

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 812 19 Bratislava

Tím 18
Space Invaders

Dizajn s použitím obohatenej reality.
(projektová dokumentácia)

PEDAGOGICKÝ VEDÚCI

Ing. Juraj Štefanovič, PhD.

ČLENOVIA TÍMU

Bc. Andrej Babinec
Bc. Matej Budzel
Bc. Martin Mihálik
Bc. Matej Podstrelenec
Bc. Dárius Šilhár
Bc. Vladislav Zálešák

ŠKOLSKÝ ROK

2010/2011

Obsah

Obsah.....	i
1. Úvod.....	1
1.1. Zadanie projektu.....	1
1.2. Vstup do problematiky.....	1
2. Analýza.....	2
2.1. Obohatená realita (Augmented reality).....	2
2.2. Riešenia využívajúce obohatenú realitu (AR).....	2
2.2.1. Možnosti použitia obohatenej reality.....	2
2.3. Využitie obohatenej reality v mobilných telefónoch.....	7
2.3.1. Všeobecné koncepty.....	7
2.3.2. Konkrétne aplikácie.....	8
2.4. ARToolKit.....	12
2.4.1. O nástroji ARToolKit.....	12
2.4.2. Určenie polohy.....	13
2.4.3. ArToolKit rozhranie.....	14
2.4.4. Štruktúra.....	14
2.4.5. Dátové typy.....	15
2.4.6. Alternatívne implementácie.....	16
2.5. Mixed Reality Toolkit.....	18
2.5.1. O nástroji MRT.....	18
2.6. QR kód.....	19
2.6.1. Využitie QR kódu.....	21
2.6.2. Alternatívy.....	21
2.6.3. QR kód ako značka pre nástroje obohatenej reality.....	23
2.7. Kamery.....	24
2.7.1. Mobilný telefón.....	24
2.7.2. Webkamera.....	24
2.7.3. Digitálny fotoaparát.....	25
2.7.4. Digitálna kamera.....	25
3. Špecifikácia riešenia.....	26
3.1. Kontext riešenia.....	26
3.2. Koncept riešenia.....	26
3.2.1. Rozšírenie katalógu nábytku.....	26
3.2.2. Aplikácia v mobilnom telefóne (klientska aplikácia).....	27
3.2.3. Server.....	27
3.3. Vymedzenie riešenia.....	28
3.4. Požiadavky.....	28
3.4.1. Hardvérové požiadavky.....	28
3.4.2. Softvérové požiadavky.....	29
3.5. Procesy.....	32
3.5.1. Načítanie modelu nábytku do telefónu.....	32
3.5.2. Správa modelov v telefóne.....	32
3.5.3. Správa virtuálnych scén.....	33
3.5.4. Úprava katalógu.....	33
3.6. Prípady použitia.....	38

3.6.1.	Načítaj model	39
3.6.2.	Prezri model	40
3.6.3.	Prezri info	41
3.6.4.	Odstráň model	41
3.6.5.	Vytvor scénu	42
3.6.6.	Uprav scénu	42
3.6.7.	Odstráň scénu	43
3.6.8.	Spracuj požiadavku	44
3.6.9.	Vygeneruj kód	44
4.	Návrh riešenia	47
4.1.	Serverová aplikácia	47
4.1.1.	Dátový model	47
4.1.2.	Komunikácia s klientskou aplikáciou	48
4.1.3.	Obsah QR kódu	49
4.2.	Klientska aplikácia	49
4.2.1.	Komponenty	49
4.2.2.	Ukladanie dát	50
5.	Prototyp (výstup zimmého semestra)	52
5.1.	Vývojové prostredie a potrebné pluginy	52
5.1.1.	Inštalácia	52
5.1.2.	Podpora kamery v emulátore	53
5.1.3.	Pokus s virtuálnym počítačom	53
5.2.	Grafické používateľské rozhranie pre Android	55
5.2.1.	Základné grafické používateľské rozhranie	56
5.3.	Prototyp pre AR	56
5.3.1.	NyARToolKit	57
5.3.2.	AndAR	57
5.3.3.	Budúce smerovanie	58
5.4.	Prototyp aplikácie na spracovanie QR kódu	58
5.4.1.	Opis prototypu	58
5.4.2.	Prostredie a testovanie	58
5.4.3.	Získavanie a pracovanie dát z kamery zariadenia	58
5.5.	3D prostredie	59
5.5.1.	3D model a modelovanie	59
5.5.2.	3D grafické aplikácie	59
5.5.3.	Možnosti 3D grafických aplikácií	60
6.	Implementácia	62
6.1.	Andar	62
6.2.	Implementácia knižnice ZXing	64
6.2.1.	BarCodeScanner	64
6.3.	Server	66
6.3.1.	Služby	66
6.3.2.	Použité prostriedky	68
6.4.	Grafické používateľské rozhranie	70
7.	Zhodnotenie	73
	PRÍLOHA A: Používateľská príručka	1

1. Úvod

1.1. *Zadanie projektu*

Obohatená realita (augmented reality) je moderná technologická verzia rozhrania človek – stroj, respektíve človek – aplikácia/ virtuálne dielo/ virtuálna realita. Príkladom jednoduchej realizácie obohatenej reality je snímanie obrazu kamerou a prezentácia tohto obrazu na monitore počítača, doplneného o virtuálne objekty. Pomerne triviálnym, avšak prekvapivo účinným a mnohými autormi používaným nástrojom je otvorená softvérová platforma ARToolKit, ktorá umožňuje zachytiť obraz z kamery v reálnom čase, prezentovať ho na obrazovke počítača a rozpoznať v ňom jednoduché orientačné značky, umožňujúce správne polohovanie virtuálnych objektov, ktoré chceme do obrazu doplniť. Cieľom projektu je predviesť prácu dizajnéra s použitím tejto metódy, ktorá umožní interaktívne vkladať navrhované riešenie ako virtuálny objekt do existujúceho prostredia a takto vizualizovať návrh. Jednou z aplikačných možností je interiérový dizajn, pri ktorom bude umožnené vidieť existujúci interiér, interaktívne doplnený o navrhované virtuálne prvky.

- Research Thesis: interactive 3d modelling in outdoor augmented reality worlds: <http://www.tinmith.net/wayne/thesis/piekarski-thesis.htm>
- príklad jednej vyspelej a ocenej implementácie: http://studierstube.icg.tu-graz.ac.at/invisible_train/

1.2. *Vstup do problematiky*

Mnohé moderné technológie, ako napríklad digitálna fotografia, digitálne video, GPS, sieťové prepojenia, počítačová grafika, alebo počítačové 3D modelovanie napriek svojmu označeniu „moderné“ existujú a sú využívané už desiatky rokov. Navzdory tomu, možnosti využitia týchto jednotlivých technológií a najmä ich kombinácií nie sú doposiaľ ani zďaleka vyčerpané. Až dnes, keď pokrok v miniaturizácii umožňuje využitie všetkých spomenutých technológií v jednom prenosnom zariadení, dostáva sa do popredia aj vývoj aplikácií, ktoré tieto možnosti vhodne využívajú.

Veľmi zaujímavým druhom takýchto aplikácií sú tie, ktoré využívajú takzvanú obohatenú realitu (Augmented reality). Jedná sa o spojenie počítačovej grafiky a vyobrazenia reality zachytávaného kamerou. Obohatenie reality je teda jej doplnenie o virtuálne objekty a prvky ktoré v skutočnosti neexistujú a to v reálnom čase a väčšinou aj v troch rozmeroch. Takéto obohatenie reality môže byť ešte efektívnejšie a efektnejšie s využitím ďalších dát ktoré je schopné poskytnúť aj mobilné zariadenie, najmä informácie o polohe zo systému GPS, informácie o smerovaní a natočení zariadenia zo zabudovaného kompasu a polohových senzorov a tiež s pomocou algoritmov na rozpoznávanie obrazu a značiek.

Práve vývojom aplikácie obohacujúcej realitu, určenej pre mobilné zariadenia, sa zaoberá aj náš tím v tomto projekte.

2. Analýza

2.1. Obohatená realita (Augmented reality)

Obohatená realita ako rýchlo rozvíjajúca sa moderná technológia informatiky spája pod sebou i ďalšie oblasti z technologického sveta. Dnešné kamery schopné digitálne zachytávať obraz a posielat' ho ďalším zariadeniam, rovnako ako aj knižnice vo vývojových prostrediach, schopné tento obraz spracovávať.

Do oblasti obohatenej reality patrí i technológia QR kódov so schopnosťou zaznamenávať informácie do dvojrozmerných matíc. Rovnako tu nájdeme i technológiu mobilných telefónov podporujúcich operačný systém Android, umožňujúcu názorne demonštrovať silu obohatenej reality. Preto pokladáme za dôležité si spomenuté technológie stručne opísať a analyzovať ich vlastnosti podporujúce obohatenú realitu. Na konci analýzy predstavíme i koncept reprezentujúci spôsob spojenia týchto technológií a neskoršieho vytvorenia systému zameraného na obohatenú realitu.

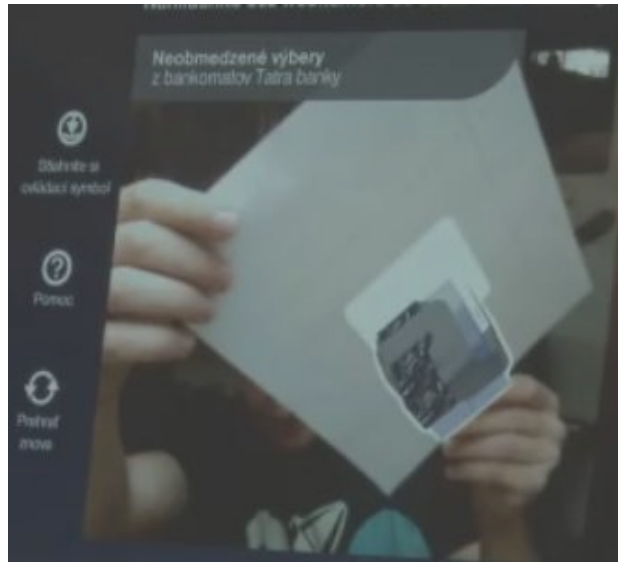
2.2. Riešenia využívajúce obohatenú realitu (AR)

2.2.1. Možnosti použitia obohatenej reality

Obohatená realita (AR), ako technológia sa stáva v dnešnej dobe čím ďalej, tým viac známou pre širokú verejnosť. Jej použiteľnosť je realizovateľná skutočne v každom smere. Rozširuje možnosti 3D vizualizácie a spracovávaní obrazu. Veľkým pôsobiskom využitia je napríklad vzdelávanie. Vo vzdelávacom procese sa už od dávna presadzuje myšlienka školy hrou resp. učenie sa novým poznatkom na základe audiovizuálnych vnemov. Jedno veľké využitie je teda vo vzdelávacom procese a to napríklad formou vytvorenia interaktívnej AR encyklopédie.

Ďalšie veľké využitie AR je v komerčnej sfére. Preto je veľmi zaujímavé použitie AR pri vytvorení katalógu produktov nejakého obchodu s použitím AR. Tak by bolo umožnené, aby si zákazník resp. potenciálny kupec tovaru tovar pozrel v priestore a z každej strany.

Na Slovensku bola prvý krát oficiálne použitá technológia obohatenej reality spoločnosťou Tatra banka [1]. Ide o jednoduchú web aplikáciu, kde je prostredníctvom web kamery ovládaná jej funkcionálna. Princíp fungovania aplikácie je v tom, že web kamera sníma značku (logo spoločnosti) vytlačenú na papieri. Na základe pohybu tejto značky pred web kamerou je táto aplikácia ovládaná. Na Obr. 1 nižšie je zobrazená táto aplikácia.



Obr. 1 - Balík služieb, aplikácia používajúca obohatenú realitu [1].

❖ Interaktívna AR encyklopédia

Bola vytvorená encyklopédia, v ktorej bola použitá obohatená realita. Projekt je realizovaný formou knižnej verzie encyklopédie, ktorá obsahovala značky. Tieto značky sú prostredníctvom webkamery a implementovaného nástroja ARToolKit načítavané a rozpoznávané. Na základe tejto značky sa vyhodnotí, aký objekt sa má na značku umiestniť. Taktiež sa vyhodnotí ako má byť zobrazovaný objekt otočený a v akej veľkosti sa má znázorniť. A na otvorenej strane encyklopédie sa na obrazovke zobrazuje 3D obraz objektu. Tak je umožnené si tento objekt preskúmať z každej strany.



Obr. 2 - Strana encyklopédie, značka ARToolKit-u [2].

Na Obr. 2 je ukázané, ako vyzerá jedna zo stránok tejto encyklopédie. Táto strana obsahuje značku, ktorá je rozpoznateľná pre knižnicu ARToolKit. A tak po namierení zariadenia, na ktorom je nainštalovaný potrebný softvér sa zobrazí na displeji tohto zariadenia 3D objekt.



Obr. 3 - Zobrazenie 3D objektu [2].

Obr. 3 ukazuje, ako vyzerá zobrazený 3D objekt obohatenej reality. Tento objekt je možné sledovať z rôznych strán.

❖ Interiérové zariadenie

Keďže téma projektu je dizajn s použitím obohatenej reality a plánujeme sa zamerať na interiérový dizajn, jedno z možných využití je v obchodnom sektore. Je množstvo obchodov s nábytkom a interiérovým zariadením. V žiadnom z týchto obchodov nie je zatiaľ použitá obohatená realita. Prostredníctvom obohatenej reality by tak bolo možné zariadiť si miestnosť obytného priestoru nábytkom, pričom nábytok by sa nachádzal v priestore len virtuálne a tak by bolo možné si vybrať medzi viacerými variantmi zariadenia interiéru.



Obr. 4 - Umiestnenie objektu do priestoru [4].

Na Obr. 4 je znázornené umiestnenie sedačky do priestoru miestnosti. Táto sedačka je obohatená realitou, čiže v skutočnosti sa v danom priestore nenachádza. Je zobrazená len na displeji mobilného zariadenia.

❖ Časopis s použitím obohatenej reality

Nemecký časopis Sueddeutsche Zeitung je prvý časopis na svete, ktorý použil technológiu obohatenej reality. Časopis je čitateľný aj sám o sebe. Obsahuje texty a obrázky. Obohatená realita je realizovaná prostredníctvom rozpoznávania obrázkov v časopise a dopĺňania obsahu časopisu na displeji mobilného zariadenia. K zobrazeniu obohatenej reality časopisu je potrebné mať mobilné zariadenie s operačným systémom iOS alebo Android a nainštalovanú voľne dostupnú aplikáciu Junaio. Na Obr. 5 je ukázané zobrazenie časopisu cez mobilné zariadenie iPhone a ako je možné vidieť, k obrázkom v časopise sú pridávané doplňujúce informácie vo forme bublín s textami.



Obr. 5 - Zobrazenie obohatenej reality [3].

Časopis ponúka aj zobrazenie 3D modelu. Táto skutočnosť napomáha k vytvoreniu lepšej predstavy používateľa resp. čitateľa o znázorňovanom objekte. Zobrazenie 3D objektu po namierení mobilného zariadenia na daný objekt v časopise je ukázané na Obr. 6 nižšie.



Obr. 6 - Zobrazenie obohatenej reality- 3D model [3].

❖ AR katalóg produktov

Vytvorenie katalógu produktov s použitím AR je samo o sebe zaujímavé. Katalóg by bol realizovaný formou klasického katalógu produktov, s tým rozdielom, že pri každom produkte by bola umiestnená značka, ktorá by reprezentovala daný produkt. Pri snímaní tohto produktu prostredníctvom zariadenia vybaveného web kamerou a príslušným softvérom by sa zobrazil na mieste značky 3D model daného tovaru. Ak by išlo napríklad o katalóg produktov so zameraním na bytové zariadenia, mohol by si človek vybraný nábytok nie len pozrieť v katalógu na obrázku, ale aj v priestore pomocou obohatenej reality. Tieto značky by mohli byť odnímateľné od konkrétnej stránky katalógu. Tak by bolo umožnené zákazníčkovi umiestniť nábytok v interiéri miestnosti a pozrieť sa cez displej mobilného zariadenia, či by sa mu daný kus nábytku hodil.

Katalóg produktov s použitím AR by uľahčil výber tovaru zákazníčkovi. A to hlavne v tom, že by bolo možné si konkrétny tovar pozrieť v priestore, čím sa vytvorí lepší vizuálny vnem.

❖ AR turistický sprievodca

Iným zaujímavým využitím obohatenej reality by bolo jej aplikovanie pri poskytovaní informácií turistom alebo návštevníčkovi múzea, či expozície. Týmto spôsobom by mohli napríklad mobilné aplikácie zobraziť identifikačný štítok a informácie k historicky zaujímavým budovám na ktoré sa turista práve pozerá objektívom svojho zariadenia a to na základe jeho polohy určenej GPS a smerovania a natočenia mobilného zariadenia. Podobná aplikácia by mohla informovať návštevníčkov galérie alebo múzea ohľadom jednotlivých exponátov a to s použitím optických značiek, ktoré by aplikácia rozpoznávala.

2.3. Využitie obohatenej reality v mobilných telefónoch

Mobilné telefóny po technickej stránke dospeli v posledných rokoch do stavu keď sa už výpočtovou kapacitou dokážu takmer vyrovnat' aj prenosným počítačom, a navyše s príchodom dotykových displejov sa výrazne zlepšila aj zobrazovacia schopnosť. Vstavaný fotoaparát je už absolútne bežnou súčasťou. A všetky tieto aspekty napomohli k tomu, aby sa mobilné telefóny stali vhodným prostriedkom pre princípy obohatenej reality. Najmä z toho dôvodu, že používateľ ho môže mať telefón pri sebe vždy a všade.

V tejto časti si najprv prejdeme na akých princípoch môže fungovať obohacovanie reality cez mobilný telefón, a potom aj na konkrétne aplikácie.

2.3.1. Všeobecné koncepty

Základom pri obohacovaní reality cez zariadenie ako je mobilný telefón, ale aj cez iné zariadenia, ktoré sprostredkovávajú informácie cez displej, je snímanie okolia kamerou, pričom sa následne do obrazu dopĺňujú externé prvky. Bol vykonaný aj pomerne obsiahly výskum v tejto oblasti na univerzitej pôde [13].

❖ Značky (a iné obrazové prvky)

Jednou z ciest ako dopĺňať značky do snímaného obrazu je pomocou sledovania špeciálnych grafických značiek. Teda pomocou platformy ARToolKit ak sa dá použiť v mobilnom telefóne, alebo aj inými cestami. Napríklad pomocou platformy StbTracker [5], ktorej princíp je rovnaký ako pri ARToolKit-e, ale bola vytvorená špeciálne pre mobilné zariadenia. Okrem toho je možné detekovať v obraze aj iné prvky, ako fotky a pod.

Takýto princíp sa využíva najmä pri dokresľovaní trojrozmerných objektov.

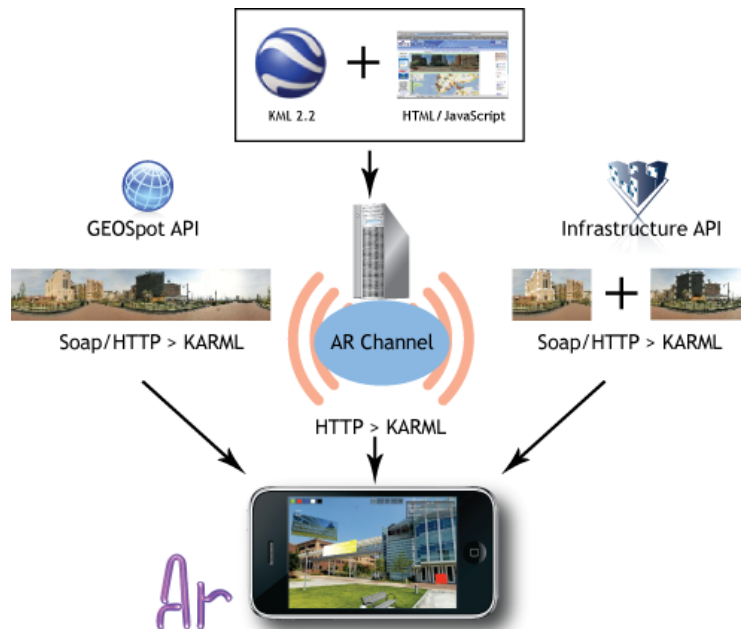
❖ Geopozícia

Mnohé moderné mobilné telefóny sú už vybavené vstavaným GPS lokalizátorom a obsahujú aj rôzne senzory na rozpoznávajúce polohu zariadenia. Kombináciou týchto dvoch prvkov sa dá taktiež doceliť dobrá orientácia v priestore. Výhodou oproti značkám je, že nie je potrebný žiaden externý prvok, ktorý by bolo potrebné snímať, ale v súčasnosti je presnosť GPS lokalizovania a vstavaných sensorov ešte pomerne povážlivá, a nemožno teda vykonávať úplne precízne dokresľovanie objektov. Ako bude opísane nižšie pri konkrétnych aplikáciách, užitočnosť využítie tohto princípu spočíva najmä v interaktívnom poskytovaní rôznych navigačných funkcií.

❖ Kharma

Pomerne nová platforma, ktorá má poskytovať možnosti obohacovania reality cez mobilné zariadenia je Kharma [6]. Nenavrhuje síce žiaden iný technický prístup, ale ide skôr o samotné informácie, ktoré sa používajú. Jej hlavnou princípom je sprístupňovanie obsahu na internete z viacerých zdrojov, pričom aplikácia na báze tejto platformy, by mala byť od ich implementácie nezávislá. Schéma prepojenia jednotlivých súčastí je na Obr. 7.

Samotná priestorová lokalizácia by mala fungovať na báze GEOSpot-ov, čiže konkrétnych bodov s geografickými, presne nameranými súradnicami. Dôvodom pre použitie takýchto bodov sú práve nedostatky v presnosti GPS lokalizovania. Používateľ by si teda pomocou mapy (a svojho GPS lokalizátora) našiel niektorý GEOSpot, indikoval by, že sa na ňom nachádza a podľa toho by sa vykonala kalibrácia aby sa informácie vykresľovali naozaj na správnom mieste.



