče nebo akce události musí uživatel navíc zvolit nějaký cílový soubor ze stromu scénáře.

Již jsme viděli, jak vypadá jednoduchá scéna ukotvená pomocí markeru (obrázek 3.1) a také jak vypadá zobrazení terénu a budov v okolí GPS bodu v případě ukotvení scény pomocí GPS (obrázek 4.2).

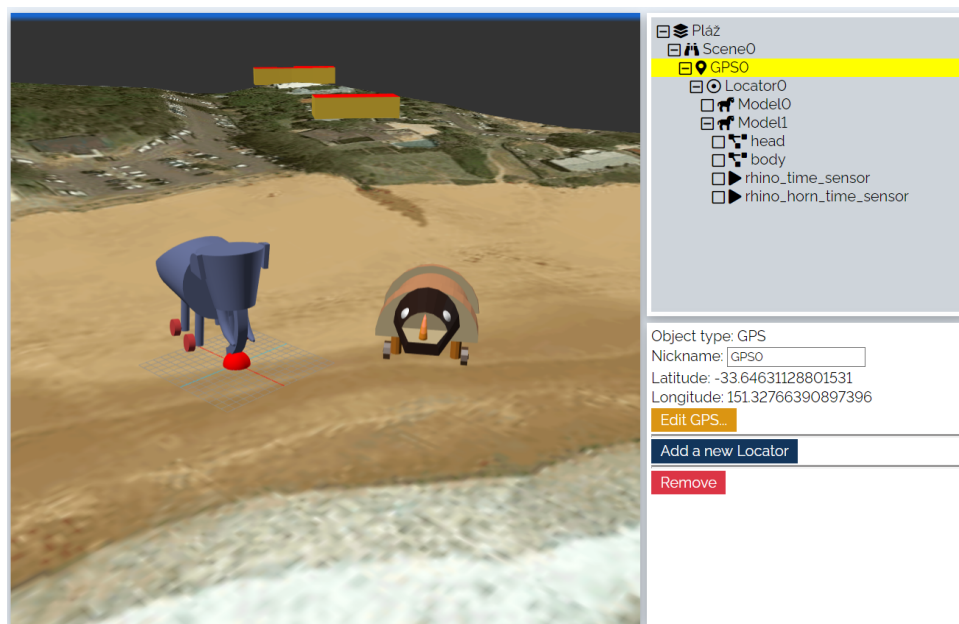
Nyní si popíšeme, jak vytvořit jednoduchý scénář s událostí. Nejdříve vytvoříme nový objekt Scene, k němu pro ukotvení připojíme objekt GPS (jeho GPS bod vybereme pomocí dynamické mapy, která se nám zobrazí v okně pro vytvoření GPS ukotvení), k němu připojíme Locator, a k němu dva objekty Model. Při vytváření objektu Model vybereme X3D soubor, který je nahráný na serveru (pokud není nahráný, můžeme ho ve stejném okně nahrát). Výsledek je na obrázku 5.1.

Dále přidáme objekt Event. Na obrázku 5.2 přidáváme spouštěč události, který událost aktivuje, pokud bude uživatel ve vzdálenosti do 5 metrů od modelu nosorožce. Šedě jsou zbarvené objekty, které nemůžeme přidat jako cílový objekt spouštěče, protože jejich typ není kompatibilní s vybraným typem spouštěče „Is in proximity“.

Na obrázku 5.3 přidáváme akci události, která zapne animaci nosorožce, jakmile bude událost aktivována.

Vytvořený scénář (a scénář z přílohy) je možné si prohlédnout po přihlášení ukázkovým účtem na produkčním serveru projektu EduARd:

5. Ukázka aplikace



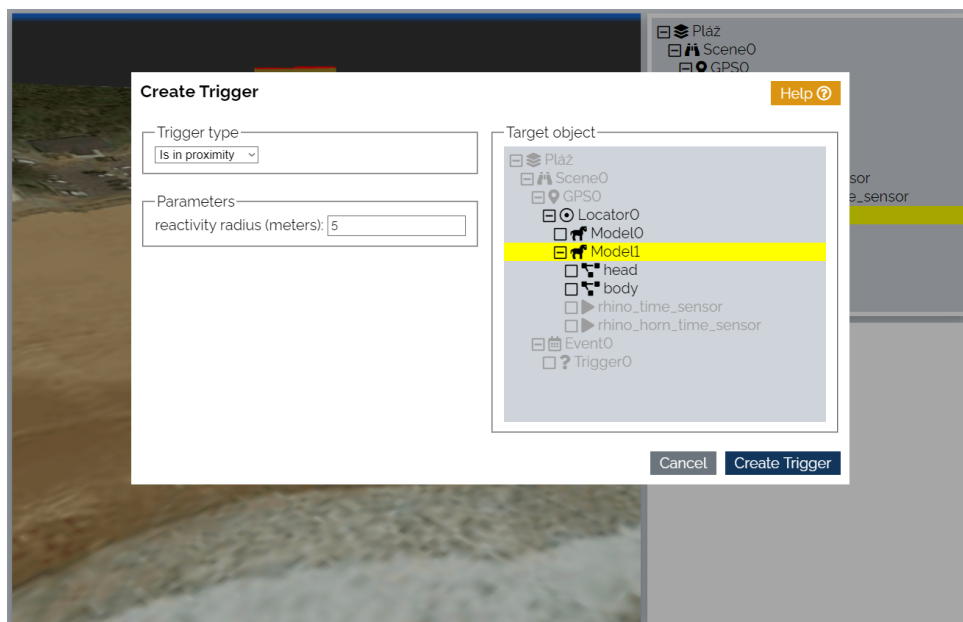
Obrázek 5.1: Modely hraček umístěné ve scéně, která je ukotvená pomocí GPS souřadnic. (Modely hraček nejsou původní prací autora.)