Obr. 7 - Schéma platformy Kharma [6].

2.3.2. Konkrétne aplikácie

Za základný prameň informácií pre bežného používateľa sa v súčasnosti dá považovať Internet. Ten sa dostáva už takmer do každej sféry ľudského života a mobilné telefóny, to taktiež samozrejme využívajú. Takže aj možnosť prístupu do Internetu je už bežnou výbavou telefónov. A na tomto fakte sú založené aj skoro všetky nižšie opísané aplikácie.

V tomto prehľade sú reálne aplikácie, ktoré sa dajú plnohodnotne využiť a nejedná sa o žiadne prototypy alebo testovacie pokusy. Práve z dôvodu informatívnosti a ľahkej a bezproblémovej dostupnosti k GPS je populárne najmä orientovanie na základe súradníc a nie podľa značiek. Preto sa tu žiadne aplikácie so značkami nevyskytujú.

❖ Layar

Aplikácia vyvinutá pre telefóny iPhone1 a operačný systém Android2 [7]. Princíp spočíva v poskytovaní rozširujúcich informácií o svojom okolí. Tie sú rozdelené do viacerých vrstiev a môžu ich rozširovať všetci používatelia. Orientácia v priestore je na báze GPS a informácie sa získavajú z centrálného bodu, na ktorý sa dajú pripájať konkrétni poskytovatelia dát.

¹ Oficiálna stránka Apple iPhone - <http://www.apple.com/iphone/>

² Oficiálna stránka platformy Android - <http://www.android.com/>

Takýmto spôsobom sa dá prechádzať po ulici a vidieť informácie o bytoch na predaj, hodnotenia reštaurácií, ceny hotelov (Obr. 8) a podobne. Okrem toho sa dajú hrať aj rôzne iné hry. Zameranie vrstvy nie je obmedzené.



Obr. 8 - Ukážka využitia aplikácie Layar [<http://site.layar.com/company/blog/>].

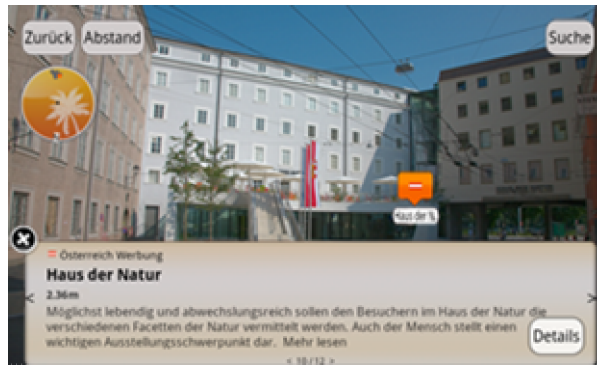
❖ Wikitude

Spoločnosť Wikitude³ poskytuje dva produkty obohatenej reality. Jedným z nich je World Browser [8] a druhým Drive[9]. Oba fungujú podobne ako Layar, a teda orientujú sa na základe GPS súradníc a senzorov v telefóne, ale World Browser (Obr. 9) je zameraný len na jednotnom poskytovaní informácií o okolí, ktoré nerozkladá do vrstiev a Drive (Obr. 10) je aplikácia na navigáciu počas šoférovania automobilu. Výhodou pri Drive je to, že aj keď sa vodič pozrie na displej telefónu, vidí na ňom aj reálnu premávku pred sebou, a teda stále môže rýchlo a presne reagovať. Obe aplikácie podporujú platformy iPhone, Android, Symbian⁴ a Bada⁵.

³ Stránka spoločnosti Wikitude - <http://www.wikitude.org/>

⁴ Oficiálna stránka platformy Symbian - <http://www.symbian.org/>

⁵ Oficiálna stránka platformy Samsung Bada - www.bada.com



Obr. 9 - Ukážka aplikácie Wikitude World browser [8].



Obr. 10 - Ukážka aplikácie Wikitude drive [9].

❖ TwittARound

Aplikácia TwittARound⁶ je príkladom prepojenia medzi sociálnymi sieťami a obohatenou realitou. V tomto prípade sa jedná o sieť Twitter⁷ a aplikácia vytvorená je pre platformu iPhone. Princíp je podobný ako pri Layar či Wikitude World browser, len s tým rozdielom, že na rozdiel od rôznych informácií serióznejšieho charakteru sú k súradniciam priradené zdieľané správy zo spomínanej siete.

❖ TagWhat

Ďalšou alternatívou zdrojov informácií, ktoré sa dajú vkladať do obrazu sú vzájomné zdieľané odkazy, obrázky a iné informácie len od samotných používateľov. Na takomto princípe funguje aplikácia TagWhat [10] (Obr. 11). Aplikácia je implementovaná na platformy iPhone a Android.

⁶ Blogový príspevok o aplikácií TwitARound - <http://i.document.m05.de/?p=685>

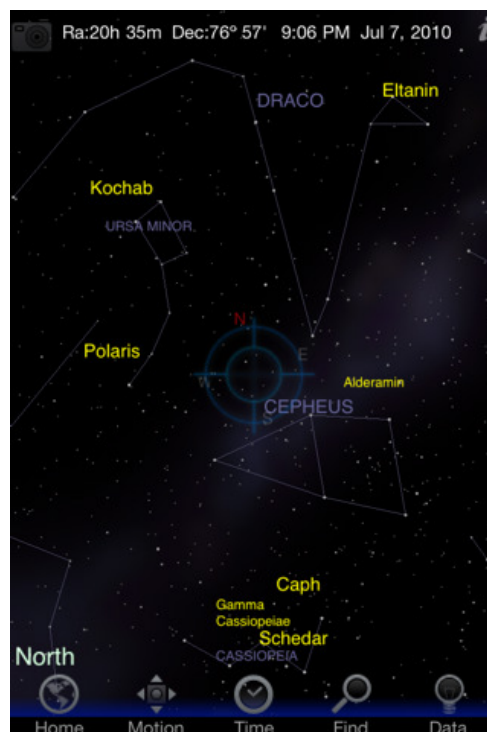
⁷ Sociálna sieť Twitter - <http://twitter.com/>



Obr. 11 - Ukážka aplikácie TagWhat [<http://www.androidtapp.com>].

❖ Pocket Universe

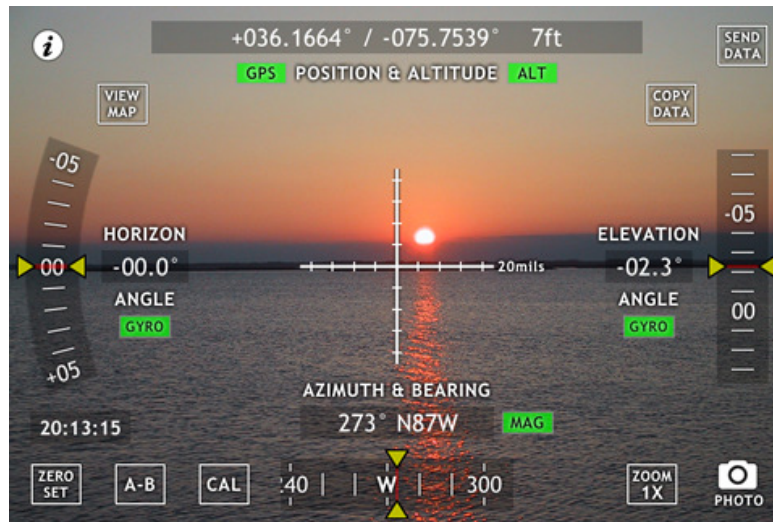
Zaujímavým obohatením reality je aj vykresľovanie hviezd, súhvezdí a ich názvov pri pohľade na oblohu cez mobilný telefón. Príkladom takéhoto prístupu je aplikácia Pocket Universe pre iPhone [11], ktorej ukážka je na Obr. 12.



Obr. 12 - Ukážka aplikácie Pocket Universe [<http://itunes.apple.com/us/app/pocket-universe-virtual-sky/id306916838?mt=8>].

❖ Theodolite

Táto aplikácia sa taktiež orientuje na základe GPS súradníc a z funkcionálneho hľadiska predstavuje aktívny kompas, ktorý do obrazu snímaného kamerou vykresľuje informácie o uhle natočenia, pozícií a podobne. [12] (Obr. 13).



Obr. 13 - Ukážka aplikácie Theodolite [12].

2.4. ARToolKit

2.4.1. O nástroji ARToolKit

ARToolKit je nástroj pre vývoj aplikácií orientovaných na obohatenú realitu. Jedná sa o systémy umožňujúce vkladanie virtuálnych objektov do reálneho sveta. Bol vyvinutý v roku 1999 pánom Hirokazu Katom z Narského inštitútu vied a technológií a bol uvedený na svet vďaka Washingtonskej univerzite. Medzi hlavné problémy obohatenej reality patrí sledovanie smeru pohľadu používateľa. Tento problém ako i mnohé iné rieši práve knižnica ARToolKit. Použitím špeciálnych algoritmov a knižníc zaoberajúcich sa uhlom medzi pohľadom kamery a rozpoznávacej značky. Príklad značky môžete vidieť na Obr. 14 [14].



Obr. 14 - Ukážka značky [14].

Dôvod pre implementáciu riešenia v prostredí ARToolKit pomôžu objasniť i nasledujúce jeho vlastnosti.

- jednoduché rozhranie pre vytváranie aplikácií
- nezávislosť od platformy
- existujú rozšírenia podporujúce všetky programovacie jazyky
- možnosti vstupu ako USB, Firewire a iné
- podpora formátov RGB, YUV420P, YUV
- viacnásobné sledovanie kamery
- modulárne API
- a iné.

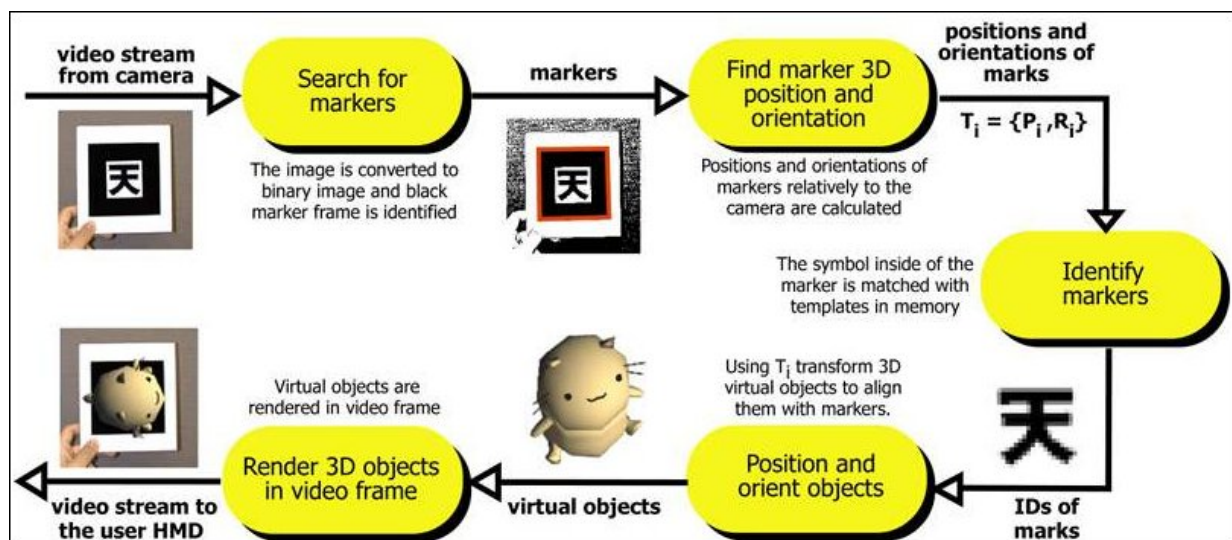
Postupne sa v rámci analýzy oboznámime so základnými vlastnosťami ako aj funkcionalitou, ktorú ARToolKit ponúka.

2.4.2. Určenie polohy

Správne určenie polohy patrí medzi zásadné vlastnosti ARToolKit-u. Postup určovania si pomocou niekoľkých krokov stručne opíšeme.

1. Kamera hľadá preddefinované značky.
2. Identifikuje štvorcový okraj obrázku, ktorý je konvertovaný do binárnej podoby.
3. Zistí pozíciu a orientáciu značky vzhľadom na kameru.
4. Symbol značky je porovnaný so symbolom uloženom v systéme.
5. Aplikuje transformácie virtuálnych 3D objektov a zarovná ich so značkou.
6. Objekty sú zobrazené na video výstupe.

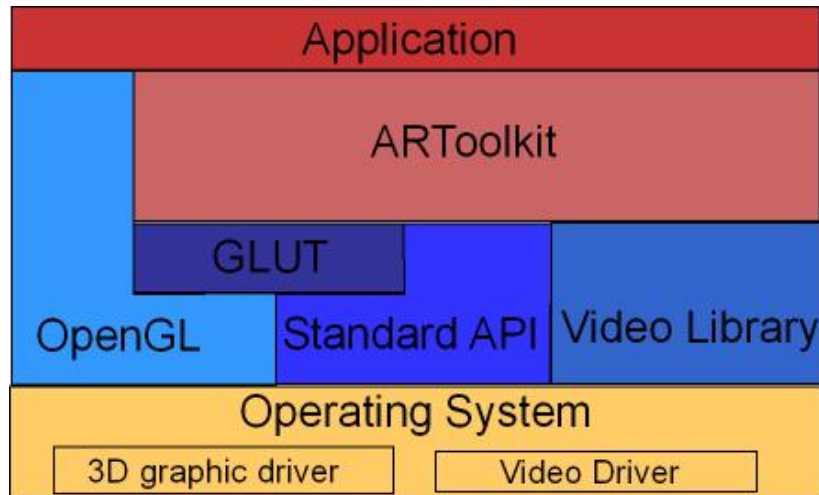
Pre lepšiu predstavu predkladáme i názornú schému na Obr. 15.



Obr. 15 - Určenie polohy [14].

2.4.3. ArToolKit rozhranie

ARToolKit, rovnako ako i GLUT patrí medzi softvérové toolkit-y. Pozostáva z predefinovaných funkcií, ktoré je potrebné volať v určitom poradí pri vytváraní AR programu. Pre renderovanie používa OpenGL a GLUT pre štandardy API, ktoré je vytvorené v jazyku C. Pre lepšiu ilustráciu uvádzame Obr. 16.



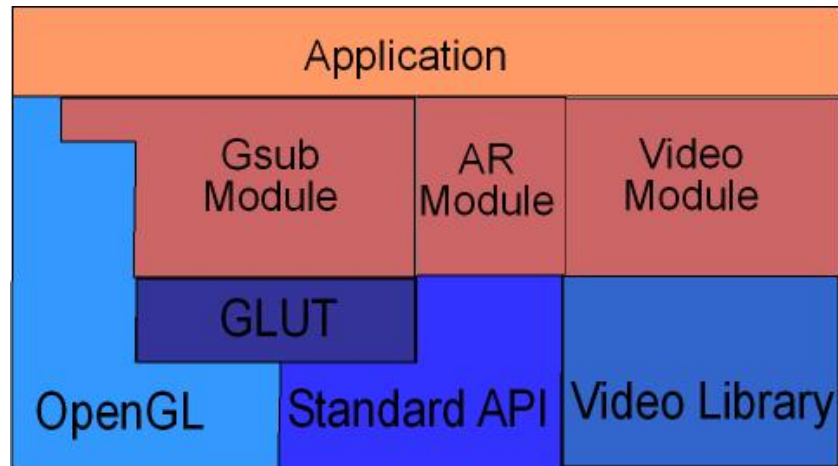
Obr. 16 - ARToolKit rozhranie [14].

2.4.4. Štruktúra

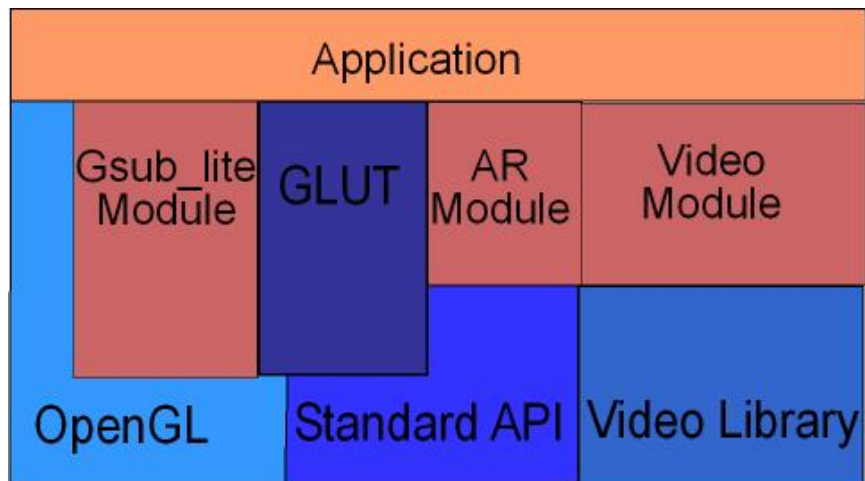
Knižnica ARToolKit-u pozostáva zo štyroch modulov:

- *AR modul* - tvorí jadro a pozostáva z rozpoznávania značky, kalibrácie a zbierania parametrov
- *Video modul* - zoskupenie funkcií pre prácu s videom. Tvorí obal nad platformou SDK a využíva jej štandardné funkcie.
- *Gsub modul* - grafické procedúry založené na knižniciach OpenGL a GLUT.
- *Gsub_Lite modul* - slúži ako náhrada modulu GSUB, ponúka efektívnejšie procedúry, ktoré sú nezávislé.

Nasledujúce dva obrázky (Obr. 17 a Obr. 18) reprezentujú hierarchickú štruktúru ArToolKit-u a jeho vzťah so závislými knižnicami.

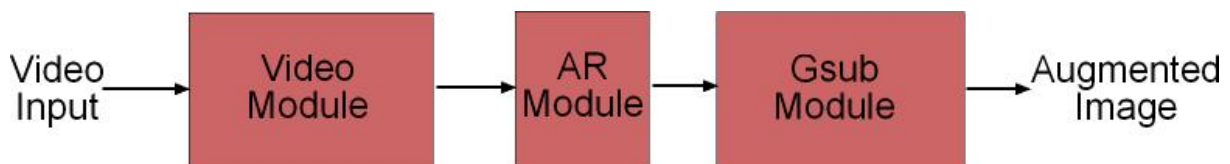


Obr. 17 - Hierarchická štruktúra ARToolkit-u [14].



Obr. 18 - Hierarchická štruktúra ARToolkit-u [14].

Dané moduly sa riadia tzv. „pipeline metaforou“ (video->sledovanie->zobrazenie). Používateľ môže tak ľahko jednotlivé moduly nahradiť ďalšími, napr. modul *gsub* za *Inverntor* renderer (Obr. 19).

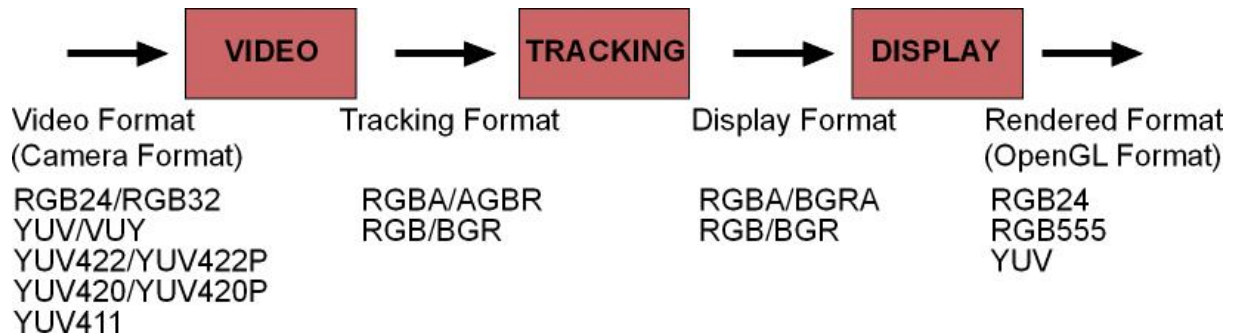


Obr. 19 - ARToolkit moduly [14].

2.4.5. Dátové typy

ARToolkit pracuje s množstvom premenných. Vo vnútri používa globálne premenné, ktoré obmedzujú opätovne zadané časti kódu. Medzi rozdielnymi modulmi používa rozdielne

formáty obrázkov. Obr. 20 zobrazuje všetky typy používaných formátov. Niektoré formáty sú k dispozícii len na niektorých platformách, alebo s niektorým hardvérom.



Obr. 20 - ARToolkit formáty [14].

2.4.6. Alternatívne implementácie

Pre otvorenie ARToolkit-u väčšiemu množstvu používateľov, boli vytvorené jeho rôzne modifikácie, umožňujúce prácu s ním i v inom ako pôvodnom jazyku. Postupne tak vznikli implementácie knižníc ako:

- ARToolkitPlus
- NyARToolkit
- FLARToolkit

Jednotlivé knižnice si v nasledujúcich odsekoch stručne opíšeme.

❖ ARToolkitPlus

AR Toolkit
Augmented Reality Tracking Library



Obr. 21 - ARToolkitPlus [15].

ARToolkitPlus bol vyvinutý v rámci projektu Hanheld AR a uvoľnený pod licenciou GPL. Je určený pre skúsených programátorov v jazyku C++. Táto verzia ARToolkitu pridáva nové funkcie, avšak nie je spätne kompatibilná keďže bolo kompletne prerobené jej API. Rovnako má problémy i s formátom geometrie VRML (Virtual Reality Modeling Language) [15].

Hlavné rysy ARToolKitPlus sú:

- objektovo riešené API
- dokáže pracovať až s 4096 značkami bez výrazného spomalenia behu
- variabilná šírka okraju značky
- podporuje prácu s MATLAB-om
- ponúka automatické práhovanie

Hlavným nedostatkom technológie ARToolKitPlus je, že od júna 2006 nie je aktualizovaný a teda zaostáva za ostatnými modifikáciami ARToolKit-u. Jeho nástupca Studierstube Tracker ponúka vyšší výkon pre osobné počítače rovnako ako i pre mobilné telefóny vďaka jeho naprogramovaniu od začiatku. Jeho hlavné výhody sú okrem vyššej rýchlosti i malé nároky na pamäť, prenositeľnosť, ľahká rozširiteľnosť a rôzne typy spracovávaných značiek [5].

❖ NyARToolKit

NyARToolKit predstavuje kompletný port ARToolKit-u napísaného výlučne v jazyku Java. Poskytuje menší výkon ako pôvodná verzia, je však úplne nezávislý na architektúre. Názov sa odvodil z mena jeho autora japonského inžiniera Nyatla v roku 2008. Napriek tomu, že domáci jazyk je Java, nástroj dokáže pracovať i s jazykom C# a s modifikáciou Javy určenej pre Android. NyARToolKit je distribuovaný ako voľne upravovateľný, nekomerčný produkt [16].

Následne si uvedieme časť funkcií, ktoré tento nástroj ponúka:

- jednoduché rozhranie pre tvorbu aplikácií
- prekrývanie 3D objektov na značkách
- viacplatformová video knižnica
- vstupné zdroje (USB, Firewire, prídavné karty, a iné)
- jednoduchá kalibrácia
- rýchle zobrazovanie založené na OpenGL
- podpora VRML
- modulárne API vytvorené v Jave
- kompletná zbierka vzoriek a utilít

❖ FLARToolKit

FLARToolKit (Flash-based Augmented Reality Toolkit) predstavuje port NyARToolKit-u a slúži pre prácu v prostredí FLASH. Bol vytvorený v roku 2008 japonskými vývojármi. Je verejne dostupný avšak s chýbajúcou dokumentáciou. Ako základný jazyk slúži Action Script 3, ktorý poskytuje oveľa menší výkon v porovnaní a verziou ARToolKitu pre Javu, alebo tou pôvodnou, pre jazyk C. Toto spomalenie znemožňuje v čase reálnu prácu s 3D modelmi. Keďže sa jedná o knižnicu odvodenú z NyARToolKit-u ponúka rovnaké funkcie ako sme spomenuli vyššie [17].

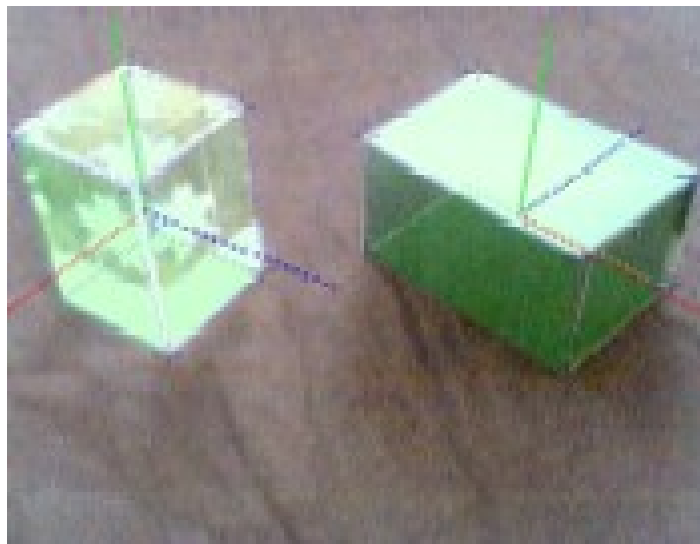
2.5. *Mixed Reality Toolkit*

2.5.1. O nástroji MRT

Mixed Reality Toolkit je knižnica určená na tvorbu obohatenej reality. Vývoj prebieha na UCL v Londýne [26].

Základná funkcionalita MRT sa od ARToolkit nelíši. Podporuje využitie značiek na určovanie relatívnej polohy a sledovanie pohybu.

System oproti ARToolkit-u umožňuje definovať objekty z reálneho sveta priamo počas behu aplikácie. Táto činnosť sa vykonáva priamo vo výstupnom videu a tým poskytuje možnosti ako zlepšiť pomocou spätnej väzby kvalitu prezentovanej obohatenej reality. V hornej časti najprv užívateľ jednoducho myšou opíše kde sa jednotlivé objekty nachádzajú, aký majú tvar a ako sú orientované (Obr. 22). Následne počítačom vygenerované objekty na tieto predmety reagujú, nie je ich cez ne vidno (Obr. 23). Môže medzi nimi dokonca dôjsť k interakcii (jednostrannej samozrejme) , napríklad odrážanie vygenerovaných objektov od reálnych predmetov.



Obr. 22 - Definovanie objektov počas behu.



Obr. 23 - Interakcia medzi virtuálnymi objektmi a reálnymi predmetmi.

V čom systém nad ARToolkit vyniká je podpora sledovania objektov. Ak počas behu programu užívateľ označí objekt, MRT sa snaží daný objekt sledovať podobne, ako by to bola značka.

MRT je určený pre operačný systém Windows, pričom podporovaný programovací jazyk je C++. Na vykresľovanie objektov využíva OpenGL spolu s rozšírením GLUT. Zároveň požaduje Microsoft DirectX9 SDK. Systém je dostupný pod licenciou GNU LGPL.

2.6. QR kód

Jedná sa o maticový obrazový kód (Obr. 24), ktorý bol vytvorený japonskou firmou Denso-Wave už v roku 1994. Skratka QR je odvodená od anglického „Quick Response,“ čím sa dáva najavo hlavný zámer tvorcov, a tým je rýchlo poskytnúť príjemcovi zakódovanú informáciu[18].

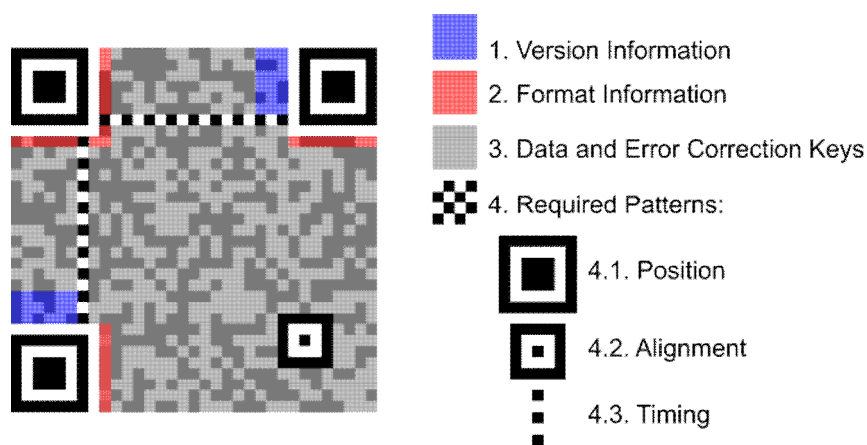


Obr. 24 - Príklad QR kódu – vytvorený pomocou generátoru od firmy Kaywa ⁸.

⁸ On-line generátor QR kódu - <http://qrcode.kaywa.com/>

Hlavnou motiváciou pre vývoj takéhoto obrazového kódovania bol kapacitný nedostatok klasického čiarového kódu, pretože pomocou neho sa dá zakódovať len niekoľko číselných údajov. Čiarový kód je obmedzený len na jeden rozmer, ale ak sa z grafického kódovania spraví mriežka, zväčší sa kapacita niekoľkonásobne.

Princíp spočíva v tom, ako môžete vidieť z Obr. 24, že celá matica je zložená z čiernych a bielych bodov, pomocou ktorých vie čítačka kódu dešifrovať dáta. Avšak zatiaľ čo pri čiarovom kóde nie je problém určiť pozíciu a jediná komplikácia môže nastať v tom či sa číta zľava alebo sprava, čo sa dá ošetriť špeciálnymi okrajovými značkami, pri maticovom kóde je situácia náročnejšia, a aj systém značenia pozície je komplikovanejší (Obr.25). Základom sú výrazné štvorce v rohoch, ale okrem nich sa objavujú značky rozloženia aj v iných častiach obrazca.



Obr. 25 - Štruktúra QR kódu

[http://commons.wikimedia.org/wiki/File:QR_Code_Structure_Example.svg].

Po obsahovej stránke sa skutočne jedná o pokrok oproti čiarovému kódu, ktorý dokázal v sebe uchovať približne len 20 čísiel. Kapacita QR kódu je v Tab. 1.

Čísla	Max. 7,089 znakov
Písmená a čísla	Max. 4,296 znakov
Binárny obsah	Max. 2,953 bajtov

Tab. 1 - Kapacita QR kódu

Uvedená kapacita je samozrejme len pri najväčšej verzii, ktorá má rozmer až 177x177 bodov. Vo väčšine prípadov však takýto rozmer, a teda ani kapacita, nie je potrebný. Každopádne je aj pri menšom rozmere kapacita dostatočná na to aby bolo možné zakódovať nielen samotnú informáciu, ale aj sprievodné dáta o verzii či formáte kódu (Obr. 25). Taktiež je možné aplikovať pri vytváraní výsledného obrazca aj rôzne procedúry, ktorými sa vo výsledku dajú detekovať a opravovať chyby pri poškodení obrazca.

2.6.1. Využitie QR kódu

Na základe vyššie uvedeného sa dá nájsť mnoho rôznorodých aplikácií QR kódu. Z hľadiska bežného použitia pre sprostredkovanie informácií na verejnosti sa ako najlogickejšie javí kódovanie textu. Avšak ak sa v samotnom texte použije špeciálna syntax, možno ukladať špecifické informácie pomerne jednoduchým spôsobom. Bolo štandardizovaných niekoľko typov a takýmto spôsobom sa dajú v QR kódoch ukladať informácie o URL adresách, kontaktoch, e-mailoch, či SMS alebo MMS správach [23], podľa ktoré môže čítacie zariadenie, zväčša mobilný telefón alebo iné prenosné zariadenie, vykonať konkrétnu operáciu. Napr. hneď načítať internetovú stránku.

Samozrejme sa dá QR kód využiť obdobne ako bežný čiarový kód na produkte. Avšak vo využití možnosti tohto kódu sa medze nekladú.

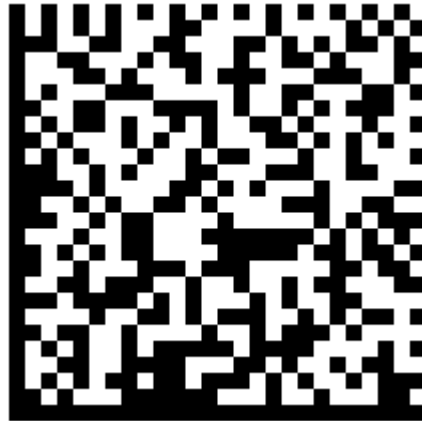
2.6.2. Alternatívy

QR kód je veľmi populárny v Japonsku, a v Ázii všeobecne, ale už sa postupne dostáva aj do zvyšku sveta. Hlavným problémom pri jeho zavádzaní aj v iných končinách sveta je fakt, že existujú rôzne konkurenčné obrazové kódy. V tejto časti sa pozrieme v krátkosti na niektoré z nich.

❖ DataMatrix

Dá sa považovať za jedného z najväčších konkurentov QR kódu a má silné zázemie najmä v Spojených štátoch amerických, ale aj v Európe. Už podľa príkladu na Obr. 26 je zjavná podoba medzi QR kódom a DataMatrix. Opäť sa jedná o plochu zloženú z bielych a čiernych bodov, ale systém na spoznanie polohy je odlišný. Namiesto výrazných štvorcov v rohoch sa využívajú okraje plochy, ktoré musia byť na dvoch susedných stranách tvorené súvislou čiarou a na zvyšných dvoch stranách striedaním bieleho a tmavého bodu [19].

Z obsahového hľadiska sa DataMatrix opäť podobá QR kódu. Taktiež možno pomocou neho zakódovať rôzne binárne či alfanumerické dáta, tiež aj štandardizované typy, ale maximálna udávaná kapacita je nižšia – 3116 čísel pri veľkosti mriežky 132x132 bodov.



Obr. 26 - Príklad DataMatrix – vytvorený pomocou generátoru od firmy Kaywa9.

❖ Pdf417

Tento typ kódu je podobne ako QR či DataMatrix taktiež ISO štandardom a je tiež pomerne rozšírený. Napríklad letecké spoločnosti ho využívajú na palubných lístkoch pre cestujúcich [21]. Avšak princíp fungovania je na rozdiel od predchádzajúci dvoch kódov o niečo iný. V prvom rade si môžete na Obr. 27 všimnúť určitú podobnosť s čiarovým kódom. Tá nie je len náhodná, pretože na rozpoznanie orientácie sa využívajú práve čiary po okrajoch. Vnútna dátová časť je zložená z niekoľkých blokov, ktoré sú už klasicky vyskladané z čiernych a bielych bodov. Dva z blokov sa využívajú na previazanie kódov medzi sebou, pretože je možné pomocou viacerých obrazcov zaznamenať obsiahlejšiu dátovú informáciu. Samotný dátový blok uchováva hodnotu z intervalu $\langle 0,929 \rangle$, ktorá sa následne reprezentuje na použiteľnú informáciu[20].

PDF417 je vždy obdĺžnik.



Powered By IDAutomation.com

Obr. 27 - Príklad kódu PDF417 – vytvorený pomocou generátoru od firmy IDAutomation¹⁰.

❖ PM kód

So zaujímavý grafickým kódom (skratka PM je odvodená od anglického Paper Memory) prišla spoločnosť Content Idea of Asia z Japonska¹¹, ktorá pôvodne dvojrozmerný QR kód obohatila o ďalšiu dimenziu pridaním farebnosti [22]. Takýmto spôsobom sa výrazne

⁹ On-line generátor DataMatrix kódu - <http://datamatrix.kaywa.com/>

¹⁰ On-line generátor kódu PDF417 - <http://www.bcgen.com/pdf417-barcode-creator.html>

¹¹ Stránka spoločnosti Content Idea of China Co., Ltd. - <http://ci-a.co.jp/>

zväčšuje kapacita. Pri jednoduchšej verzii s menším počtom bodov a s využitím 8-24 možno v jednom obraze uchovávať informáciu o veľkosti 0,6-1,8 MB. V extrémnom prípade, keď sa použije 256 farieb a veľký rozmer mriežky, možno zakódovať až 1,236 GB dát (Obr. 28).

Nevýhodou tohto kódu však je, že ešte nie je celosvetovo štandardizovaný a používaný. Avšak potenciál ukrýva veľký.



Obr. 28 - Príklad PM kódu ¹².

2.6.3. QR kód ako značka pre nástroje obohatej reality

Spomínali sme, že QR kód sa dá využívať viacerými spôsobmi a vďaka svojmu výraznému tvaru, a výzoru, už bol využitý aj ako značka pre aplikácie obohatej reality. Využitie QR kódu v takomto prípade v sebe v podstate ukrýva dve veci v jednom, pretože sa v ňom dajú uložiť informácie o vykresľovanom trojrozmernom objekte, a aj podľa kódu určuje orientácia modelu.

Jednu alternatívu predstavila ruská firma it-arts.com, ktorá vyvinula vlastný systém na sledovanie orientácie QR kódu v priestore. Celý ich výsledný produkt používa informácie z QR kódu ako ukazovateľ na miesto v internete, odkiaľ sa dajú stiahnuť konkrétne dáta, s ktorými aplikácia pracuje [24].

Veľmi podobný návrh bol predstavený aj v [25]. Opäť sa QR kód využíva dvojúčelovo, ale rozdiel oproti ruskej verzii je v tom, že na sledovanie pohybu použili platformu ARToolKit, pričom ju prispôbili tak aby sa zameriavala na štvorce v rohoch kódu, a nebol teda potrebný čierny rám okolo celého obrazca, tak ako u predchádzajúceho konceptu.

¹² Internetový príspevok o PM kóde - <http://symbloggy.blogspot.com/2007/01/gr-based-pm-code-best-3d-symbology-ever.html>

2.7. Kamery

V tejto časti sa budeme venovať stručnému porovnaniu niekoľkých kamerových zariadení a zhodnoteniu ich výhod-nevýhod, najmä z hľadiska nášho projektu.

2.7.1. Mobilný telefón

❖ Nokia 6120 classic

rozlíšenie: 2 Mpix (1600 x 1200 b.), LED blesk, záznam videí (MPEG4, 320 x 240 b.)

pamäť: 35 MB + microSD (do 2 GB), zdieľaná, RAM 64 MB (po zapnutí 20 MB)

lokálny prenos dát: Bluetooth 2.0 + EDR (A2DP), miniUSB 2.0, mass storage

OS Symbian v9.2

procesor: single CPU, ARM 11, 369 MHz

- Výhody: malé rozmery, veľká dostupnosť, jednoduchá obsluha, lokálny prenos dát
- Nevýhody: horšia kvalita snímania, nemožnosť práce s optikou, nemožnosť použitia statívu, menšia kapacita pamäte

❖ HTC Desire

rozlíšenie: 5 Mpix, LED blesk, autofokus, záznam videí (800 x 480 b., 25 fps)

pamäť: 576 MB RAM + 512 MB ROM + 4 GB microSDHC (do 32 GB)

lokálny prenos dát: Bluetooth 2.1 + EDR, kábel (microUSB), mass storage, Wi-Fi (802.11b/g),

OS Android 2.1

procesor: Qualcomm Snapdragon QSD8250, 1 GHz

- Výhody: malé rozmery, jednoduchá obsluha, vyššia kvalita snímania, OS Android, veľa možností lokálneho prenosu dát, vyššia kapacita pamäte
- Nevýhody: nemožnosť práce s optikou, nemožnosť použitia statívu, menšia dostupnosť

2.7.2. Webkamera

❖ Logitech QUICKCAM C160

rozlíšenie: 640 x 480 pixelov video: 640 x 480 pixelov

statické fotografie: až 1,3 megapixelu (softwarovo vylepšené)

- Výhody: malé rozmery, možnosť použitia statívu, veľká kapacita pamäte(hdd)
- Nevýhody: horšia kvalita snímanie, nemožnosť práce s optikou, k prevádzke je potrebný pc

2.7.3. Digitálny fotoaparát

❖ SONY DSC-W180

rozlíšenie: 10,1 megapixlov, optický zoom: 3x, zaostrovanie: AF, multi, center, predvoľby 0,5, 1, 3, 7 m a nekonečno, meranie expozície: multi, stred, bod

- Výhody: malé rozmery, možnosť použitia statívu, väčšia kapacita pamäte(pamäťová karta), možnosť práce s optikou, vyššia kvalita snímania
- Nevýhody: zložitejšia obsluha, menšia dostupnosť

2.7.4. Digitálna kamera

❖ Sony HDR-XR350VE

rozlíšenie: 2.65 Mpix (1.920 x 1.080i), režim FOTO: 7.1 MPX, Optický zoom 12 x, Kapacita HDD 160 GB

- Výhody: vysoká kvalita snímania, možnosť použitia statívu, veľká kapacita pamäte, veľa možností práce s optikou
- Nevýhody: zložitejšia obsluha, malá dostupnosť, väčšie rozmery

3. Špecifikácia riešenia

Obsahom tejto kapitoly je zadefinovanie kontextu a konceptu riešenia, pričom sú vymedzené aj jeho obmedzenia. Následne sú zadefinované požiadavky na riešenia a sú špecifikované procesy, ktoré sa v riešení vykonávajú. Taktiež sú identifikované prípady použitia.

3.1. Kontext riešenia

Keďže naša aplikácia bude orientovaná na interiérový dizajn, rozhodli sme sa pre vytvorenie systému, ktorý bude predstavovať interaktívnu nadstavbu pre katalóg nábytku. Nadstavba by spočívala v tom, že pre každú položku by bol vytvorený QR kód, ktorý by jednak obsahoval samotné údaje o objekte (nábytku) a zároveň by slúžil ako značka pre vykreslenie 3D modelu objektu pomocou obohatenej reality.

Na základe predošlej analýzy sme sa rozhodli pre vytvorenie aplikácie pre mobilné telefóny s operačným systémom Android. Mobilné telefóny v súčasnej dobe sú vybavené fotoaparátmi, ktorých parametre plne postačujú pre naše riešenie. Zároveň však ponúkajú aj veľa možností pripojiteľnosti k sieti internet, ktorá je pre našu aplikáciu nevyhnutná, pretože údaje o objektoch nie sú uložené v mobilnom telefóne, ale na vzdialenom servery.

Aplikáciu budeme vytvárať pomocou nástroja ARToolKit, keďže jeho nezávislosť na platforme nám umožňuje vytvorenie aplikácie pre systém Android.

3.2. Koncept riešenia

Vytváraný systém pozostáva z troch základných prvkov (Obr. 29):

- rozšírenia existujúceho katalógu nábytku
- aplikácia v mobilnom telefóne, slúžiaca na načítanie objektu a následné vykreslenie
- server, ktorý bude prijímať požiadavky od aplikácie na poskytnutie údajov o objekte

3.2.1. Rozšírenie katalógu nábytku

Rozšírenie klasického katalógu s nábytkom, by spočívalo v tom, že ku každému nábytku bude priradený QR kód, ktorý bude šifrovať URL adresu, kde sú uložené všetky údaje potrebné na vykreslenie objektu. Taktiež by boli poskytnutá značka (špeciálna strana v tlačenej katalógu alebo stiahnuteľný obrázok na vytlačenie), ktoré budú slúžiť ako spoločná značka na vykreslenie všetkých objektov, ktoré sme načítali z katalógu. To umožňuje zobrazovať nábytok na ľubovoľnom mieste, kde špeciálnu značku umiestnime.

3.2.2. Aplikácia v mobilnom telefóne (klientska aplikácia)

Aplikácia by mala vykonávať tri základné činnosti:

1. načítanie objektu pomocou údajov získaných z QR kódu
2. uloženie objektu do pamäte
3. vykreslenie uložených objektov
4. vytváranie virtuálnych scén zo získaných objektov

Pomocou kamery mobilného telefónu je nasnímaný QR kód, aplikácia tento kód dešifruje, čím získa adresu k súboru, kde sú uložené údaje, ktoré obsahujú priestorové údaje potrebné na vykreslenie objektu. Takto získané údaje sú ukladané do pamäte a používateľ si pri prezeraní interiéru vyberie nábytok z pamäte, ktorý chce vykresliť. Značky, na ktorých sa bude vykresľovať nábytok, budú reprezentované buď samotným QR kódom nábytku, ako náhľad, alebo spoločnou značkou zobrazenou v katalógu za cieľom vizualizácie umiestnenia nábytku v miestnosti.