url: <https://editor.eduard.fel.cvut.cz/>

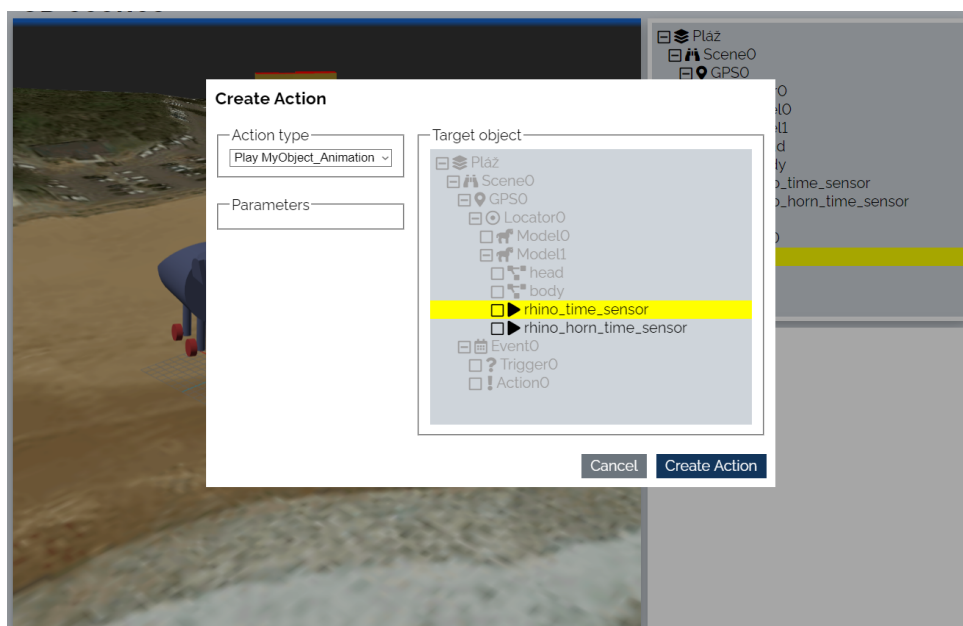
přihlašovací jméno: mrazmic7.ukazkovyuceteduard@email.cz

heslo: \$7_K9ccH

Po přihlášení vidíme portál pro tvorbu edukačních úkolů, na který naše aplikace navazuje. Naše aplikace je vložena do tohoto portálu pomocí HTML uzlu iframe a zobrazí se po kliknutí na tlačítko „3D scény“ na levém panelu. Občas je nutné se odhlásit a přihlásit, aktualizace stránky zatím nevrací uživatele na přihlašovací obrazovku, pokud už přestal platit jeho autentizační token (nutný pro komunikaci aplikace se serverem). V aplikaci je možné si prohlížet připravené scénáře a případně si vytvořit vlastní. Pokud bychom již měli otevřený nějaký scénář, je možné nový scénář vytvořit po zavření otevřeného scénáře.



Obrázek 5.2: Přidávání spouštěče události



Obrázek 5.3: Přidávání akce události

Kapitola 6

Testování

Testovalo se celkem s pěti uživateli. Dokument, který byl předložen účastníkům testování je v příloze B. Obsahuje stručný úvod, ke kterému byli uživatelé ústně seznámeni s AR, dva úkoly s obecnými instrukcemi, aby nebyla příliš odhalena objektová struktura scénáře již v zadání (aby se zjistilo, jak je pro uživatele tato struktura intuitivní), a v závěru je několik otázek, které mají uživatele podnítit k poskytnutí užitečné zpětné vazby. Součástí zpětné vazby bylo i pozorování chování uživatele během testování. Dále jsou v dokumentu přiloženy dva obrázky, které uživateli ukazují, jak by měl vypadat výsledek jeho snažení.

6.1 Výsledky testování

Lokátor. Objekt Locator je v jeho nynější podobě celkem neintuitivní, uživatelé si ho nespojují s přidáním modelu, navíc jeho role splývá s rolí ukotvení scény (dokonce s GPS souřadnicemi bodu, který byl vybrán na dynamické mapě při tvorbě ukotvení scény). O roli lokátoru nedostali uživatelé od autora předem žádnou informaci, jedná se tedy o jejich intuitivní poznatky při setkání s ním. Po seznámení některých uživatelů s rolí lokátoru uznali, že dává smysl. Samotný název lokátor může znít nesympaticky až zarádívě.