3.2.3. Server

Server, na ktorom budú uložené všetky potrebné údaje o objektoch(nábytkoch). Keďže sa jedná o vzdialený server, aplikácia bude vyžadovať pripojenie na internet, aby mohla posielat' požiadavky na server a následne prijímať údaje.



Obr. 29- schéma konceptu riešenia

3.3. **Vymedzenie riešenia**

Riešenie s navrhnutým konceptom by mohlo prerásť do veľmi komplexného a robustného projektu, ale na jeho zrealizovanie by nebol reálne dostatočný čas. Preto sme identifikovali nasledovné vymedzenia:

- primárnym produktom je aplikácia v mobilnom telefóne a mala by umožňovať
 - dešifrovanie údajov z QR kódov
 - virtuálne zobrazenie 3D modelov
 - zobrazenie 3D modelov v obraze z kamery na špeciálnych značkách
 - zobrazenie 3D modelov na QR kóde
 - spravovanie získaných 3D modelov
 - vytváranie virtuálnych scén zo získaných 3D modelov a zobrazenie scén v obraze nasnímanom kamerou
 - komunikáciu so serverom a získavanie 3D modelov
- aplikácia v telefóne nebude umožňovať
 - vytváranie objednávok
- strana servera je pri našom riešení až sekundárnym problémom, preto nebudeme implementovať žiaden katalóg, ale určíme aké vlastnosti by bolo potrebné pridať do už existujúcich katalógov a vytvoríme jednoduchú web aplikáciu, ktorou by sa odsimulovalo fungovanie takejto nadstavby. Aplikácia by teda mala umožňovať:
 - príjem požiadaviek z mobilného telefónu
 - odosielanie dát do mobilného telefónu – 3D model a informácie o nábytku – získaných z databázy
 - generovanie QR kódov pre nábytok
- aplikácia na serveri nebude poskytovať žiadne komplexné grafické rozhranie, alebo len také, ktoré bude dostačujúce na prezentovanie riešenia

3.4. **Požiadavky**

3.4.1. **Hardvérové požiadavky**

Pre realizáciu a i aplikáciu bude potrebných niekoľko zariadení. Keďže projekt zahŕňa funkcie ako snímanie obrazu, komunikáciu s web serverom, renderovanie 3D objektov a iné. Pre splnenie nasledovných požiadaviek, si stručne predstavíme potrebný hardvér a jeho základné požiadavky.

❖ Web server

- osobný počítač alebo serverová stanica
- procesor 2000 MHz
- operačná pamäť 1GB

❖ Mobilný telefón

- procesor 1000 MHz
- operačná pamäť 512 MB
- podpora pre platformu Android
- vstavaná kamera

3.4.2. Softvérové požiadavky

Pre naplnenie špecifikácie je potrebné spolupracovať s rôznymi technológiami. Technológiami z oblasti web-serverov, čítania QR kódov, rovnako ako i s vývojovými prostrediami podporujúcimi platformu Android a v neposlednom rade 3D grafickými editormi. Po analýze dostupných možností sme sa dohodli na konkrétnych riešeniach. Náš výber následne stručne zdôvodníme vymenovaním ich vlastností v nasledujúcich kapitolách dokumentu.

❖ Apache server

Pre potreby projektu zabezpečiť komunikáciu medzi našou aplikáciou a okolitým svetom je potrebné vybudovať interakciu s kvalitným web serverom. Keďže systém ihneď po rozpoznaní značky pošle požiadavku na server kde obratom dostane potrebné informácie pre zobrazenie 3D objektu, musia byť tieto služby spoľahlivé a rýchle. Náš tím sa preto rozhodol pre technológiu Apache, ktorá spĺňa predchádzajúce uvedené požiadavky. V nasledujúcom odseku si túto technológiu stručne predstavíme [27].



Obr. 30 - Logo Apache server [□].

Apache http server je výsledkom kolaboratívnej práce za cieľom vytvoriť funkčne bohatého no najmä voľne dostupného http serveru. Bol vyvinutý v polovici deväťdesiatich rokov skupinou dobrovoľníkov. Pôvodne navrhnutý pre operačný systém UNIX, dnes však už bežne dostupný i pod platformou Windows. Podporuje protokol SSL, ktorý vďaka kryptovaniu zabezpečuje komunikáciu, rovnako ako podporuje i CGI a virtuálne domény. Doposiaľ boli predstavené jeho tri verzie. Posledná Apache 2.2 v roku 2006 priniesla nové, flexibilné moduly pre autentifikáciu ako i zdokonalenú prácu z prechodnými (angl. cache) pamäťami.

❖ ZXing

ZXing (Zebra Crossing) je voľne dostupný nástroj zameraný na dekódovanie 1D/2D čiarových kódov. Je implementovaný v jazyku Java a pracuje pod operačným systémom Android. Pre náš projekt je však zaujímavý najmä fakt, že táto technológia plnohodnotne pracuje so vstavaným fotoaparátom na mobilnom telefóne a prostredníctvom tohto zariadenia dokáže dekódovať čiarové kódy bez komunikácie so serverom [28]. Dokáže pracovať s nasledovnými formátmi kódov:

- UPC-A a UPC-E
- EAN-8 a EAN-13
- Code 39
- QR Code
- Data Matrix
- a mnohé iné.



Obr. 31 - Logo ZXing [28].

❖ Android

Pre vývoj nášho projektu je potrebné kvalitné vývojové prostredie podporujúce platformu Android. Android SDK [29] je predurčený pre túto platformu a preto sme sa rozhodli ho zvoliť ako náš základný vývojový prostriedok. Jedná sa o plnohodnotný nástroj obsahujúci debugger, knižnice, emulátor telefónu ako i kvalitne spracovanú dokumentáciu a ukážkové kódy. V súčasnosti je tento nástroj podporovaný architektúrou počítačov x86 a spolupracuje takmer s každým operačným systémom. Medzi jeho ďalšie požiadavky patrí i Java Development Kit, Apache Ant a Python verzie 2.2 a vyššie. Odporúčaným vývojovým prostredím je Eclipse obsahujúci ADT (Android Development Tools) podporujúcim jazyk Java pre platformu Android. Aplikácia by mala podporovať Android vo verzií 1.5 a vyššie.



Obr. 32 - Logo Android [29].

3.5. Procesy

V pre naše riešenie sme identifikovali nasledovné procesy, ktoré sú bližšie rozpracované v samostatných podkapitolách:

1. Načítanie modelu nábytku do telefónu
2. Správa modelov v telefóne
3. Správa virtuálnych scén
4. Úprava katalógu

Prvé tri procesy sa týkajú strany klienta a sú medzi sebou naviazané takým spôsobom, že procesy č.2 a 3 vychádzajú z výstupov procesu č.1.

Proces č.4 sa týka serverovej strany a jeho výstupy používa proces č.1.

Pri procesoch boli identifikovaní nasledovní účastníci:

- **Klient** – používateľ aplikácie v mobilnom telefóne
- **Klientska aplikácia** – aplikácia v mobilnom telefóne
- **Server** – stroj, na ktorom je prevádzkovaný katalóg a aplikácia na komunikáciu s klientskymi aplikáciami
- **Administrátor katalógu** – osoba, ktorá upravuje položky v katalógu nábytku

3.5.1. Načítanie modelu nábytku do telefónu

Proces načítania modelu nábytku do telefónu predstavuje postupnosť krokov, ktoré sa vykonávajú ak používateľ aplikácie v telefóne chce načítaním QR kódu získať model nábytku zo servera. Táto postupnosť krokov je znázornená na diagrame na Obr. 34.

Procesu predchádza prezeranie katalógu či už v tlačenej forme alebo cez internet.

Klient načíta pomocou telefónu QR kód, klientska aplikácia QR kód spracuje a ak je obsah QR kódu v priadku, tak si od servera vyžiada údaje. Ak je požiadavka korektná, tak server pošle požadované dáta, inak oznámi chybu. Podľa prijatých údajov klientska aplikácia buď zobrazí model alebo chybovú správu. Klient si model prezrie a ak sa mu páči tak si ho uloží alebo načíta iný QR kód, prípadne celý proces načítavania modelov skončí.

3.5.2. Správa modelov v telefóne

Proces správy modelov v telefóne predstavuje postupnosť krokov, ktoré sa vykonávajú ak sa organizujú stiahnuté 3D modely v telefóne (Obr. 33).

Klient si vyberie model, ten si následne prezrie ako virtuálny 3D model a následne sa rozhodne či si model ponechá alebo ho odstráni z pamäte telefónu.

3.5.3. Správa virtuálnych scén

Proces správy virtuálnych scén predstavuje postupnosť krokov, ktoré sa vykonávajú pri organizácii scén vytvorených zo stiahnutých 3D modelov, ale taktiež aj pri ich samotnej tvorbe.

Klient si na začiatku zvolí či chce upravovať už existujúcu scénu, ktorú vytvoril predtým alebo či chce vytvoriť úplne novú scénu. V prípade, že chce upravovať už existujúcu scénu, tak si jednu z nich zvolí a tú buď odstráni alebo si ju prezrie. Ak si ju chce prezrieť tak zachytí kamerou telefónu orientačné značky a scéna sa mu zobrazí. Obdobne zachytí značky aj keď vyvára novú scénu.

V tomto bode si klient môže zvoliť či chce scénu predsa len odstrániť, alebo či chce zvoliť inú, či chce ukončiť celý proces spravovania virtuálnych scén alebo či chce scénu upraviť.

V prípade úprav buď vloží do scény nový objekt z tých, ktoré si stiahol alebo sa upraví už vložený objekt. Ak upravuje vložený objekt, tak ho môže umiestňovať alebo meniť jeho vzhlád obdobne ako pri vložení nového objektu alebo môže objekt odstrániť zo scény.

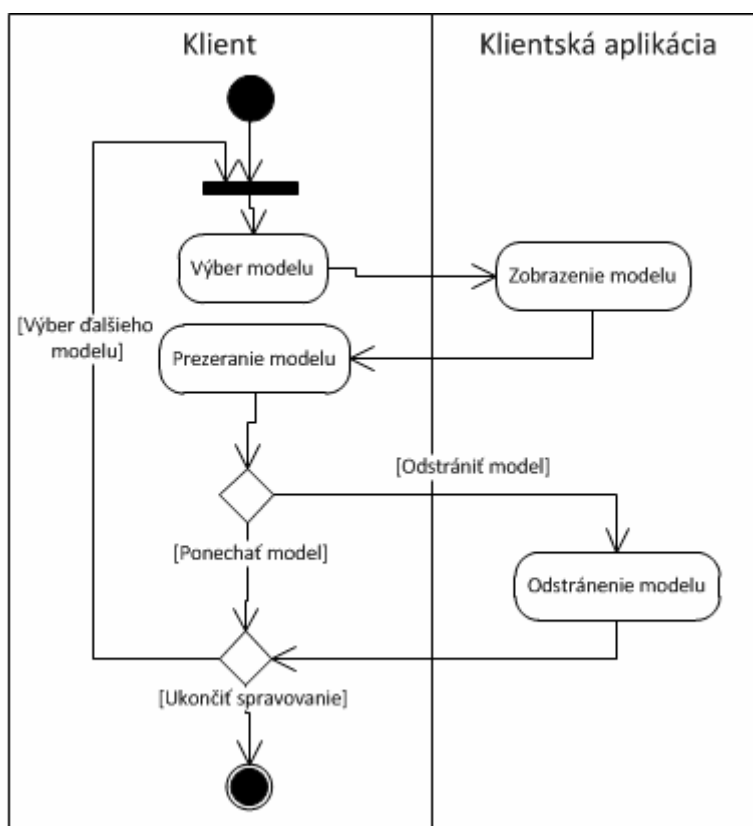
Ak už je editácia scény hotová, tak môže klient zmeny uložiť alebo ich zahodiť. V oboch prípadoch sa pokračuje rozhodovaním v bode či chce scénu odstrániť, upraviť, zvoliť si inú alebo celý proces práce so scénami ukončiť.

3.5.4. Úprava katalógu

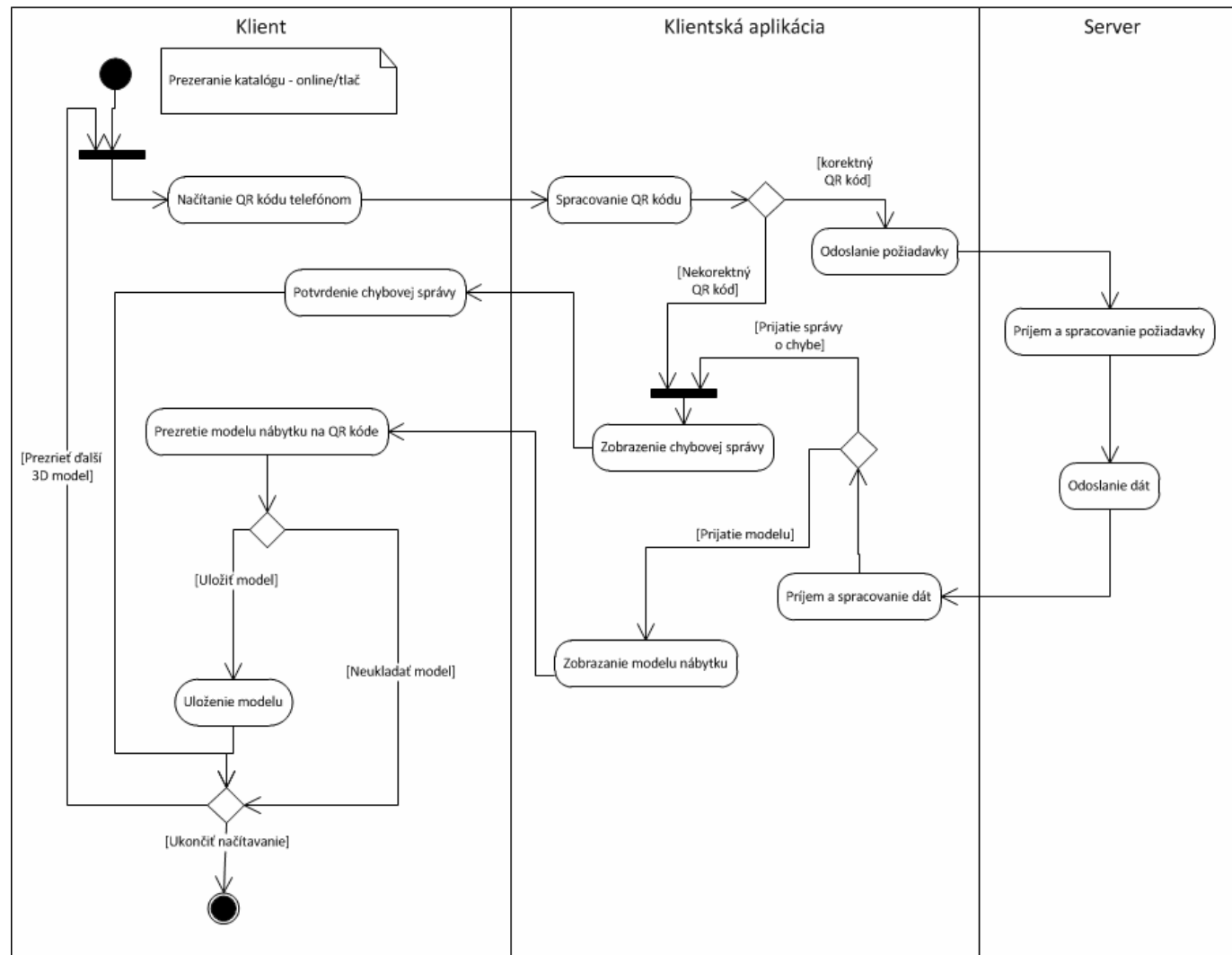
Proces úpravy katalógu predstavuje postupnosť krokov, ktoré vykonáva administrátor katalógu pri správe katalógových položiek (Obr. 36)

Aj keď naše riešenie nebude priamo umožňovať úpravu katalógu, tak je potrebné mať uvedený aj tento proces a vychádzať z neho. Je totiž potrebné ho mať na zreteli pri určovaní požiadaviek na rozšírenie existujúceho katalógu.

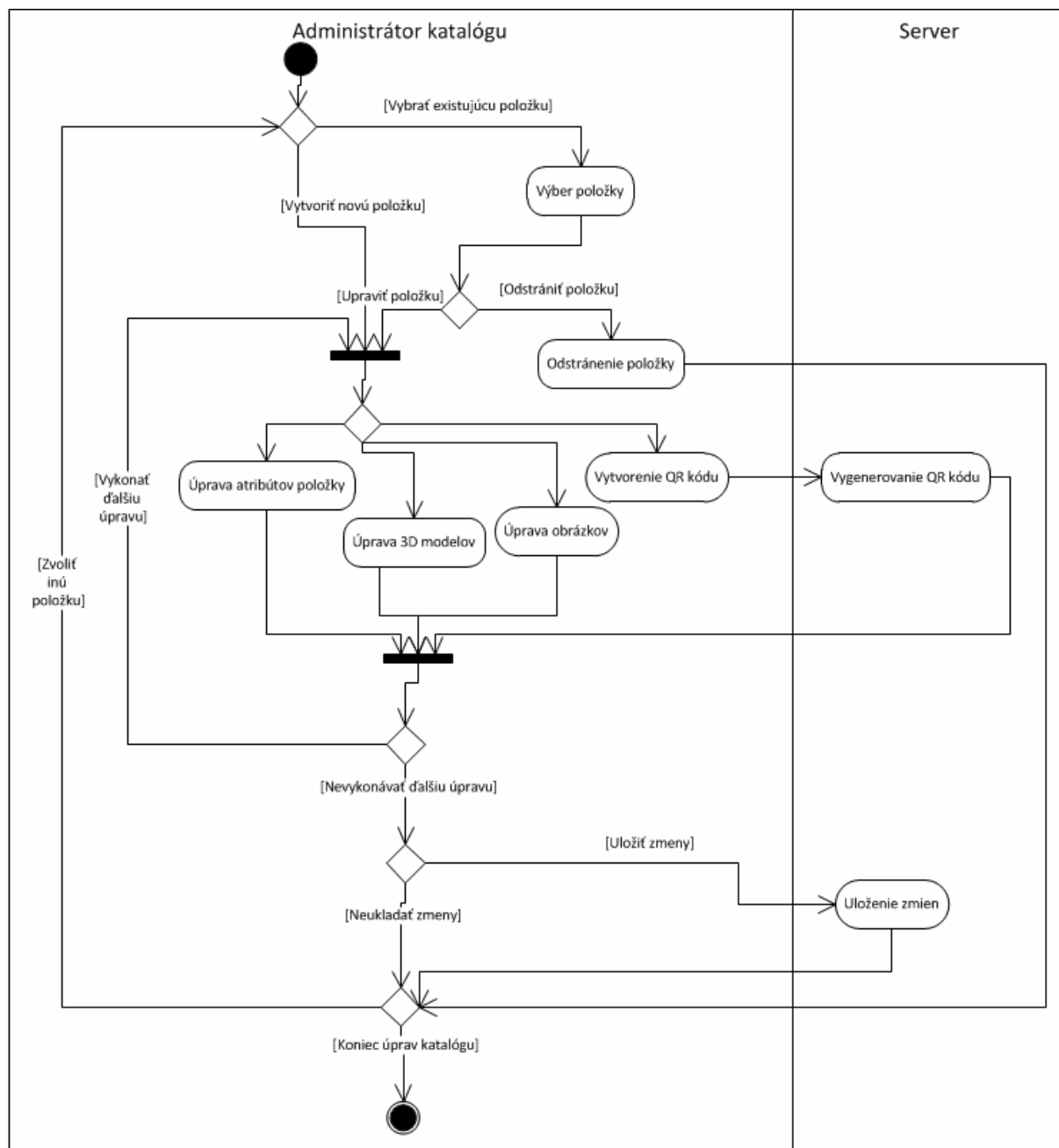
Administrátor si zvolí či chce vložiť novú položku alebo upraviť existujúcu. Ak si volí existujúcu, tak ju môže odstrániť alebo ju upravovať obdobne ako novú položku. Pod úpravu spadá zmena informácií o produkte, teda zmena atribútov položky, ďalej môže upravovať obrázky, ktoré sú k produktu priradené a pre potreby nášho riešenia môže obdobne spravovať 3D modely produktu a môže priradiť aj QR kód, ktorá sa vygeneruje systém. Po vykonaní všetkých úprav môže zmeny buď uložiť alebo neuložiť a následne buď ukončiť celý proces úpravy alebo môže znova upraviť inú položku, či prípadne vložiť novú.



Obr. 33 - Diagram procesu správa modelov v telefóne



Obr. 34 - Diagram procesu Načítanie modelu nábytku do telefónu



Obr. 36 - Úprava katalógu

3.6. Prípady použitia

Pre naše riešenie sme identifikovali týchto účastníkov v prípadoch použitia:

- **Klient** – používateľ aplikácie v mobilnom telefóne
- **Server** – stroj, na ktorom je prevádzkovaný katalóg a aplikácia na komunikáciu s klientskymi zariadeniami
- **Administrátor katalógu** – osoba, ktorá upravuje položky v katalógu nábytku

Pre účastníkov sme identifikovali nasledovné prípady použitia:

- **Načítaj model** – klient si chce pomocou informácie uloženej v QR kóde stiahnuť 3D model nábytku
- **Prezri model** – klient si chce prezrieť 3D model nábytku
- **Prezri info** – klient si chce prezrieť podrobné informácie o nábytku ako o produkte – teda cenu, rozmery a pod.
- **Odstráň model** – klient chce odstrániť z pamäte telefónu model, ktorý si predtým stiahol
- **Vytvor scénu** – klient si chce vytvoriť virtuálnu scénu z modelov, ktoré si stiahol, a chce scénu aplikovať na okolie, v ktorom sa nachádza
- **Uprav scénu** – klient chce upraviť scénu, objektov, ktoré sa v nej nachádzajú
- **Odstráň scénu** – klient chce odstrániť z pamäte telefónu scénu
- **Spracuj požiadavku** – server spracuje požiadavku klientskej aplikácie a odošle späť požadované údaje
- **Vygeneruj kód** – systém vygeneruje QR kód pre produkt, cez ktorý sa bude dať prístupovať ku 3D modelu a informáciám o produkte
- **Prirad' model** – administrátor katalógu priradí v databáze 3D model k produktu
- **Pridaj produkt** – administrátor katalógu chce pridať do katalógu nový produkt. Ak pridáva ak produktu aj 3D model, tak sa aktivuje prípad použitia „Prirad' model.“
- **Uprav produkt** - administrátor katalógu chce upraviť niektorý produkt v katalógu – zmena informácií, úprava priradených obrázkov či 3D modelov. Pri pridel'ovaní produktov sa aktivuje prípad použitia „Prirad' model.“
- **Odstráň produkt** – administrátor katalógu chce z katalógu odstrániť zvolený produkt – v prípade ak odstraňovaný produkt bol prístupný cez QR kód mal by si systém uchovať informáciu o tom, že tento produkt bol prítomný a vedel klienta informovať o tom, že tento produkt je už ďalej nedostupný, ak sa ho klient bude pokúšať získať
- **Prihlás sa** – prihlásenie administrátora do systému
- **Autentifikuj** – systém autentifikuje administrátora pri jeho prihlasovaní –rozširuje prípad použitia „Prihlás sa.“

Prípady použitia sú jednotlivo rozpracované v samostatných kapitolách, pozornosť je venovaná len tým prípadom použitia, ktoré sa týkajú Klienta a z prípadov použitia týkajúcich sa Servera len tie okrem „Autentifikuj.“ Ostatné sú uvedené preto, aby bol lepší prehľad o fungovaní ak sa používa aj reálny katalóg nábytku, ale keďže my katalóg vytvárať nebudeme, nie je potrebné aby boli tieto prípady použitia preberané príliš do hĺbky.

3.6.1. Načítaj model

Klient si chce pomocou informácie uloženej v QR kóde stiahnuť 3D model nábytku.

Vstupné podmienky: Klient má k dispozícii QR kód ktorý chce použiť na získanie modelu.

Výstup: Stiahnutý a zobrazený 3D model alebo upozornenie na chybu.

Účastníci: Klient, Server

Hlavný tok:

1. Klient zameria kameru telefónu na QR kód.
2. Klientska aplikácia dešifruje obsah QR kódu.
3. Klientska aplikácia sa spýta či má stiahnuť model nábytku. Ak klient odpovie kladne, tak model siahne, ak nie prípad použitia končí.
4. Klientska aplikácia vytvorí na základe obsahu QR kódu požiadavku pre server.
5. Klientska aplikácia zašle požiadavku na server.
6. Aktivuje sa prípad použitia *Spracuj požiadavku* a zo server sa stiahne model.
7. Stiahnutý model sa zobrazí na v obraze snímanom kamerou na mieste kde je QR kód.
8. Prípad použitia končí.

Alternatívne toky:

A1 Obsah QR kódu nie je korektný

Tok sa aktivuje v kroku 2 hlavného toku, ak dešifrovaný obsah QR kódu nie je určený pre túto aplikáciu.

1. Klientska aplikácia zobrazí správu o nevhodnom obsahu QR kódu.
2. Prípad použitia končí

A2 Zlyhala komunikácia so serverom

Tok sa aktivuje v kroku 5 alebo kroku 6 hlavného toku ak zlyhá komunikácia so serverom.

1. Klientska aplikácia zobrazí správu o chybe v komunikácii so serverom.
2. Prípad použitia končí

A3 Nekorektná požiadavka

Tok sa aktivuje v kroku 6 hlavného toku ak server informuje o tom, že zaslanej požiadavke sa nedá vyhovieť a model neexistuje.

1. Klientska aplikácia zobrazí správu o chybovej požiadavke.

2. Prípád použitia končí

3.6.2. Prezri model

Klient si chce prezrieť 3D model nábytku.

Vstupné podmienky: Uložené modely v pamäti telefónu alebo prístup ku katalógu s QR kódmi.

Výstup: Zobrazený 3D model nábytku alebo správa o chybe.

Účastníci: Klient

Hlavný tok:

1. Klient si v klientskej aplikácii zvolí prezeranie modelov
2. Klient si zvolí prezeranie modelov uložených v pamäti.
3. Klientska aplikácia zobrazí textový zoznam modelov.
4. Klient si zvolí model, ktorý si chce prezrieť.
5. Klientska aplikácia zobrazí 3D model. [Bod rozšírenia: *Prezrieť informácie*]
6. Klient si prezerá 3D model jeho rotovaním pohybom prstu po dotykovom displeji.
7. Klient sa môže vrátiť späť na zoznam modelov a pokračovať krokom 2 hlavného toku.
8. Prípád použitia končí.

Alternatívne toky:

A1 Prezretie nového modelu

Tok sa aktivuje v kroku 2 hlavného toku asi si klient chce prezrieť nový model nábytku.

1. Klient si zvolí stiahnutie nového modelu.
2. Aktivuje sa prípad použitia *Načítaj model*.
3. Ak sa zobrazí správa o chybe prípad použitia končí.
4. Klient si prezerá stiahnutý 3D model priamo nad QR kódom.
5. Ak klient zvolí možnosť uložiť model, tak klientska aplikácia model uloží do pamäte
6. Pokračuje sa krokom 2 hlavného toku alebo prípad použitia končí návratom do hlavného menu aplikácie.

A2 Pohyb medzi 3D modelmi

Tok sa aktivuje v kroku 5 hlavného toku, ak sa chce klient pohybovať medzi 3D modelmi a nie si ich vyberať cez zoznam.

1. Klient spraví prudký pohyb palcom zľava doprava alebo sprava doľava
2. Klientska aplikácia zobrazí model, ktorý je v zozname za, resp. pred aktuálnym modelom.
3. Pokračuje sa krokom 1 tohto toku alebo krokom 6 hlavného toku.

3.6.3. Prezri info

Klient si chce prezrieť podrobné informácie o nábytku ako o produkte. Prípád použitia je rozšírením prípadu použitia *Prezri model* v bode rozšírenia *Prezrieť informácie*.

Vstupné podmienky: Stiahnutý model nábytku.

Výstup: Zobrazovaný 3D model nábytku alebo správa o chybe.

Účastníci: Klient

Hlavný tok:

1. Klient si zvolí zobrazenie informácií o nábytku
2. Klientska aplikácia zobrazí tabuľku s informáciami, ktoré boli stiahnuté zo serveru.
3. Klient zvolí návrat späť na prezeranie modelu.
4. Prípád použitia končí.

3.6.4. Odstráň model

Klient chce odstrániť z pamäte telefónu model, ktorý si predtým stiahol.

Vstupné podmienky: Uložený model v pamäti telefónu.

Výstup: Model odstránený z pamäte.

Účastníci: Klient

Hlavný tok:

1. Klient si v klientskej aplikácii zvolí prezeranie modelov
2. Klient si zvolí prezeranie modelov uložených v pamäti.
3. Klientska aplikácia zobrazí textový zoznam modelov.
4. Klient si zvolí model, ktorý chce odstrániť.
5. Klient zvolí odstránenie modelu.
6. Klientska aplikácia si vyžiada potvrdenie odstránenia. Ak klient potvrdí, model bude odstránený.
7. Pokračuje sa krokom 3 alebo krokom 8.
8. Prípád použitia končí.

Alternatívne toky:

A1 Model je zahrnutý v scéne

Tok sa aktivuje v kroku 6 hlavného toku ak je odstraňovaný model zahrnutý do niektorej scény.

1. Klientska aplikácia zobrazí upozornenie o tom, že model je zahrnutý do scény a vypíše zoznam scén, v ktorých sa nachádza.

2. Pokračuje sa krokom 6 hlavného toku. Ak bude model odstránený, tak sa odstráni aj jeho výskyt v scénach

3.6.5. Vytvor scénu

Klient si chce vytvoriť virtuálnu scénu z modelov, ktoré si stiahol, a chce scénu aplikovať na okolie, v ktorom sa nachádza.

Vstupné podmienky: Uložený model v pamäti telefónu.

Výstup: Žiaden alebo scéna uložená v pamäti telefónu.

Účastníci: Klient

Hlavný tok:

1. Klient si zvolí vytvorenie novej scény zo uložených modelov nábytku
2. Klientska aplikácia vytvorí prázdnu scénu
3. Aktivuje sa prípad použitia *Uprav scénu* v alternatívnom toku *Konkrétna scéna*, pričom ako vstup sa použije prázdna scéna vytvorená v kroku 2.
4. Prípad použitia končí.

3.6.6. Uprav scénu

Klient chce upraviť scénu, objektov, ktoré sa v nej nachádzajú.

Vstupné podmienky: Existencia scén v pamäti telefónu alebo konkrétna scéna, s ktorou sa má pracovať

Výstup: Upravená scéna.

Účastníci: Klient

Hlavný tok:

1. Klient si zvolí upravovanie scén.
2. Klientska aplikácia zobrazí textový zoznam scén.
3. Klient si zvolí scénu, ktorú chce upraviť.
4. Klientska aplikácia sa prepne do režimu zachytávania obrazu kamerou.
5. Klient zachytí do obrazu značky, na ktoré sa má aplikovať scéna.
6. Klientska aplikácia dokreslí do obrazu objekty scény.
7. Klient si zvolí zobrazenie zoznamu objektov v scéne.
8. Klientska aplikácia zobrazí zoznam objektov.
9. Klient si zvolí, ktorý objekt chce upraviť.
10. Klientska aplikácia vyznačí objekt v scéne.
11. Klient si zvolí či chce objekt presunúť alebo otočiť
12. Klient vykonáva operáciu zvolenú v predošlom kroku pohybom prstu po obrazovke.

13. Pokračuje sa krokom 11 ak klient chce ešte vykonávať zmeny alebo sa pokračuje krokom 14.
14. Pokračuje 7 alebo prípad použitia končí

Alternatívne toky:

A1 Konkrétna scéna

Tok sa aktivuje v kroku 1 hlavného toku ak už je vopred určená scéna, s ktorou sa má pracovať.

1. Pokračuje sa krokom 4 hlavného toku

A2 Pridanie nového objektu

Tok sa aktivuje v kroku 9 hlavného toku ak chce klient pridať nový objekt do scény.

1. Klient si zvolí pridanie nového objektu do scény.
2. Aktivuje sa prípad použitia *Prezri model*.
3. Ak si klient zvolí vložiť prezeraný model, tak som model vloží do zoznamu objektov v scéne a pokračuje sa krokom 10 hlavného toku, pričom je vyznačený práve vložený objekt.
4. Ak si klient nevyberie objekt, pokračuje sa krokom 8 hlavného toku.

A3 Odstránenie objektu

Tok sa aktivuje v kroku 9 hlavného toku ak chce klient odstrániť objekt zo scény.

1. Klient si zvolí objekt, ktorý chce odstrániť zo scény.
2. Klient zvolí odstránenie objektu.
3. Klientska aplikácia si vyžiada potvrdenie odstránenia.
4. Ak klient potvrdí kladne, tak objekt bude odstránený zo scény
5. Pokračuje sa krokom 8 hlavného toku

A4 Uloženie zmien

Tok sa aktivuje v kroku 13 hlavného toku ak chce klient uložiť zmeny v scéne.

1. Klient si zvolí uložiť zmeny
2. Ak scéna nemá určené svoje meno, tak si klientska aplikácia vyžiada jeho zadanie.
3. Ak je meno korektné a jedinečné pokračuje sa krokom 4, ak nie tak sa pokračuje krokom 2
4. Klientska aplikácia uloží zmeny do pamäte.
5. Pokračuje sa krokom 13 hlavného toku.

3.6.7. Odstráň scénu

Klient chce odstrániť z pamäte telefónu scénu.

Vstupné podmienky: Uložená scéna v pamäti telefónu.

Výstup: Scéna odstránená z pamäte.

Účastníci: Klient

Hlavný tok:

1. Klient si v klientskej aplikácii zvolí prezeranie scén
2. Klientska aplikácia zobrazí textový zoznam scén.
3. Klient si zvolí scénu, ktorú chce odstrániť.
4. Klient zvolí odstránenie scény.
5. Klientska aplikácia si vyžiada potvrdenie odstránenia. Ak klient potvrdí, scéna bude odstránená.
6. Pokračuje sa krokom 3 alebo krokom 8.
7. Prípád použitia končí.

3.6.8. Spracuj požiadavku

Server spracuje požiadavku klientskej aplikácie a odošle späť požadované údaje.

Vstupné podmienky: Odoslaná požiadavka na odoslanie 3D modelu.

Výstup: Odoslaný model nábytku alebo chybová správa.

Účastníci: Server

Hlavný tok:

1. Server prijme požiadavku na odoslanie modelu klientskej aplikácii.
2. Server vyhľadá model podľa požiadavky
3. Server vytvorí súbor obsahujúci 3D model a informácie o nábytku.
4. Server odošle súbor klientskej aplikácii
5. Prípád použitia končí.

Alternatívne toky:

A1 Nekorektná požiadavka

Tok sa aktivuje v kroku 2 hlavného toku, ak server nevie nájsť požadovaný model.

1. Server nenašiel požadovaný model
2. Server odošle klientskej aplikácii správu o chybe.
3. Prípád použitia končí

3.6.9. Vygeneruj kód

Systém vygeneruje QR kód pre produkt, cez ktorý sa bude dať pristupovať ku 3D modelu a informáciám o produkte.

Vstupné podmienky: Zvolený produkt a jeho farebná modifikácia.

Výstup: Vygenerovaný obrázok s QR kódom uložený v databáza.

Účastníci: Server, Administrátor katalógu

Hlavný tok:

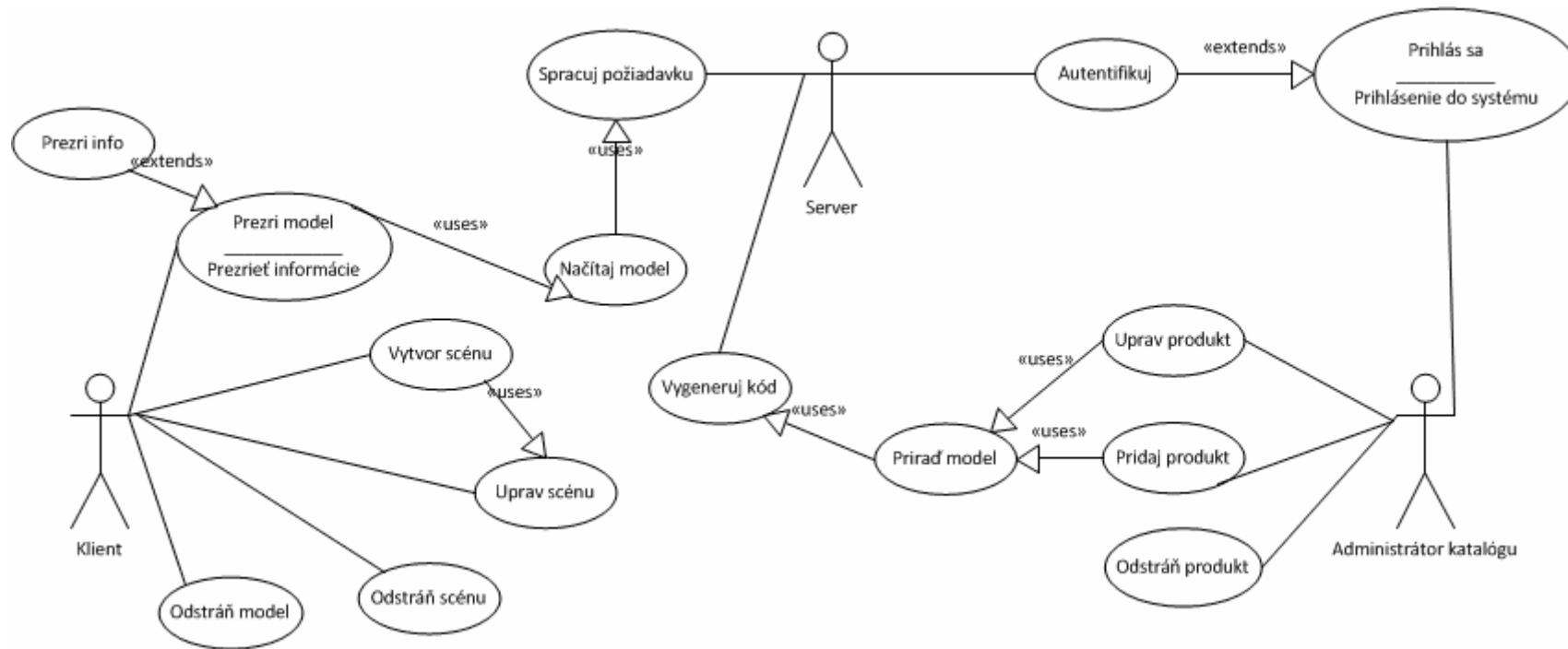
1. Administrátor katalógu zvolí vygenerovanie QR kódu pre zvolený produkt, prípadne aj pre jeho konkrétnu farebnú modifikáciu.
2. Administrátor katalógu určí veľkosť mriežky QR kódu.
3. Administrátor katalógu určí typ výstupného súboru (jpg, png, ...).
4. Server vygeneruje výstupný súbor na základe zadaných údajov.
5. Server zobrazí obrázok s QR kódom.
6. Server uloží obrázok do databázy.

Alternatívny tok:

A1 Kód už existuje

Tok sa aktivuje v kroku 4 hlavného toku ak pre zadaný produkt a jeho modifikáciu existuje QR kód s danou mriežkou a je uložený v súbore daného typu.

1. Server zobrazí informáciu o tom, že požadovaný QR kód už existuje.
2. Ak si administrátor katalógu zvolí možnosť zobrazit' súbor, tak mu ho server zobrazí.
3. Prípád použitia končí.



Obr. 37 - Diagram prípadov použitia

4. Návrh riešenia

Návrh nášho riešenia sa dá rozdeliť na 2 samostatné časti. Prvou je aplikácia, ktorá je prevádzkovaná na serveri a druhou je návrh klientskej aplikácie pre telefón.

4.1. Serverová aplikácia

Pri serverovej aplikácii sa v našom prípade ráta už s existenciou existujúceho katalógu a pre naše potreby prezentácie funkcionality aspoň jednoduchú aplikáciu, ktorá dokáže generovať QR a reagovať na požiadavky klientských aplikácií, zaslaním dát. Preto je potrebné určiť dátový model databázy, z ktorej sa dáta čerpajú, taktiež pravidlá generovania obsahu, ktorý sa zakóduje v QR kódoch a je potrebné určiť aj spôsob akým bude server komunikovať s klientskymi aplikáciami.

Naša, ukázková aplikácia bude vytvorená cez PHP5.

4.1.1. Dátový model

Databáza pre našu ukázkovú aplikáciu bude vytvorená na MySQL 5.1. Fyzický dátový model je uvedený na Obr. 38.

Je predpoklad, že dátový model už existujúceho katalógu s nábytkom je organizovaný podobne ako ten, ktorý sme my navrhli. A teda, že centrom je tabuľka so samotným produktom (Furniture), ktorý môže spadať pod viacero typov nábytku (Type) (typom sa myslí stôl, stolička, posteľ a pod.), a taktiež je priradený do viacerých kategórií (Category) (kategóriou sa myslí jedálenský nábytok, kancelársky nábytok, a pod.).

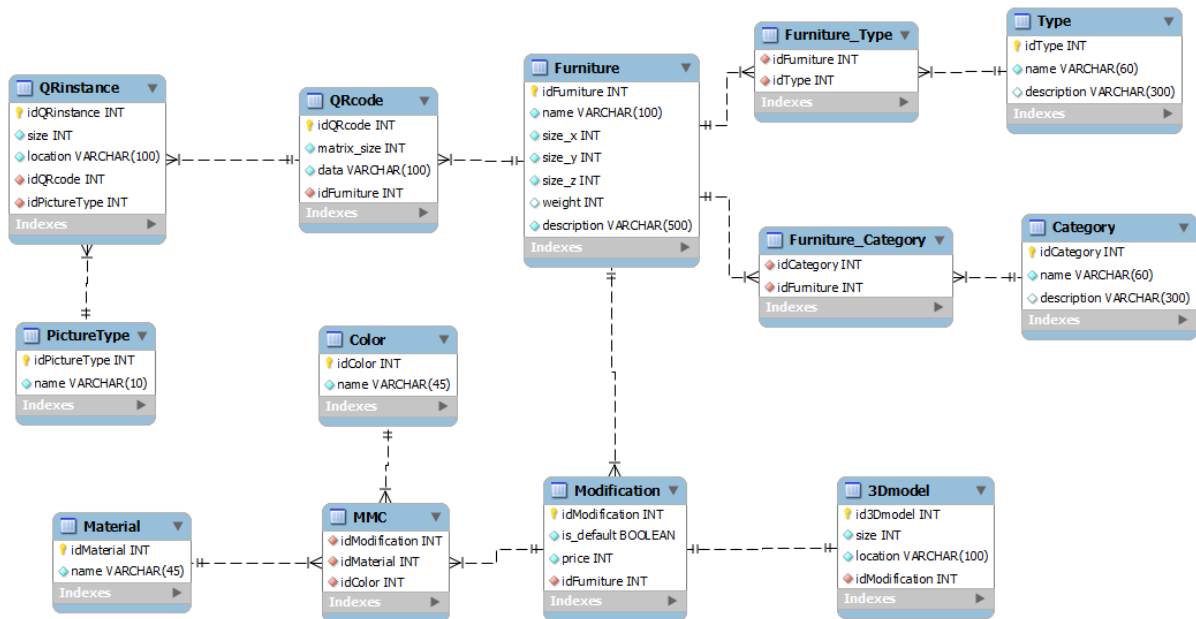
Ďalej, každý nábytok môže mať viacero farebných modifikácií, v ktorých sa ponúka – od modifikácie závisí aj konkrétna cena výrobku. K modifikácií môže byť priradených viacero kombinácií materiál-farba.

Rozšírením sú tabuľky QRcode, QRinstance, PictureType a 3Dmodel.