S problémem by mohlo pomoci automatické přidání jednoho lokátoru zároveň s přidáním ukotvení, tím by se uživatel s lokátorem setkal a viděl by, jaké možnosti má k dispozici. Měli bychom přehodnotit důvody, proč lokátor existuje a jestli by nemohl být nahrazen jinými mechanikami. Model by se například po vytvoření mohl přidat na místo, na které uživatel klikl ve scéně. Také bychom mohli přidat možnost přidání modelu přímo jako potomka ukotvení. V tom případě by bylo vhodné přidat i možnost přesunout model v objektovém stromě na jiné místo. To by šlo udělat pomocí mechaniky „drag

and drop“ nebo pomocí kopírování a vkládání.

Jeden uživatel by chtěl moci přesouvat lokátor pomocí „drag and drop“, kde by se lokátor automaticky chytil terénu, markeru nebo plochy $y=0$ na místě, kam by ukazovala myš.

Ukotvení. Ukotvení by mohlo převzít některé schopnosti lokátoru, například by mohlo být cílem spouštěče události s tím, že poloha ukotvení by byla v počátku.

Jednotky. Uvádění jednotek by mělo být pravidlem a použití stupňů by bylo vhodnější než použití radiánů. Rotace nově využívají stupně a jednotky uvádíme.

Model. Po přidání modelu by měl být dobře vidět v 3D okně, neměl by být tak malý, aby ho uživatel neviděl. Gizmo nově mění velikost podle vzdálenosti od kamery, to by mělo s problémem pomoci.

Osy. Dva uživatelé zmínili, že by očekávali, že osa Z bude směřovat do nebe. Osy by podle nich také měly být znázorněny přímo v 3D view, možná mít i zobrazené jméno.

Gizmo. Gizmo by mělo být tak velké, aby s ním šlo manipulovat. Gizmo se při testování skrývalo za objekty. Gizmo nově mění svoji velikost podle vzdálenosti kamery.

Outliner. Jeden uživatel považoval prázdnou verzi tlačítka pro rozbalení / zabalení prvku v sekci outliner za zaškrtačací políčko. Jeden uživatel zkusil klikat pravým tlačítkem myši na prvek v sekci outliner pravděpodobně v očekávání nějaké kontextuální nabídky. Zde bychom mohli naimplementovat již zmíněné možnosti kopírovat a vložit.

DragInput. Některé uživatele nenapadlo, že za pole DragInput lze zatáhnout myší. Napadlo to toho, kdo má zkušenosti např. s 3D editačním softwarem.

Způsob změny hodnot tažením myši může být pro někoho zvláštní, ale pro tento případ je tady možnost zadávání hodnoty pomocí klávesnice.

Tlačítko. Ke tlačítku by se měla přidat vlastnost pro určení nápisu, který se v mobilní aplikaci zobrazí na tlačítku. Tlačítko by se mělo dát vytvořit i z okna pro vytvoření spouštěče události.

Tlačítko nyní není intuitivní asi proto, že nemá skoro žádné možnosti a kromě položky v sekci outliner a základních vlastností se ani nic ohledně něj nezobrazuje.

Mohli bychom vytvořit nápovědu, která by přímo v aplikaci vysvětlila účel objektu Button.

Jeden uživatel by u tlačítka očekával možnost vytvořit spouštěč. Mohli bychom tedy k tlačítku přidat možnost pro vytvoření spouštěče, který by měl toto tlačítko nastaveno jako cílový objekt.

Událost. Systém událostí, spouštěčů a akcí vypadá, že funguje dobře. I když je někdy zprvu neintuitivní, po chvíli na to uživatel přijde a naučí se to. Systém by měl zůstat zachován, měl by se přidat počet cest, kterými lze některé objekty vytvořit.

Uživatelská kontrola fungování scénáře. Mělo by být umožněno, aby uživatel mohl vytvořený scénář rychle a jednoduše vyzkoušet, aby si potvrdil, že scénář vytvořil správně. Pro zkoušení by také mohly být k dispozici možnosti jako například simulace GPS polohy. To je ovšem spíše úkol pro mobilní aplikaci.

X3DOM. Mělo by být nějakým způsobem zamezeno tomu, aby mohl uživatel ovládat prohlížení scény všemi způsoby, které X3DOM nabízí. To znamená ovládání klávesnicí, různé klávesové modifikátory při manipulaci myši a také ovládání prostředním tlačítkem myši.

Scéna. Po vytvoření scény by měl být resetován střed otáčení scény a také pozice kamery, každá scéna by si měla pamatovat svoji poslední pozici kamery. Uživatel u vytváření scény očekával, že již bude moci vybrat GPS souřadnice na mapě.

GPS. Model terénu by měl zespoda alespoň nemít stejnou texturu, jako má ze shora, nyní je to matoucí. K modelu terénu by se například daly přidat stěny a podstava, aby působil jako reálný „výřez“ terénu.

Rotace. Skládání rotačních matic v pořadí Z*Y*X nemusí být intuitivní. Jeden uživatel by raději viděl, kdyby např. rotace podle osy x v sekci properties otáčela celou aktuální rotací modelu podle globální osy x.

Hierarchie. Měli bychom prozkoumat, jak zjednodušit hierarchii objektů, prozkoumat možnost, že by více tlačítek pro vytváření objektů bylo zpřístupněno globálně bez výběru nějakého rodičovského objektu. Prvními kandidáty by mohla být tlačítka pro přidání objektů událost, video a obrázek. Viz. snímek obrazovky ze služby Augmara 2.2, kde je takto přístupných šest objektů.