V tabuľke QRkód sa ukladajú informácie o QR kódoch, ktoré boli k produktu vygenerované. Je zaznamenaná veľkosť použitej mriežky a zakódovaný text. Každý QR kód môže byť vygenerovaný vo viacerých verziách (QRinstance), pričom verzie každá verzia má určený aj akého typu je výstupný obrázok (PictureType). Atribút size v QRinstance predstavuje veľkosť v kB, ktorá má obrázok a atribút location miesto v súborovom systéme kde je obrázok uložený.

3D model sa priraďuje ku konkrétnym farebným modifikáciám produktov. Atribút size je veľkosť súboru, v ktorom je model uložený, v kB. Atribút location predstavuje miesto v súborovom systéme kde je súbor s modelom uložený.

Atribút is_default v tabuľke Modification určuje či je daný záznam základnou verziou, ktorá sa má použiť. Je to potrebné ak sa v požiadavke klientskej aplikácie nešpecifikuje presne, s ktorou modifikáciou či modelom sa má pracovať.



Obr. 38 - Fyzický dátový model

4.1.2. Komunikácia s klientskou aplikáciou

Komunikácia bude prebiehať cez internet bežným protokolom HTTP. Klientska aplikácia si cez konkrétnu adresu s parametrami vyžiada dáta – balík informácií o produkte vo formáte XML a 3D model ako súbor typu „obj“. Celý balík bude jeden súbor zip archív s koncovkou „dat“. Server vydá dáta len keď identifikuje žiadateľa ako našu aplikáciu.

Adresa s požiadavkou je opísaná nižšie v kapitole pojednávajúcej o obsahu QR kódu.

Obsah odoslaného XML súboru vychádza z dátového modelu a bude nasledovný:

```
<product>
  <id>[jednoznačný identifikátor produktu]</id>
  <name>[názov produktu]</name>
  <dim_x>[dĺžka produktu]</dim_x>
  <dim_y>[šírka produktu]</dim_y>
  <dim_z>[hĺbka produktu]</dim_z>
  <weight>[hmotnosť produktu]</weight>
  <description>[slovný opis produktu]</description>
  <type>[typ výroku]</type> <!--typov výroku môže byť aj viac ako jeden
- typom sa myslí napr. stolička, posteľ, a pod. -->
  <category>[kategória výroku]</category> <!--kategórii, do ktorých
výrobok môže byť aj viac ako jedna - kategóriou sa myslí napr. kancelársky
nábytok, jedáľenský nábytok, a pod. -->
  <modification>
    <id>[jednoznačný identifikátor farebnej modifikácie]</id>
    <material>[názov materiálu]</material> <!--materiálov môže byť
viac než 1 -->
    <color>[názov použitej farby]</color> <!--farieb môže byť viac
než 1 -->
  </modification>
```

```

    <!--parametrom sa myslia ostatné udaje ako tie, ktoré sú uvedené
vyššie údaje ako rozmery -->
    <param>
        <name>[nejaký názov parametru]</name>
        <value>[hodnota]</value>
    </param>
    <!--parametrov môže byť 0 - n -->
    <modification>
    </modification>
    <!--sú uvedené všetky dostupné modifikácie aj keď model sa posiela len
k jednej z nich -->
</product>

```

4.1.3. Obsah QR kódu

Obsahov QR kódov, ktorý sa použijú je len jednoduchá URL adresa, ktorej tvar bude nasledovný:

```

http://[adresa_na_ktorej_beži_web_server]/[názov_skriptu_ktorý_spracováva_požiadavky]?i
d=[id_číslo_nábytku_z_tabulky_furniture]&modif=[id_farebnej_modifikácie_z_tabulky_mod
ification]

```

Parameter „modif“ môže byť nepovinný. Každý nábytok by mal mať totiž určenú svoju základnú modifikáciu, s ktorou sa pracuje.

4.2. Klientska aplikácia

Keďže sme narazili na viacero komplikácií pri implementácii, tak výsledný návrh klientskej aplikácie uvedieme až ďalšom semestri. Pribeh tvorby počas tohto semestra je uvedený v kapitole 5.

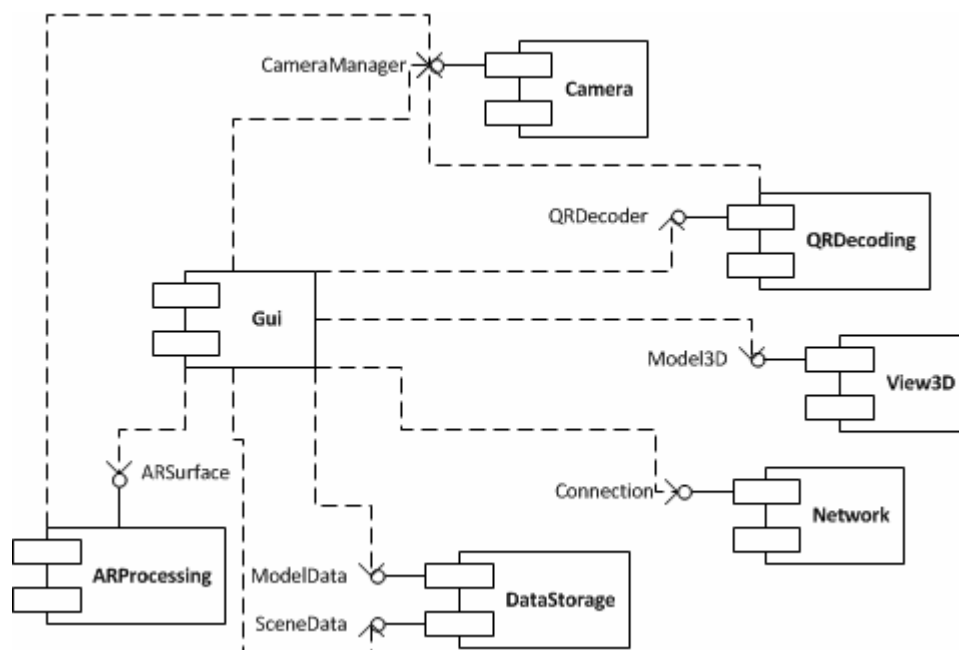
Zatiaľ sme stanovili komponenty, z ktorých bude výsledný aplikácia zložená aj s návrhom rozhraní, ktoré by mali poskytovať. Taktiež sme určili akým spôsobom sa budú ukladať dáta v telefóne.

4.2.1. Komponenty

Pre klientsku aplikáciu sme identifikovali nasledovných 7 komponentov:

- **Camera** – komponent, ktorý rieši prácu s integrovanou kamerou telefónu – jej sprístupnenie, uvoľnenie, atď. Prístup ku kamere sa bude riešiť cez rozhranie CameraManager
- **View3D** – komponent, ktorý umožňuje pristupovať k 3D modelom a upravovať ich vlastnosti pri zobrazovaní (rotácia, umiestnenie, atď.). Prístup k 3D modelu sa rieši cez rozhranie Model3D

- **Network** – komponent na pripojenie aplikácie do siete internet a komunikáciu so serverom. K pripojeniu sa pristupuje cez rozhranie Connection.
- **DataStorage** – komponent na správu súborov uložených v pamäti telefónu. K obsahu sa pristupuje cez rozhrania ModelData (dáta modelu) a SceneData (informácie o virtuálnych scénach)
- **QRDecoding** – komponent, ktorým sa dekoduje obsah QR kódu nasnímaného kamerou (prístup ku komponentu Camera). Prístup k dekodovaniu sa rieši cez rozhranie QRDecoder.
- **ARProcessing** – komponent, ktorý vkladá, pomocou špeciálnych značiek, do obrazu snímaného kamerou (prístup ku komponentu Camera) 3D modely. K upravenému obrazu z kamery sa pristupuje cez rozhranie ARSurface.
- **Gui** – komponent grafického rozhrania aplikácie, ktorý pristupuje ku všetkým vyššie spomínaným komponentom, pretože z nich všetkých čerpá informácie (minimálne tie o chybových hláškach.)



Obr. 39 - Diagram komponentov klientskej aplikácie

4.2.2. Ukladanie dát

Ukladanie dát v telefóne bude realizované v jednom spoločnom adresári pre všetky dáta aplikácie.

❖ Produkty

Každý stiahnutý model bude mať svoj vlastný priečinok nazvaný podľa jeho jednoznačného identifikátora. V tomto priečinku bude XML súbor s informáciami o produkte – stiahnutý

XML súbor bez vetvy o modifikácií – a každá modifikácia bude mať svoj adresár, v ktorom bude uložený stiahnutý model a XML súbor s informáciami o modifikácií – extrahovaná vetva modifikácie zo stiahnutého XML súboru.

❖ Scény

Scény budú uložené v jednom spoločnom XML súbore, ktorého štruktúra bude nasledovná:

```
<scenes>
  <scene>
    <name>[jednoznačný názov scény]</name>
    <object>
      <id>
objektu] <obj_id>[jednoznačný identifikátor vloženého
modifiákacie] </mod_id>
      </id>
      <position><!-- vzdialenosť od centrálného bodu scény -->
        <pos_x>[desatinné číslo]</pos_x>
        <pos_y>[desatinné číslo]</pos_y>
        <pos_z>[desatinné číslo]</pos_z>
      </position><!-- otočenie objektu po jednotlivých osiach -
hodnota v radiánoch -->
      <orientation>
        <rot_x>[desatinné číslo]</rot_x>
        <rot_y>[desatinné číslo]</rot_y>
        <rot_z>[desatinné číslo]</rot_z>
      </orientation>
      <scale>[násobok pôvodnej veľkosti objektu]</scale>
    </object>
  </scene>
</scenes>
```

5. Implementácia – zimný semester

Obsahom tejto kapitoly je informovanie o výstupe, ktorý sa nám podarilo vytvoriť počas zimného semestra.

Cieľom bolo vyhotoviť prototyp a našim pôvodným plánom bolo vytvoriť aplikáciu, ktorá dokáže načítať obsah QR kódu a podľa tohto obsahu vykresliť pomocou AR metód konkrétne stanovený jednoduchý objekt. Vyskytlo sa však viacero komplikácií, ktorým sme museli čeliť, a tak je výstupom súbor čiastkových poznatkov z jednotlivých oblastí, ktoré bude potrebné v ďalšej fáze projektu zlúčiť do jedného celku. Každá časť, ktorej sme sa venovali je opísaná v samostatnej podkapitole, pričom sú uvedené naše poznatky, zistené problémy a aj ďalšie smerovanie.

5.1. Vývojové prostredie a potrebné pluginy

Výber vývojového prostredia súvisel priamo s výberom platformy Android. Keďže oficiálne stránky platformy Android pre vývojárov odporúčajú vývojové prostredie Eclipse a členovia tímu majú s týmto prostredím skúsenosti, výber prostredia bol jednoznačný.

Na vývoj aplikácií pre Android bolo potrebné stiahnuť a nainštalovať Android SDK a ADT.

Android SDK je vývojársky balíček obsahujúci JAVA triedy potrebný na vývoj aplikácií pre Eclipse. Súčasťou SDK je aj AVD, čo je nástroj na vytváranie a správu virtuálnych strojov so systémom Android.

ADT je plugin do Eclipse a rozširuje možnosti vývojového prostredia za cieľom umožniť vyvíjať aplikácie pre Android jednoduchšie a efektívnejšie.

5.1.1. Inštalácia

Je predpokladaný výskyt Javy na danom počítači.

Prvým krokom inštalácie je stiahnutie vhodnej verzie vývojového prostredia Eclipse. V čase písania je to verzia 3.4 a novšia. Inštalácia Eclipse je jednoduchá, je potrebné stiahnuť daný archív z oficiálnej stránky Eclipse a rozbaľiť jeho obsah do vhodného adresára. Okrem toho nie je potrebná žiadna inštalácia.

Následne je potrebné stiahnuť Android SDK. Inštalácia vyžaduje stiahnutie štartovacieho balíčka SDK zo stránky developer.android.com a následnú inštaláciu. V prípade platformy Windows je tento proces jednoduchý, keďže získame spustiteľný Windows inštalačný súbor.

Posledným krokom je prepojenie Eclipse a Android SDK. To sa deje pomocou ADT, čo je plugin pre Eclipse. Proces je takmer plne automatický, keďže využíva „Update Manager“ prostredia Eclipse. Presný postup aj s vysvetlením je uvedený na stránke Androidu pre vývojárov, v dokumentácii uvádzame len samotný postup prevzatý z danej stránky.

Eclipse 3.5 (Galileo) a 3.6 (Helios)	Eclipse 3.4 (Ganymede)
<ol style="list-style-type: none"> 1. Spustite Eclipse, vyberte si v ponuke Help > Install New Software... 2. Kliknite na Add, v pravom hornom rohu. Do poľa <i>Name</i> zadajte „ADT Plugin“ a do poľa <i>Location</i> „https://dl-ssl.google.com/android/eclipse“. Kliknite na OK. 3. V dialógovom okne Available Software zaškrtnite „Developer Tools“ a kliknite na Next. 4. V ďalšom dialógu bude zverejnený zoznam nástrojov, ktoré budú stiahnuté. Kliknite na Next. 5. Prečítajte a akceptujte licenčné podmienky, kliknite na Finish. 6. Po skončení inštalácie reštartujte prostredie Eclipse. 	<ol style="list-style-type: none"> 1. Spustite Eclipse, vyberte si v ponuke Help > Install New Software... V dialógu ktorý sa objaví vyberte záložku Available Software. 2. Vyberte možnosť Add Site. 3. Do poľa <i>Location</i> zadajte: „https://dl-ssl.google.com/android/eclipse/“ 4. Kliknite na OK. 5. Zobrazí sa vám zoznam ponúkaných pluginov zoradených podľa URL. Vyhľadajte vyššie uvedený URL a zaškrtnite „Developer Tools“. Kliknite na Ok 6. V nasledujúcom okne budú zobrazené všetky Vami vybrané pluginy. Kliknite na Next. 7. Prečítajte a akceptujte licenčné podmienky, kliknite na Finish. 8. Po skončení inštalácie reštartuje prostredie Eclipse.

Zdroj: <http://developer.android.com/sdk/eclipse-adt.html#installing>

5.1.2. Podpora kamery v emulátore

Počas testovania jednotlivých častí navrhovaného sa objavil problém s kamerou. V súčasnosti emulátor nepodporuje prepojenie s externou kamerou na testovanie aplikácií, ktoré kameru využívajú. Takéto testovanie vyžaduje využitie hardvérového telefónu s kamerou. Pri štúdiu bolo objavené riešenie pomenované WebcamBroadcaster. Princípom bolo spustenie kamery pripojenej k počítaču a zaznamenávanie obrazu pomocou WebcamBroadcastera. Následne bol nasnímaný obraz poslaný cez TCP/IP Android emulátoru. Pôvodne sme predpokladali možnosť využitia tohto postupu na simulovanie kamery v Androide a testovanie vyvíjanej aplikácie. Toto riešenie sa však ukázalo ako neschodné a hrozil vznik problému s testovaním a ladením aplikácie z dôvodu, že by nebola dostupná platforma, na ktorej by sa dala skúšať potrebná funkcionálnosť.

5.1.3. Pokus s virtuálnym počítačom

Po neúspechu s WebcamBroadcasterom sme skúsili ešte jednu možnosť ako sprevádzkovať kameru pripojenú k počítaču na virtuálnom počítači s operačným systémom Android

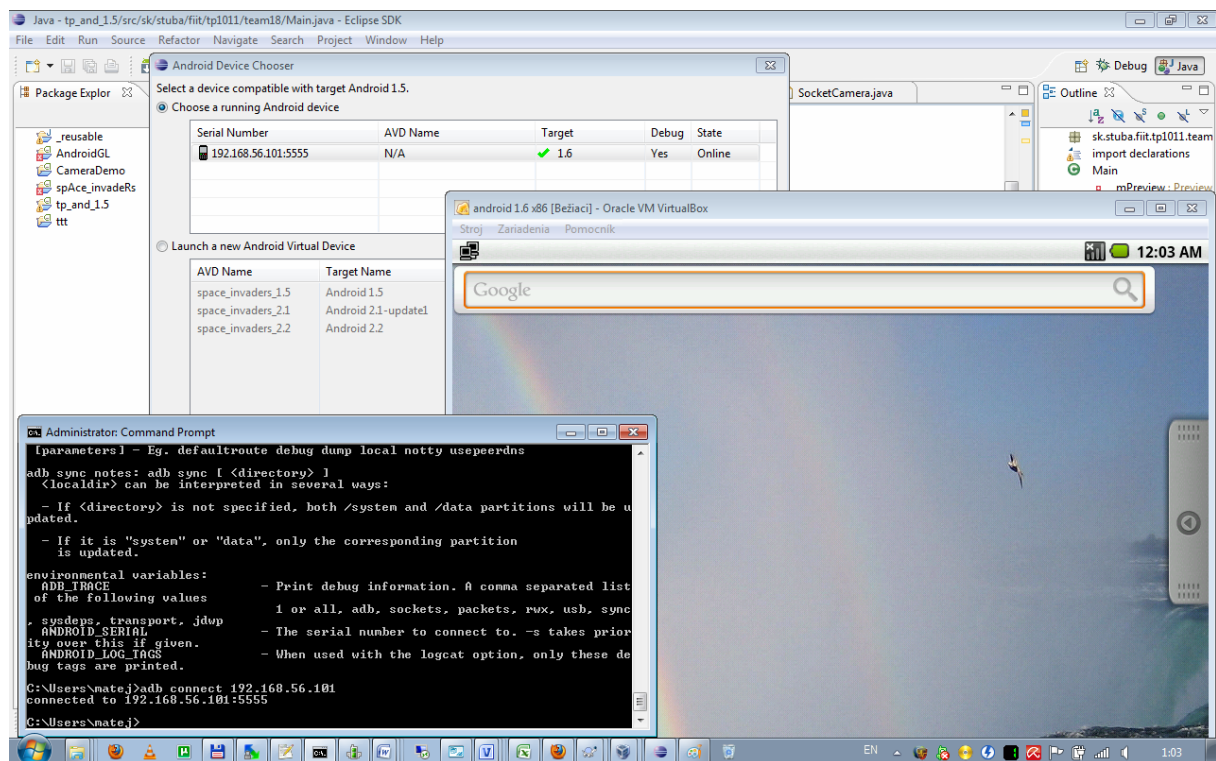
Základom bol Android 1.6 preportovaný na x86 architektúry, ktorý je dostupný z <http://www.androidx86.org/>.

Nainštalovali sme tento operačný systém na virtuálny stroj cez aplikáciu VirtualBox <http://www.virtualbox.org/>.

Postupovali sme podľa odporúčaní pri spúšťaní tohto systému cez VirtualBox, ale niektoré vlastnosti sme si zvolili sami. Parametre virtuálneho stroja boli nasledovné:

- Typ OS: Linux 2.6
- RAM: 512 MB
- Video pamäť: 8MB – bez podpory 2D alebo 3D akcelerácie
- Audio: zrušené – nepotrebujeme pracovať so zvukom
- Sieť: Host-only Adapter – aby sa dalo stroj použiť na debugovanie
- Pevný disk: 8GB

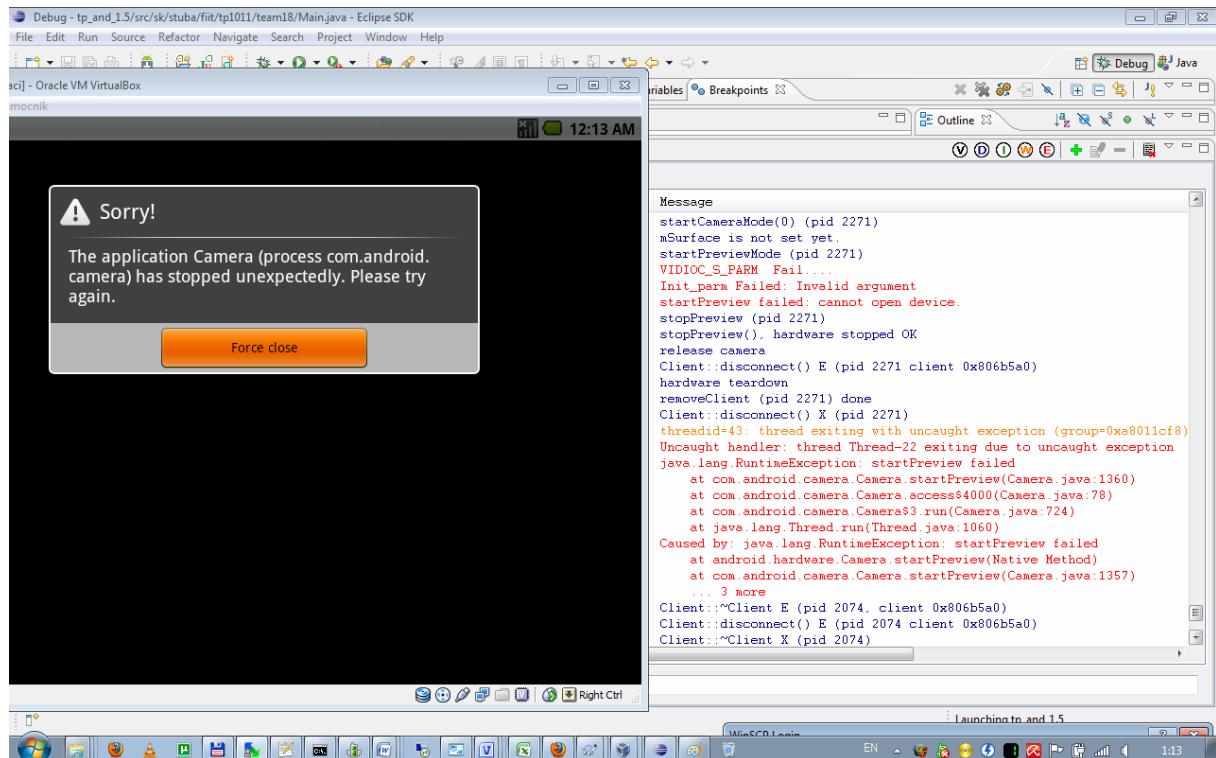
Všetko fungovalo dobre, dokonca systém mal rýchlejšiu odozvu pri ovládaní ako oficiálny emulátor. Ak sa nastavilo sieťové pripojenie pre virtuálny stroj na typ Host-Only a na hostovskom operačnom systéme sa spustila cez príkazový riadok aplikácia adb tak, aby počúvala na IP adrese virtuálneho stroja, dal sa tento virtuálny stroj použiť taktiež priamo na debugovanie cez Eclipse podobne ako reálne pripojený fyzický telefón s Androidom (Obr.40)



Obr. 40.: Pripojenie virtuálneho počítača s Android x86 operačným systémom s hostovským počítačom cez adb.