Mapy. Při testování byl problém s načtením bodu na dynamické mapě a také s načtením statické mapy. Tato situace by měla být ošetřena.

Trigger. Spouštěč nejde vytvořit, pokud nemá vybraný cílový objekt. Pokud žádný vhodný objekt neexistuje, musí uživatel odejít z okna vytváření spouštěče a nejdříve vytvořit vhodný objekt. To by se dalo vyřešit přidáním možnosti vytvoření cílového objektu přímo do okna vytváření spouštěče. Nebo povolením vytvoření dočasného spouštěče, který nemá specifikovaný cílový objekt. Nebo by si okno spouštěče mohlo zapamatovat doposud vyplněné hodnoty, uživatel by se vrátil na hlavní obrazovku, vytvořil potřebný cílový objekt a poté by se vrátil do okna vytváření spouštěče.

Uživatelé často nepostřehli, že šedé zbarvení některých objektů znamená, že jsou pro tento typ spouštěče zakázané.

Do okna vytváření spouštěče jsme přidali nápovědu. Podobná nápověda by mohla být přidána i do dalších oken.

Upozornění. Uživatel by měl dostávat upozornění, která mu budou sdělovat, proč nemůže konkrétní akci provést.

Editace objektu. Okna pro vytváření jednotlivých objektů by mohla zároveň sloužit pro jejich editaci. Editace by se zpřístupnila pomocí dvojitého kliknutí v Outlineru nebo možností „Editovat...“ v okně s vlastnostmi. Dvojitě kliknutí na objekt pro editaci by bylo pravděpodobně intuitivní, ale aktuálně je dvojitě kliknutí využíváno pro zaměření pohledu kamery na objekt.

Možnost editace je nově přidána pomocí možnosti editovat v sekci properties.

Prohlížení scény. Tlačítek pro změnu rychlosti přibližování a přepnutí pohledu si nikdo nevšiml. Měly by být buď lépe popsány a mít nějaký sjednocující nadpis nebo ještě lépe by jejich funkce měly být zautomatizovány, aby tato tlačítka vůbec neměla důvod k existenci. Nastavování rychlosti přibližování kamery by se řídilo vzdáleností kamery od středu otáčení obsahu 3D okna a pohled by se nějak vhodně měnil sám, např. po vytvoření nového objektu by se na tento objekt zamířil.

Rychlost prohlížení se nově řídí automaticky a zaměřit kameru na objekt lze dvojitým kliknutím v sekci outliner.

Neexistující scéna. Vytvořit nějaký obsah pro případ, že nebude existovat žádná scéna, třeba odkaz na vytvoření nové scény nebo nápovědu.

Nově tento obsah existuje.

Krok zpět. Během testování jsme přišli na to, že krok zpět jako ve službě Augmara by byl velmi užitečný např. po nechtěném vymazání objektu.

Scénář. Scénář se po vytvoření automaticky nevybral a také není jak poznat, jak se aktuálně otevřený scénář jmenuje. Nově se vybírá automaticky.

Video. Uživatel by uvítal, kdyby si mohl odkazovaný soubor v objektu video přímo přehrát v aplikaci.

Struktura aplikace. Uživatele prý nezajímá, jakou má strom objektů strukturu. Uživatel chce například přidat model, ale to mu není jednoduše umožněno. Aby mohl přidat model, musí nejdříve vytvořit scénář, scénu, ukotvení, lokátor a pak až může přidat model. Navíc ho musí ještě nahrát (vybrat soubor na svém počítači a kliknout na možnost nahrát) a poté vybrat z nabídky modelů nahraných na serveru. To je mnoho překážek, zvláště když uživatel systém používá poprvé. Tedy náš systém by se z hlediska intuitivnosti a jednoduchosti dal zlepšit. Každá možnost je přístupná jen z jednoho místa. To je sice dobré z hlediska přehlednosti, ale aplikace kvůli tomu může být náročnější na používání.

Zkratky v aplikaci. Z předchozí zpětné vazby vyplývá, že by uživatelé ocenili různé zkratky jako vytvoření objektu Scene a Marker/GPS zároveň, vytvoření objektu Trigger přímo jako možnost objektu Button. Je otázkou, jestli takové zkratky implementovat, protože by v důsledku uživatel mohl mít menší přehled o tom, co právě vykonal a zmíněné vytvoření objektu Trigger z objektu Button by odporovalo v aplikaci používanému pravidlu, že nové objekty se vytvářejí jako potomci vybraného objektu. Jestli ale uživateli na těchto negativech záleží a jestli by převážila užitečnost zkratek, to by se nejlépe zjistilo dalším testováním.



Kapitola 7

Budoucí vývoj

Budoucí vývoj by měl z velké části reagovat na zbývající podněty uživatelů z kapitoly 6.

Měli bychom se snažit zvýšit intuitivnost aplikace. Objekt Locator by měl být dobrovolný, případně by se měl vytvářet automaticky. Pokud bychom měnili hierarchii scénáře, bylo by ale nutné dané změny reflektovat i v mobilní aplikaci pro prohlížení AR obsahu. Měli bychom zvážit přidání různých zkratk do uživatelského rozhraní.