Problém sa však opäť vyskytol pri práci s kamerou. Nepodarilo sa podľa dostupných informácií (dokumentácia k Androidu x86 je v tomto smere slabá) pripojiť a sfunkčeniť webkameru hostovského počítača (notebook so vstavanou kamerou) a ani externú webkameru, pripojenú cez USB. Kamera nefungovala ani po viacerých nastaveniach v jadre Linuxu, na ktorom je postavený Android, a ani vtedy keď bola kamera priamo pripojená na virtuálny počítač, tak že ani hostovský operačný systém ju neregistroval.

Reakciu na spustenie aplikácie na prácu s kamerou, ktorá je základným vybavením systému Android, spolu s výstupom na konzole môžete vidieť na Obrázku 41.



Obr. 41.: Chybové správanie sa aplikácie na prácu s kamerou na virtuálnom počítači s Android x86.

Problémy s kamerou v emulovanom Androide sa nakoniec podarilo vyriešiť, keď si viacero členov tímu kúpilo mobilný telefón založený na platforme Android (konkrétne 4 zo 6 členov vlastní telefóny s OS Android). Toto prebehlo viac-menej nezávisle od existujúceho projektu. Pozitívne však je, že sa bude môcť pri ďalšom postupe testovať priamo na fyzických strojoch.

5.2. Grafické používateľské rozhranie pre Android

Pre implementovanie aplikačného systému bolo potrebné oboznámiť sa s tvorbou grafického používateľského rozhrania pre Android. A to z toho dôvodu, že grafické používateľské rozhranie tvorí základ každej aplikácie. Ak ide o aplikácie pre platformu Android, pri vytváraní grafického rozhrania treba rátať s tým, že zariadenia, na ktorých bude aplikácia používaná majú štandardne dotykové displeje. Tento fakt je nutné brať do úvahy pri vytváraní grafického používateľského rozhrania.

Na implementáciu grafického rozhrania, ako aj celého projektu používame vývojové prostredie Eclipse.

5.2.1. Základné grafické používateľské rozhranie

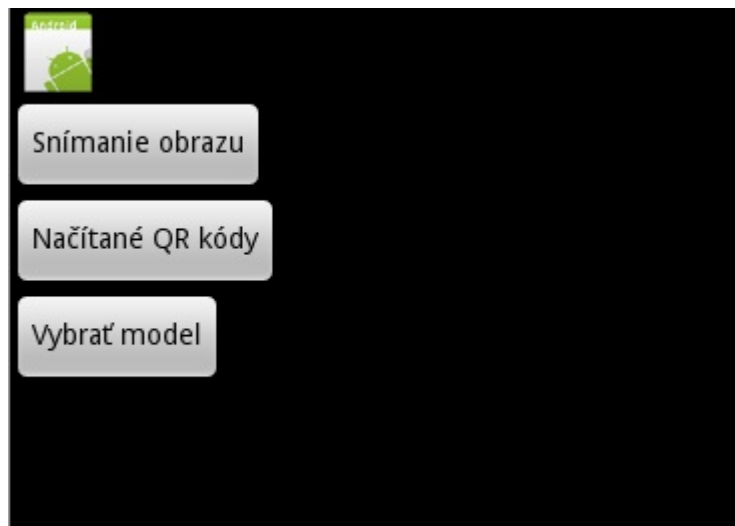
Vytvorili sme základné grafické používateľské rozhranie pre náš projekt. Toto grafické rozhranie sa zobrazí používateľovi po spustení aplikácie. Na obrazovke sa používateľovi zobrazia na vyber 3 možnosti vo forme tlačidiel.

Prvé tlačidlo s názvom snímanie obrazu reprezentuje spustenie kamery mobilného zariadenia a následný začiatok snímania QR kódu z obrazu získaného z kamery. Po zvolení tohto tlačidla bude na obrazovke zobrazovaný snímaný obraz z kamery.

Druhé tlačidlo nazvané načítané QR kódy umožní zobrazit' na obrazovke telefónu už načítané QR kódy. Ak neboli žiadne QR kódy načítané, toto tlačidlo nemá žiadnu funkcionality.

Tretie tlačidlo s názvom vybrať model slúži na výber modelu na vizualizáciu. Význam tohto tlačidla je, ak má používateľ v aplikácii načítané už nejaké modely, tak si vyberie zo zoznamu modelov model, ktorý si želá vizualizovať.

Hlavné menu grafického používateľského rozhrania aplikácie je ukázané na obrázku 42.



Obr. 42.: Hlavné menu aplikácie.

5.3. Prototyp pre AR

Pri implementácii prototypu reprezentujúceho aspoň základné črty budúceho projektu sme vychádzali s knižnicou ARToolKit. Keďže je to základná knižnica, z ktorej sú odvodené všetky ostatné pre iné platformy, bolo potrebné spoznať jej dôležité funkcie. Jednalo sa predovšetkým o porozumenie ako je schopná knižnica načítať daný obraz, spracovať ho a neskôr do neho vložiť 3D objekt.

Podstatná časť nášho projektu je tvorená pomocou mobilného telefónu s platformou Android. Aby sme boli schopní načítavať značky a následne vykresľovať 3D modely museli sme tomu prispôbiť i používanú knižnicu. Knižnica ARToolKit nie je určená pre mobilné telefóny a tak sme siahli po jej upravenej verzii a síce NyARToolKit.

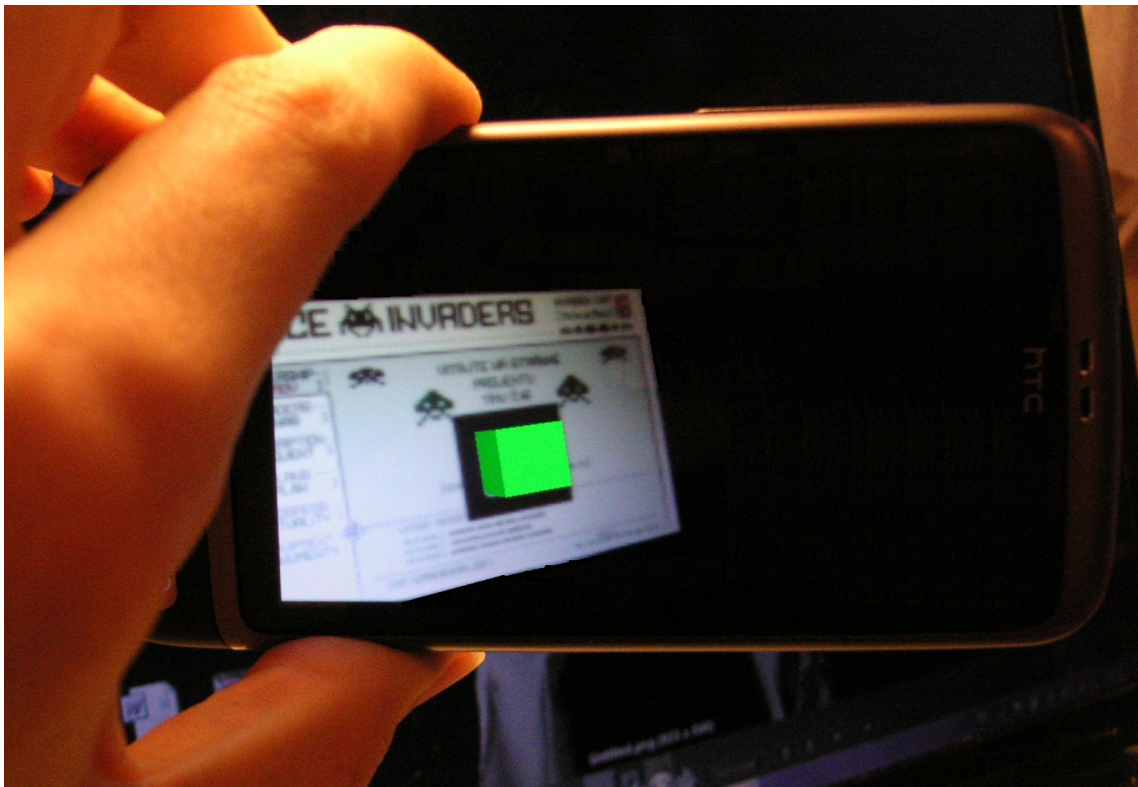
5.3.1. NyARToolKit

Táto knižnica umožňovala chod pod platformou Android a bola vytvorená v jazyku Java. Táto skutočnosť bola pre nás kľúčová, keďže väčšina tímu je zvyknutá robiť práve v tomto jazyku. Jej výraznou nevýhodou je však skutočnosť, že bola vytvorená v Japonsku a preto väčšina materiálov s ňou spojená bola v tomto jazyku. Ale nedali sme sa odradiť a stiahli sme si túto knižnicu. Obsahovala v sebe zároveň i ukážku čo by v konečnej miere predstavovalo dobrý štart pre vytvorenie nášho prototypu. Avšak po niekoľko dnovom trápení sa a ladenia knižnice sa nám ju nepodarilo úspešne spustiť na mobilnom telefóne. Dosiahli sme snímanie obrazu, no žiadne vykresľovanie 3D objektov. Príčinu neúspechu sa nám nepodarilo zistiť pretože kompilátor a ani samotný mobilný telefón neprotestoval. Každopádne získali sme aspoň nové skúsenosti a prehĺbili motiváciu nájsť vhodnú knižnicu pre náš projekt a spustiť prvé úspešné riešenie.

5.3.2. AndAR

Projekt AndAR predstavoval voľne šíriteľnú verziu knižnice ARToolKit, špeciálne pre platformu Android. Ponúkal dokumentáciu v anglickom jazyku spolu s názornými ukážkami využitia tejto knižnice. Po neúspechu s knižnicou NyARToolKit sme sa rozhodli ho vyskúšať a hneď pri prvej kompilácii sa nám podarilo tento projekt spustiť a vykresliť 3D kocku nad zadanou značkou. Previedli sme drobné úpravy a experimenty, ktoré iba potvrdili že práve AndAR je tá správna cesta. Výsledok našej snahy môžete vidieť na obrázku č.43 pod textom.

Tým sme sa dostali k bodu kedy časť projektu realizujúca odchyťovanie obrazu a vykresľovanie 3D modelov bola pripravené na prepojenie s ostatnými časťami akými sú knižnica ZXing pre prácu s čiarovými kódmi alebo samotné GUI budúcej aplikácie.



Obr. 43.: Názorná ukážka nášho prototypu.

5.3.3. Budúce smerovanie

V budúcnosti čaká náš projekt ešte veľa práce. Stojí pred nami úloha obohatiť našu prototypovú aplikáciu o vlastnosť aby dokázala spolupracovať s čiarovými kódmi. Rovnako náročne bude vytvoriť i používateľovi jednoduché grafické rozhranie s prepojením na el. obchod. V rámci samotnej knižnice AndAR bude potrebné preniesť sa cez vykresľovanie geometrických útvarov a dosadiť namiesto nich 3D modely. Preto odhodlanie s akým sa ku projektu postavíme v letnom semestri bude jeho úspech kľúčový.

5.4. Prototyp aplikácie na spracovanie QR kódu

Táto kapitola sa venuje tej časti prototypovania, v ktorej sme sa venovali spracovaniu obrazu z kamery zariadenia a jeho spracovaniu, teda najmä rozpoznaní prípadných QR kódov. Prototyp týchto funkcií bol vytváraný ako samostatná aplikácia pre systém Android, nezávisle od ostatných prototypov v tomto projekte.

5.4.1. Opis prototypu

Na rozpoznanie QR kódov je, aj pre platformu Android, možné použiť knižnicu ZXing, opísanú v samostatnej kapitole tohto dokumentu. Predpokladom na úspešné rozpoznanie QR kódov je preto zahrnutie tejto knižnice do projektu a schopnosť pristupovať k dátam z kamery a spracovávať ich.

5.4.2. Prostredie a testovanie

Prototypová aplikácia je vytvorená v prostredí Eclipse 3.6, s využitím balíka Android SDK, ako „Android Project“. Keďže emulátor nepodporuje simuláciu využitia kamery a tento problém nedokázalo vyriešiť ani použitie virtuálneho zariadenia, debuggovanie a testovanie aplikácie sa musí uskutočňovať priamo na zariadení so systémom Android.

5.4.3. Získavanie a pracovanie dát z kamery zariadenia

Prototyp vie aktivovať kameru a pristupuje k dátam z nej, konkrétne zobrazuje ukážku na displej v reálnom čase. Problém nastáva pri odovzdaní týchto binarizovaných dát parseru knižnice ZXing, kedy aplikácia neočakávane skončí. Tento problém je stále vo fáze riešenia a debuggovania, keďže zdroje informácií o podobných problémoch sú zatiaľ skromné.

5.5. 3D prostredie

V súčasnosti sa najviac simuluje trojrozmerné prostredie. Základom trojrozmerného prostredia sú začiatok a tri osi (alebo súradnicové vektory – x , y , z), ktoré tvoria sústavu súradníc. Pre čo najlepšie a najjednoduchšie zobrazenia reálneho sveta do sveta imaginárneho sa používa karteziánska sústava súradníc. V karteziánskej sústave súradníc sú všetky tri základné osi na seba kolmé. Každý bod je reprezentovaný sústavou troch súradníc. Súradnice každého bodu sú súradnice jeho polohového vektora. Veľkosť polohového vektora je vzdialenosť bodu od začiatku súradnicovej sústavy. Začiatok súradnicovej sústavy je prienik troch osí x , y , z . Pomocou trojrozmerného prostredia môžeme vyjadriť polohu objektu, jeho šírku, výšku a hĺbku.

5.5.1. 3D model a modelovanie

V počítačovej grafike je 3D model výsledok modelovacieho programu. Je reprezentovaný množinou bodov v priestore. Tieto sú pospájané rôznymi geometrickými objektmi ako priamky, plochy atď. Model opisuje základné vlastnosti objektu: výška, šírka, hĺbka.

Modelovanie je vytváranie 3D modelu, ktorý reprezentuje objekt v trojrozmernom prostredí. V počítačovej grafike sa tým označuje proces vytvárania matematickej, drôtovej reprezentácie hocakého trojrozmerného objektu vytvoreného špeciálnym softwarom (modelovací software).

5.5.2. 3D grafické aplikácie

3D grafický software zahŕňa aplikácie, používané na vytváranie počítačovo generovaných 3D modelov. Existuje veľké množstvo 3D aplikácií, a ešte väčšie množstvo pluginov pre ne, keďže z veľkej miery je architektúra týchto programov orientovaná na pluginy.

TrueSpace

Bol vytvorený spoločnosťou Caligari. Momentálne je to voľne dostupný program na tvorbu 3D modelov, animácií a videí. Program je vhodný najmä pre začiatočníkov hlavne pre jednoduchý interface. Má nástroje aj na polygonálne modelovanie, NURBS, metaballs, implicitné plochy. Firmu Caligari odkúpila firma Microsoft, najmä kvôli projektu Microsoft Virtual Earth. Firma Microsoft sprístupnila trueSpace zadarmo, aby podnietila používateľa tvoriť v ich programe a pridávať vytvorené objekty na Virtual Earth. Práve kvôli tomu má posledná verzia niekoľko noviniek. TrueSpace je postavený na plug-in architektúre, čo poskytuje používateľovi vytvárať nástroje na rozšírenie základného jadra aplikácie. Obsahuje 2 interné renderovacie engine: lightworks, virtallights. V najnovšej verzii sa pridal ešte real-time DX9 renderer, ktorý sa približuje k fotorealistickej kvalite.

5.5.3. Možnosti 3D grafických aplikácií

Polygonálne modelovanie

Patrí k najčastejšie používaným typom modelovania. Každá plocha objektu reprezentuje polygón. Polygón alebo mnohouholník je geometrický útvar v rovine ohraničený úsečkami, ktoré spájajú rôzne body, z ktorých žiadne neležia na jednej priamke. Tento spôsob modelovania sa aplikuje na renderovanie po riadkoch. Z tohto dôvodu sa využíva v počítačovej grafike v reálnom čase. Základom modelovania je bod. Dva body spojené úsečkou tvoria hranu.

Tri body reprezentujú najjednoduchšiu plôšku, trojuholník. Každý modelovací software umožňuje editáciu polygónov, kde využíva tzv. box modeling. Tento má dva nástroje: subdivide (rozdeľuje plochy, hrany na menšie časti) a extrude (tento nástroj sa používa na plochy alebo skupinu plôch, vytvára novú plochu s rovnakou veľkosťou a rovnakým tvarom, ktorá je spojená každou existujúcou hranou práve jednou plochou).

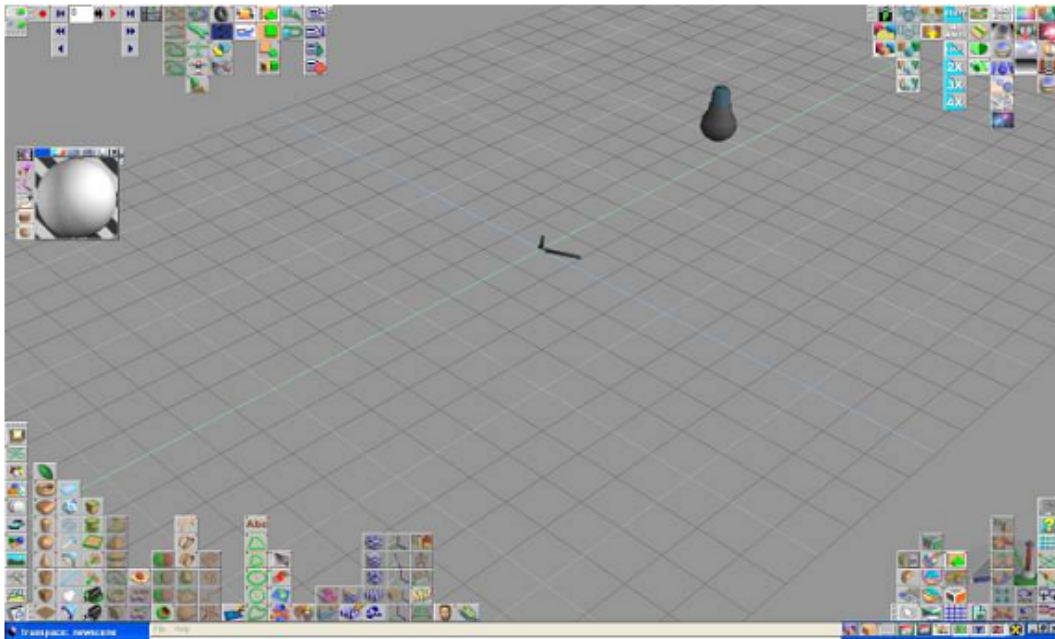
NURBS

Je skratka od Nonuniform rational B-spline. Technika matematického modelu sa využíva v počítačovej grafike na tvorbu a reprezentáciu kriviek. NURBS sa používa z viacerých dôvodov. K veľkým výhodám patrí nemennosť pri afinných transformáciách. Ich tvar sa definuje kontrolnými bodmi a uzlovým vektorom.

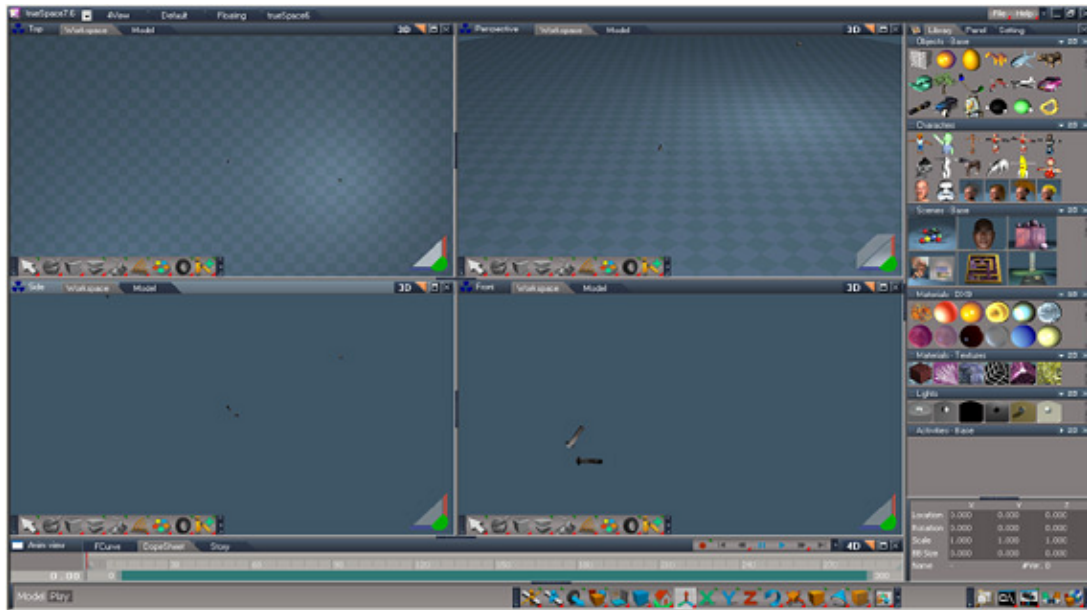
Prostredie trueSpace

TrueSpace ponúka veľmi jednoduché a pre používateľa prehľadné grafické rozhranie.

V úvodnom okne sa zobrazí perspektívny pohľad na virtuálnu scénu s klasickou mriežkou.