Měli bychom lépe ošetřit chybové stavy, vytvořit nápovědu v aplikaci pro jednotlivá okna, přidat možnost prohlížení nahraného obsahu (videa, obrázky) přímo v aplikaci, zlepšit organizaci kódu aplikace. Do aplikace by měla být přidána podpora češtiny a měla by umět za běhu mezi jazyky přepínat.

Měli bychom zlepšit práci s X3DOM (zamezit přeskokování kamery, zakázat některá ovládání, stabilizovat rychlost přibližování, zamezit zakrytí gizma) (zmíněno v 4.3 a 6). Tyto změny by však mohly vyžadovat zásah přímo do kódu frameworku X3DOM.

Kapitola 8

Závěr

Naším úkolem bylo navrhnout a implementovat webový portál pro vytváření AR obsahu, testovat ho s uživateli a při vývoji reagovat na jejich podněty.

V úvodu jsme se seznámili s možnými významy slova augment, s vzrůstající oblíbeností rozšířené reality a s tím, v jakých odvětvích ji můžeme očekávat. Přiblížili jsme si její nejvíce uznávanou definici, podle které může nabývat i mnoha běžně neznámých forem. Ukázali jsme si možnosti ukotvení vizuální formy AR a to jak pomocí markeru, tak pomocí souřadnic GPS.

Uvedli jsme si požadavky na naší aplikaci, seznámili jsme se se službami Augmara a Augment a uvedli jsme, v čem jim naše navrhované řešení bude podobné a v čem se bude lišit. Jednotlivé požadované funkcionality jsme převedli na objekty a vytvořili jednotný způsob práce s nimi. Definovali jsme strukturu AR obsahu pro využití mobilní aplikací.

Popsali jsme, jak se s aplikací pracuje, popis jsme doprovodili snímky obrazovky a poskytli jsme možnost vyzkoušení aplikace. V kapitole Testování jsme shrnuli zpětnou vazbu od uživatelů, navrhli jsme možná řešení vyvstalých problémů, některá jsme při vývoji rovnou aplikovali.

Náš úkol jsme splnili, ale reakce na některé získané podněty je ponechána pro budoucí vývoj.



Literatura

- [1] PTC Inc, *Image Targets*, navštíveno 20. prosince 2018 [<https://library.vuforia.com/articles/Training/Image-Target-Guide>].
- [2] Vakoms, *Everything You Need to Know to Build Location-Based AR App - Vakoms Blog* [<https://blog.vakoms.com/everything-you-need-to-know-to-build-location-based-ar-app/>] (navštíveno 20. prosince 2018)
- [3] Web3D Consortium, *What is X3D?* [<http://www.web3d.org/x3d/what-x3d>] (navštíveno 20. prosince 2018)
- [4] Fraunhofer-Gesellschaft, - *x3dom.org* [<https://www.x3dom.org/>] (navštíveno 20. prosince 2018)
- [5] Behr, Johannes, Peter Eschler, Yvonne Jung a Michael Zöllner, *X3DOM: a DOM-based HTML5/X3D integration model*, Web3D (2009) [<http://www.web3d.org/wiki/images/3/30/X3dom-web3d2009-paper.pdf>] (navštíveno 3. ledna 2019)
- [6] Příspěvatelé OpenStreetMap, *OpenStreetMap*, [<https://www.openstreetmap.org/>] (navštíveno 20. prosince 2018)
- [7] Google, *Google maps API* [<https://developers.google.com/maps/>] (navštíveno 20. prosince 2018)
- [8] Augment SAS, *Project visualization platform | Augment* [<http://www.augment.com/>] (navštíveno 20. prosince 2018)
- [9] Augmara Technologies GmbH, *Augmara - The Augmented Reality Technology Platform* [<https://www.augmara.com/>] (navštíveno 3. ledna 2019)

- [10] digital agency Takondi AG, *TARTT - Augmented Reality CMS, SDK, App* [<https://www.tartt.io/>] (navštíveno 18. června 2018)
- [11] Augmara Technologies, *Discover Augmara* [<https://www.youtube.com/watch?v=W6Qa1QrxmzI>] (navštíveno 20. prosince 2018)
- [12] Augment, *Augment - Getting Started* [<https://www.youtube.com/watch?v=DpKKw71ct4U>] (navštíveno 20. prosince 2018)
- [13] Dan Foy, *Apple* [<https://www.flickr.com/photos/orangeacid/212833788/>] (navštíveno 20. prosince 2018)
- [14] Fonticons, Inc., *Font Awesome* [<https://fontawesome.com/>] (navštíveno 20. prosince 2018)
- [15] Google, *Maps Static API* [<https://developers.google.com/maps/documentation/maps-static/>] (navštíveno 20. prosince 2018)
- [16] shukshin.ivan, odpověď na *Converting Pixels to LatLng Coordinates from google static image* [<https://stackoverflow.com/questions/47106276/>] (navštíveno 20. prosince 2018)
- [17] Google, *Elevation Service* [<https://developers.google.com/maps/documentation/javascript/examples/elevation-simple>] (navštíveno 20. prosince 2018)
- [18] Dieter Schmalstieg, and Tobias Hollerer, *Augmented Reality: Principles and Practice (Usability)*, Addison Wesley, 2016, 25-27
- [19] Azuma, R. T., *A survey of augmented reality. Presence: Teleoperators and Virtual Environments* 6, 4, MIT Press, 1997, 355–385.
- [20] Micheal Lanham, *Augmented Reality Game Development*, Packt Publishing, 2017, 210.
- [21] Micheal Lanham, *Augmented Reality Game Development*, Packt Publishing, 2017, 272.
- [22] Martin Malý, *REST: architektura pro webové API* [<https://www.zdrojak.cz/clanky/rest-architektura-pro-webove-api/>] (navštíveno 14. prosince 2020)
- [23] Fraunhofer-Gesellschaft, - *x3dom.org* [<https://doc.x3dom.org/author/Time/TimeSensor.html>] (navštíveno 20. prosince 2020)

- [24] Brian Close, *8 Industries Benefitting from Augmented Reality* [<https://lightguidesys.com/blog/industries-benefitting-from-augmented-reality/>] (navštíveno 28. prosince 2020)

Příloha A

Serializovaný scénář

Následuje příklad formátu souboru, který slouží pro uložení scénáře na server a následně pro použití mobilní aplikací pro prohlížení AR obsahu.

```
{
  "objectType": "Scenario",
  "nickname": "",
  "name": "Scenario0",
  "_comment": "",
  "namingCounters": {
    "Scenario": 1,
    "Scene": 3,
    "Marker": 1,
    "GPS": 1,
    "Locator": 2,
    "Annotation": 1,
    "Model": 3,
    "ModelPart": 4,
    "MyObject_Animation": 4,
    "Button": 2,
    "Event": 1,
    "Trigger": 1,
    "Action": 1,
    "Video": 1,
    "Picture": 1
  },
  "scenes": [
    {
      "objectType": "Scene",
      "nickname": "",
      "name": "Scene1",
      "_comment": "",
      "annotation": {},
      "anchor": {
        "objectType": "Marker",
        "nickname": "",
        "name": "Marker0",
        "_comment": "",
        "markerOrientation": "floor",
        "imageAssetFilename": "apple.jpg",
        "width": "2",
        "imageAssetId": "160",
        "locators": [
          {
            "objectType": "Locator",
            "nickname": "",
            "name": "Locator0",
            "_comment": "",
            "annotation": {},
            "translation": {
              "x": -0.661195245632527,
              "y": 0,
              "z": 0
            }
          }
        ]
      }
    }
  ]
}
```

A. Serializovaný scénář

```
"models": [
  {
    "objectType": "Model",
    "nickname": "",
    "name": "Model0",
    "_comment": "",
    "annotation": {
      "objectType": "Annotation",
      "nickname": "",
      "name": "Annotation0",
      "_comment": "",
      "content": "This is a rhino model. It contains parts and animations."
    },
    "translation": {
      "x": 0,
      "y": 0.1083959626037215,
      "z": 0
    },
    "rotation": {
      "x": 0,
      "y": 0,
      "z": 0,
      "w": 1
    },
    "scale": {
      "x": 0.3,
      "y": 0.3,
      "z": 0.3
    },
    "modelAssetFilename": "rhinoWithModelParts.x3d",
    "modelAssetId": "161",
    "modelParts": [
      {
        "objectType": "ModelPart",
        "nickname": "head",
        "name": "ModelPart0",
        "_comment": "",
        "annotation": {},
        "x3dId": "modelPart_head"
      },
      {
        "objectType": "ModelPart",
        "nickname": "body",
        "name": "ModelPart1",
        "_comment": "",
        "annotation": {},
        "x3dId": "modelPart_body"
      }
    ],
    "animations": [
      {
        "objectType": "MyObject_Animation",
        "nickname": "rhino_time_sensor",
        "name": "MyObject_Animation0",
        "_comment": "",
        "x3dId": "rhino_time_sensor"
      },
      {
        "objectType": "MyObject_Animation",
        "nickname": "rhino_horn_time_sensor",
        "name": "MyObject_Animation1",
        "_comment": "",
        "x3dId": "rhino_horn_time_sensor"
      }
    ]
  }
],
"buttons": [
  {
    "objectType": "Button",
    "nickname": "turn on",
    "name": "Button0",
    "_comment": ""
  },
  {
    "objectType": "Button",
    "nickname": "turn off",
```

```

        "name": "Button1",
        "_comment": ""
    }
    ]
}
],
"events": [
    {
        "objectType": "Event",
        "nickname": "",
        "name": "Event0",
        "_comment": "",
        "triggers": [
            {
                "objectType": "Trigger",
                "nickname": "",
                "name": "Trigger0",
                "_comment": "",
                "triggerType": "Is in proximity",
                "parameters": [
                    "reactivity radius (meters)",
                    "5"
                ],
                "targetName": "Locator0"
            }
        ],
        "actions": [
            {
                "objectType": "Action",
                "nickname": "",
                "name": "Action0",
                "_comment": "",
                "actionType": "Play MyObject_Animation",
                "parameters": [],
                "targetName": "MyObject_Animation0"
            }
        ]
    }
],
"pictures": [
    {
        "objectType": "Picture",
        "nickname": "",
        "name": "Picture0",
        "_comment": "",
        "assetId": "160",
        "filename": "apple.jpg"
    }
],
"videos": [
    {
        "objectType": "Video",
        "nickname": "",
        "name": "Video0",
        "_comment": "",
        "assetId": "162",
        "filename": "mujPrvniFilm.wmv"
    }
]
},
{
    "objectType": "Scene",
    "nickname": "",
    "name": "Scene2",
    "_comment": "",
    "annotation": {},
    "anchor": {
        "objectType": "GPS",
        "nickname": "",
        "name": "GPS0",
        "_comment": "",
        "gpsCoordinates": {
            "lat": 50.07615251773119,
            "lng": 14.419712282072341,
            "ele": 207.1505432128906
        }
    }
},

```

A. Serializovaný scénář

```
    "locators": [
      {
        "objectType": "Locator",
        "nickname": "",
        "name": "Locator1",
        "_comment": "",
        "annotation": {},
        "translation": {
          "x": 0,
          "y": 0,
          "z": 0
        },
        "models": [
          {
            "objectType": "Model",
            "nickname": "",
            "name": "Model2",
            "_comment": "",
            "annotation": {},
            "translation": {
              "x": 15.351139451099012,
              "y": 0,
              "z": -17.143470360840553
            },
            "rotation": {
              "x": 0,
              "y": 0.9340708854570924,
              "z": 0,
              "w": 0.3570876376205193
            },
            "scale": {
              "x": 18.8,
              "y": 18.8,
              "z": 18.8
            },
            "modelAssetFilename": "Deer.x3d",
            "modelAssetId": "163",
            "modelParts": [],
            "animations": []
          }
        ],
        "buttons": []
      }
    ],
    "events": [],
    "pictures": [],
    "videos": []
  }
}
```



Příloha B

Instrukce k testování

Testování programu pro vytváření scén v AR

Úvod

Program, který budete testovat, je nástroj pro vytváření podkladů pro rozšířenou realitu (AR). Takto vytvořený obsah je určen pro následné prohlížení uživateli mobilní aplikace (není součástí testování).

V programu vytváříme Scény. Scéna je vlastně strom objektů. Pro ukotvení Scény slouží buď **GPS** souřadnice (místo v reálném světě), nebo **Marker** (obrázek). Každý další objekt, který přidáváme, připojujeme k už nějakému existujícímu objektu.

Pokud Vám není jasné něco ohledně účelu aplikace, AR nebo zadání úkolů, zeptejte se.

Testování

Úkoly byste se měli snažit vyřešit sami. Až tehdy, pokud Vám něco nepůjde delší dobu vyřešit, se zeptejte.

Pokud nebudete rozumět nějakému anglickému slovu, zeptejte se.

Scéna 1 – jelen

Vytvořte scénu na Karlínském náměstí, kde bude stát jelen v nadživotní velikosti a bude mít hlavu položenou na kostelní věži. Potom, co uživatel přijde na náměstí, zobrazí se mu popis toho, co vidí. Může si také stisknutím tlačítka přehrát video (potřebné video je tady v počítači).

Scéna 2 – nosorožec

Vytvořte scénu, která bude ukotvená k obrázku jablka. Ve scéně bude nosorožec (v .x3d souboru nosorožce (soubor je zde v počítači) jsou zahrnuté 2 animace a model je rozdělen na 2 podčásti). Nosorožec se bude jmenovat Bob. Potom, co se uživatel podívá PŘÍMO na model, zapnou se obě animace modelu.

Závěr

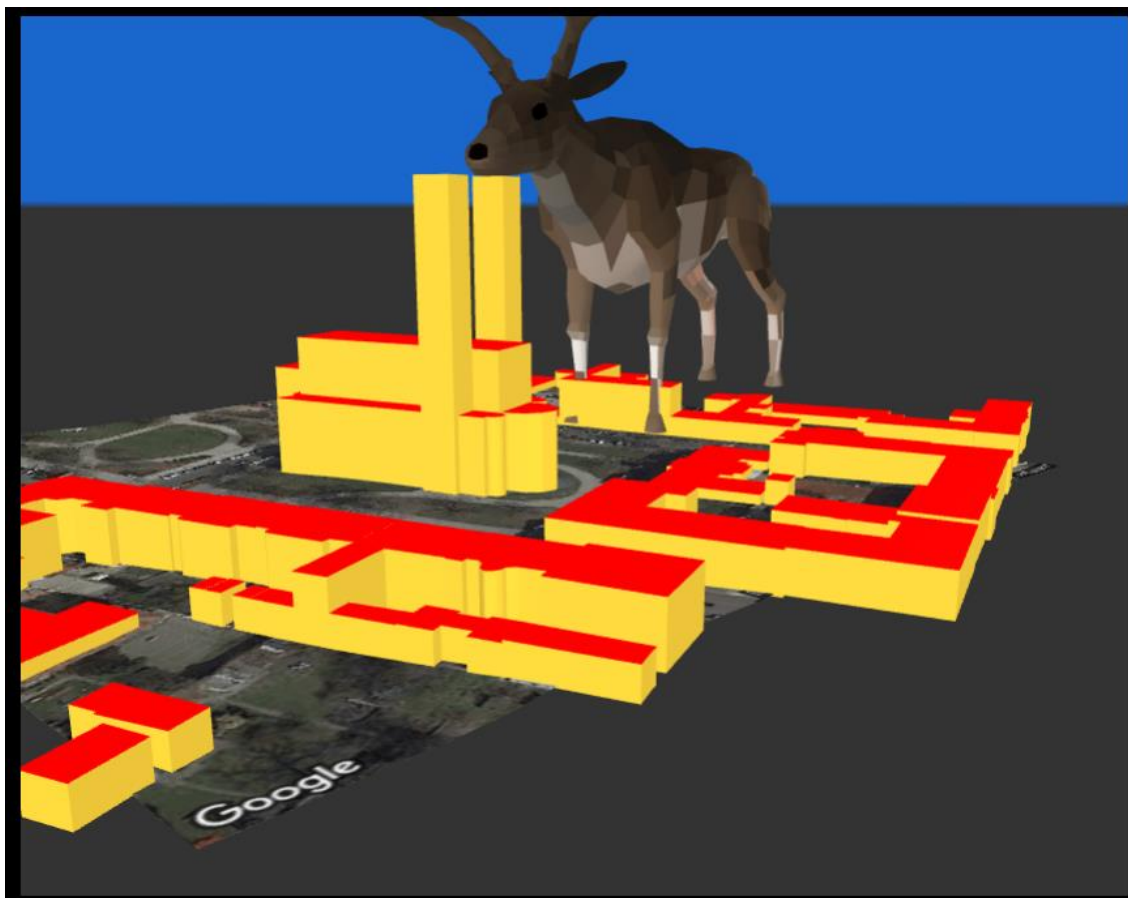
Co Vám dělalo problémy?

Co a jak byste zlepšili?

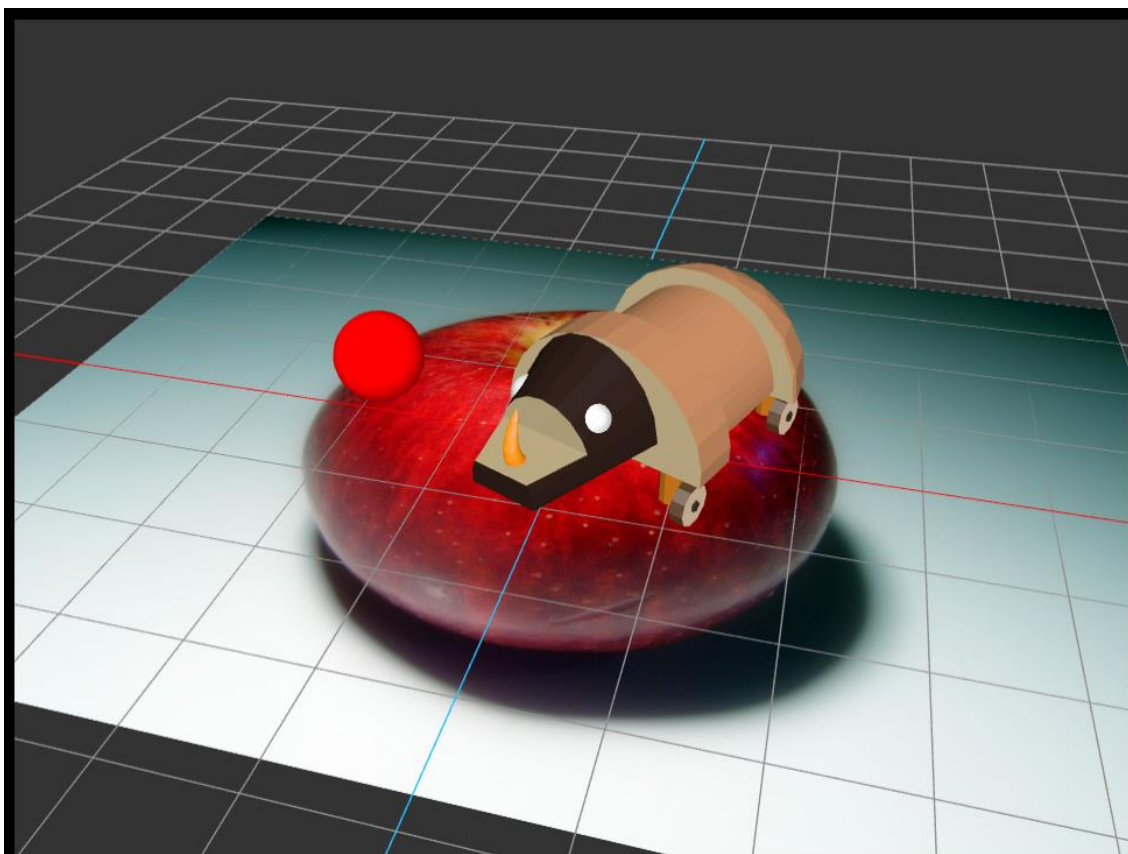
Zdál se Vám program intuitivní? Jak byste intuitivnost zlepšili?

Očekávaný výsledek

Úkol 1



Úkol 2





Příloha C

Obsah přiloženého CD

Složka	Obsah
Sources	Zdrojové kódy aplikace
Assets	Soubory pro testování aplikace
Thesis	Text bakalářské práce
ReadMe	Manuál