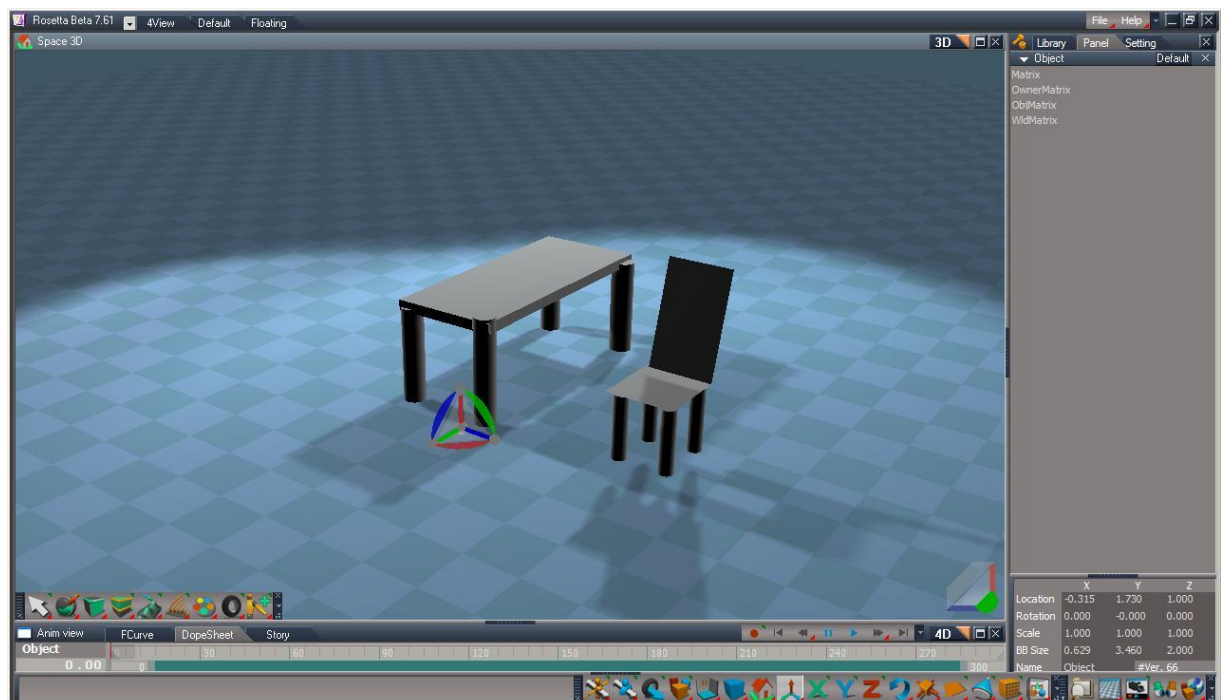
Obr. 44.: Prostredie TrueSpace.



Obr. 45.: Prostredie TrueSpace.

Základné panely nástrojov sú rozmiestnené po okrajoch okna. Ich rozloženie je možné kreatívne prispôbiť vlastným potrebám, alebo ak nie sú potrebné, tak úplne skryť. Všetky nástroje majú svoju vlastnú symbolickú ikonu. Pri zotrvaní myšou na danom tlačidle sa objaví krátky popis. Tento sa súčasne objaví aj v spodnej lište, hneď vedľa hlavného menu. Treba poznamenať, že hlavné menu sa nachádza netradične v spodnej časti obrazovky, taktiež ako aj základné nástroje na nastavenie scény, voľby objektov, voľby svetla, atď.

Pre zobrazenie v telefóne je potrebné vytvorené modely exportovať v .obj formáte.

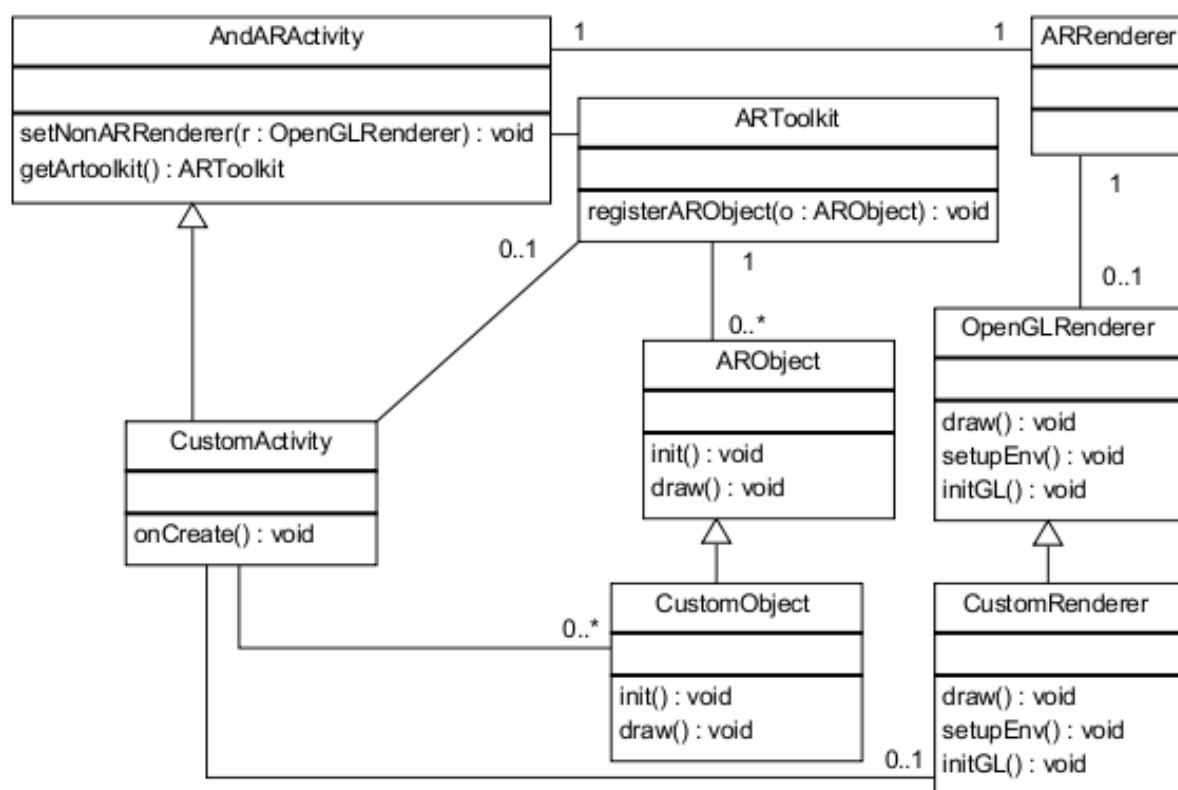


Obr. 46.: Vytvorený 3D model stolu a stoličky.

6. Implementácia – letný semester

6.1. Andar

Knižnica AndAR predstavuje hlavnú časť nášho projektu. Okrem jej hlavnej funkcie tvorby obohatenej reality, v našom projekte slúži i ako prepájací modul doplnkových komponentov. Jej architektúru možno opísať obrázkom uvedeným pod textom. Vybrané triedy, ktoré zohrávali dôležitú funkciu v projekte si následne stručne priblížime.



Obr. 47.: Architektúra knižnice AndAR

V nasledovných odstavcoch si priblížime jednotlivé triedy a ich význam.

CustomActivity – každá Android aplikácia pozostáva z jednej alebo z viacerých aktivít. Aktivita zobrazuje grafické rozhranie a rieši špecifickú úlohu. Súčasne môže byť aktívna iba jedna aktivita. *CustomActivity* teda predstavuje základnú aktivitu a zároveň i jediná v projekte AndAR. Je určená pre zobrazenie výstupu z kamery, obohateného o daný 3D model.

AndARActivity – predstavuje abstraktnú triedu, ktorá tvorí jadro obohatenej reality v projekte. Trieda realizuje snímanie obrazu z kamery, detekciu značiek a zobrazovanie videa. Zaujímavosťou tejto triedy je, že ponúka možnosť vytvorenia tzv. „screenshotu“ a uloženia ho na pamäťové médium.

ARObjects – je rovnako ako AndARActivity abstraktná trieda. Používa sa na registrovanie inštancií ARToolKitu. Inštancie predstavujú značky, ktoré sa nachádzajú v systémovom priečinku *assets*.

ARRenderer – je trieda zodpovedná za spoluprácu s OpenGL. Ak chceme v projekte využívať 3D modely vsadené do reality, ale i klasické 3D modely, tak ARRrenderer spolu s triedou OpenGLRenderer ponúka riešenie.

Po predstavení základných tried si uvedieme vybrané rozšírenia, ktoré sme uskutočnili s cieľom vylepšenia knižnice ako i prispôsobenia projektu naším cieľom.

Medzi dôležité rozšírenia rozhodne patrí pridanie metód, ktoré zachytávajú a ošetrojú jednotlivé stavy aktivít definovaných v OS Android. Pridané metódy sú nasledovné:

- onStart()
- onCreate()
- onPause()
- onResume()
- onDestroy()

Uvedené metódy zohrávali esenciálnu úlohu najmä pri práci s kamerou, kedy bolo potrebné ihneď po prejdení na novú aktivitu odovzdať kameru. OS Android po prepnutí na novú aktivitu, ponechá starú v pamäti a nastaví jej stav *Pause*. Ovládač kamery však neuvolní a preto nasledujúca aktivita nemôže používať kameru. Daný problém sme vyriešili implementáciou metódy onCreate(). Táto situácia nastávala, keď po načítaní QR kódu, bolo potrebné sa opätovne prepnúť naspäť na hlavnú aktivitu umožňujúcu prácu pomocou kamery s obohatenou realitou.

Okrem metód sme knižnicu obohatili i o niekoľko tried. Vybrané z nich si stručne priblížime.

SceneSetupActivity – predstavuje aktivitu, ktorá poskytuje grafické rozhranie pre výber tried. Používateľ si vďaka nej môže nastaviť umiestnenie objektov na jednotlivé značky a následne uloženie takéhoto nastavenia. Pri ďalšej interakcii tak nemusí opätovne priradovať značky k modelom, ale z priehľadného menu vyberie danú scénu. Aktivita úzko spolupracuje s triedou ScenePickerActivity.

ModelPickerActivity – umožňuje výber modelov, ktoré budú zobrazené nad jednotlivými značkami. Aktivita úzko spolupracuje s vyššie uvedenou aktivitou.

ScanActivity – aktivita, ktorá spolupracuje s komponentom ZXing. Jej úloha je pomocou tzv. *intent-u*, zavolať, požiadať o službu aplikáciu BarcodeScanner, ktorá predstavuje knižnicu ZXing. Po zavolaní, a vykonaní načítania QR kódu, ScanActivity obdrží URL adresu, spolu s názvom daného modelu. Úlohou aktivity je i poskytnutie používateľovi modálne okno, dotazujúce sa na potvrdenie stiahnutia modelu.

Vo vyššie uvedených odstavoch sme pre stručnosť uviedli iba zlomok zmien, ktoré boli v knižnici AndAR realizované. V nasledovnej kapitole si predstavíme najdôležitejší komponent ktorý bol bo našej aplikácie pridaný, a síce knižnicu ZXing.

6.2. Implementácia knižnice ZXing

Implementácia knižnice predstavovala postrach nášho projektu. Úlohy sa zúčastnili striedavo traja členovia tímu a až po dlhých týždňoch sa nám konečne podarilo prečítať daný QR kód a poslať získanú URL adresu aktivite, v rámci našej aplikácie. Hlavný problém, na ktorý sme dlhú dobu nevedeli prísť spočíval v úprave súboru *AndroidManifest.xml*, ktorý obsahuje definície všetkých zúčastnených aktivít, rovnako ako i nastavenia aplikácie a povolenia. Pre ilustráciu uvádzame kód, ktorý bolo potrebné do manifestu pridať, a ktorý nám spôsobil nemalé starosti:

```
<activity android:name=".ScanActivity"
          android:label="@string/app_name"
          android:screenOrientation="landscape"
          android:configChanges="keyboardHidden|orientation">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
  </intent-filter>
</activity>
```

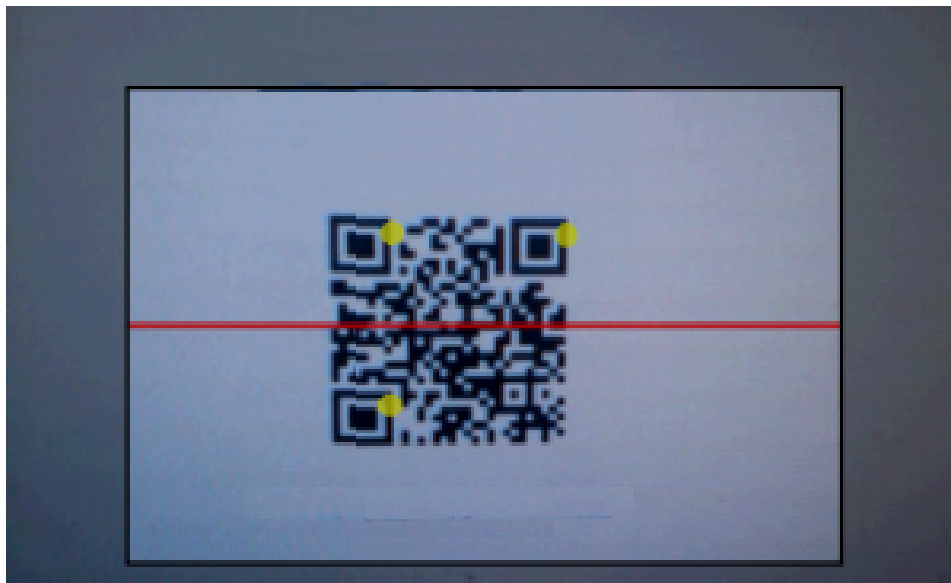
Jedná sa pritom najmä o definíciu tzv. *intent-filtra*. Absencia tejto časti kódu spôsobovala časté pády aplikácie rovnako ako i vypisovanie nezmyselných chýb odkazujúce sa na zlyhanie blesku mobilného telefónu. Po doplnení uvedeného kódu, a niekoľkých ďalších úpravách a rozšíreniach sa nám konečne podarilo úspešné načítanie QR kódu. Hlavnú úlohu pritom zohrávala aplikácia BarcodeScanner, ktorú vyššie opísaná ScanActivity volala. Pri absencii danej aplikácie, bol používateľovi poskytnutý dialóg umožňujúci mu jej stiahnutie. Aplikáciu BarcodeScanner, považujeme sa vhodné si pomocou niekoľkých odstavcov priblížiť.

6.2.1. BarcodeScanner

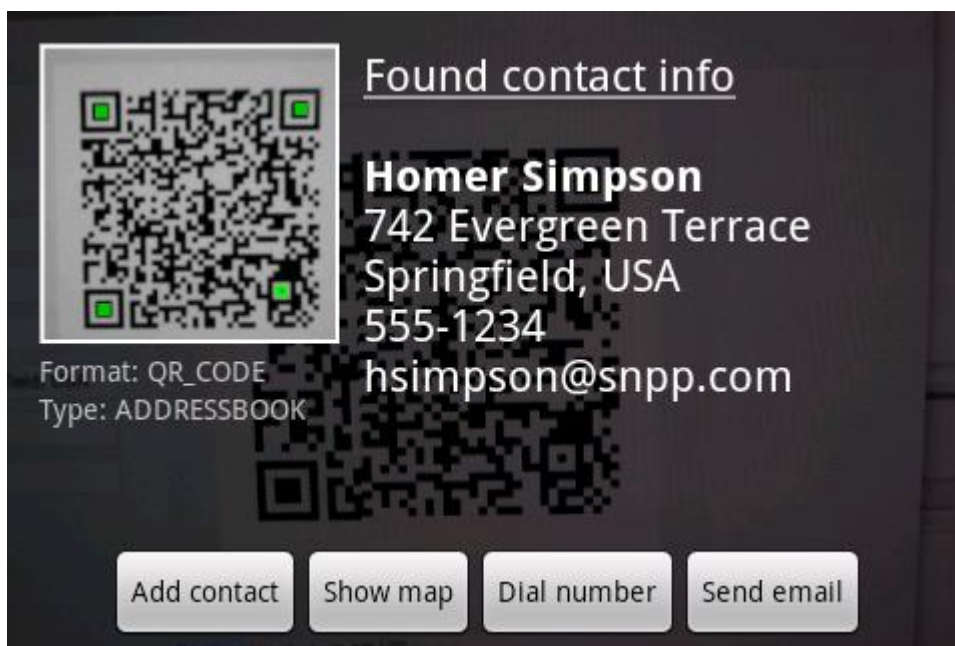
Pre čítanie QR kódov sme sa rozhodli neimplementovať nové metódy, ale použijeme už existujúce riešenia. Po vyhodnotení niektorých aplikácií ako napr. QR Droid alebo QuickMark QR Code, sme sa rozhodli použiť voľne dostupnú aplikáciu Barcode Scanner.

Barcode Scanner je súčasťou knižnice ZXing (Zebra Crossing), ktorá je určená pre čítanie 1D/2D čiarových kódov. Knižnica ZXing je implementovaná vo viacerých jazykoch, medzi ktoré patrí i Android Java. Táto skutočnosť vytvára ideálne podmienky pre prepojenie ZXing-u s knižnicou AndAR. Hlavné zameranie knižnice je orientované na mobilné telefóny obsahujúce kameru, pričom pri dekodovaní daného kódu nie je potrebné spojenie so serverom.

Pre ilustráciu uvádzame i obrázky zobrazujúce čítanie QR kódu, rovnako ako i neskoršie spracovanie informácie.



Obr. 48.: Načítanie QR kódu.



Obr. 49.: Zobrazenie dát QR kódu.

Okrem komponentu sú v projekte zahrnuté i ďalšie komponenty ako sťahovanie a dekompresia stiahnutých súborov. Dané komponenty však nebudeme bližšie v rámci dokumentu približovať.

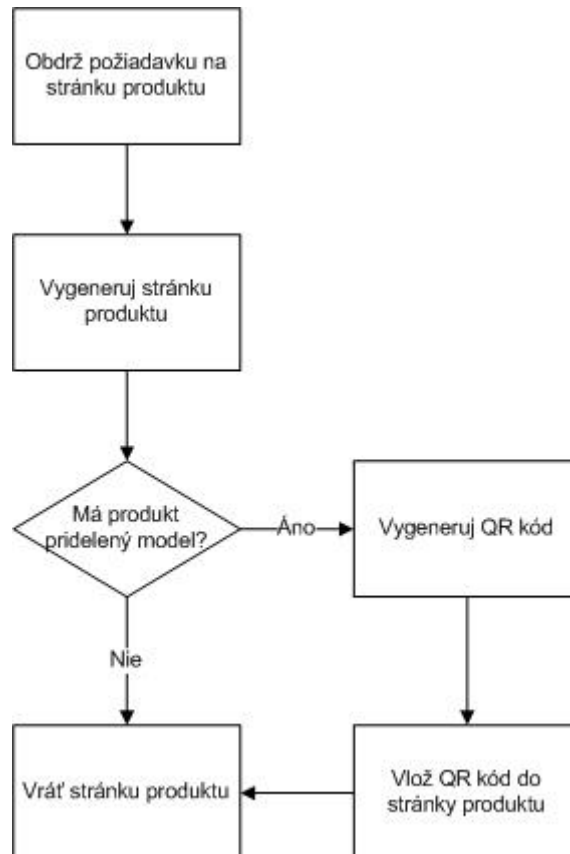
6.3. Server

V tejto kapitole je priblížený spôsob realizácie serverovej časti a rovnako sú opísané jednotlivé služby, ktoré server poskytuje.

6.3.1. Služby

Integrácia do internetového obchodu

Jednou z vlastností, ktorú sme plánovali do nášho riešenia zakomponovať, bolo prepojenie mobilnej aplikácie s predajným systémom. Pre otestovanie tejto možnosti sme sa rozhodli využiť internetový obchod. Ten obsahuje jednotlivé produkty, ku ktorým máme možnosť pridelovať 3D modely. Ak má daný produkt pridelený 3D model, systém vygeneruje QR kód a uverejní ho na stránke daného produktu.



Obr. 50.: Diagram fungovania servera.

Z hľadiska užívateľa je v prípade internetového obchodu daný QR kód jedinou zmenou. Z hľadiska prevádzkovateľa je nutné zabezpečiť modely a službu, ktorá bude na základe URL obsahnutom v QR kóde dané modely poskytovať. Integrácia QR kódu do systému je jednoduchá, prevádzkovateľ je zbavený potreby ručného generovania albo pridelovania QR kódov.

Stiahnutie modelu

Model je stiahnutý na základe URL obsiahnutom v QR kóde. Vo všeobecnosti môže ísť buď o URL na službu, ktorá tento požiadavok ďalej spracuje alebo už na predpripravený súbor s modelom. Podmienkou je dodržanie formátu.

V našom prípade je daným súborom archív ZIP obsahujúci nasledujúce súbory:

- *Info.xml* – obsahuje údaje o produkte, jeho štruktúra je nasledujúca:


```
<?xml version="1.0"?>
<product id="31">
  <name>Pohodlo</name>
  <price>40.0000</price>
  <desc>Pohodle - the sofa for your living room</desc>
</product>
```
- Ako vindo, obsahuje názov produktu, cenu a opis produktu.
- *Prod.png* – je obrázkom daného produktu, užívateľovi sa obrazí pri áhlade na produkt pre názornejšiu ukážku.
- *Prod.obj* - predstavuje samotný model produktu, ide o formát obsahujúci popis jeho povrchu.
- *Prod.mtl* – je doplnkovým povinným súborom k súboru formátu *Obj*. Obsahuje popis materiálov.

Naše riešenie generuje ZIP archív pre daný produkt na základe požiadavky, obsluhujúcim súborom je *getModel.php*, ktorý ako parameter prevezme ID produktu. Následne z databázy načíta príslušné modelové súbory, obrázkov a informácie a vloží ich do ZIP archívu. Táto činnosť sa deje bez vytvorenia archívu v rámci súborového systému na strane servera, klientovi je archív predný pomocou HTTP hlavičky:

```
header("Content-type: application/octet-stream");
header("Content-disposition: attachment; filename=test.zip");
```

Vygenerovanie QR kódu

QR kód je generovaný len pre produkty, ktoré majú pridelený model. Samotné generovanie je vykonané pomocou knižnice, celý zdrojový kód je nasledujúci:

```
<?php
header("Content-type: image/png");
require('../extras/phpqrcode/phpqrcode.php');
QRcode::png($_GET['data']);
?>
```

Parametrom pre tento skript je URL na skript *getModel.php* spolu s parametrom ID produktu, ktoré má byť do QR kódu uložené.

Riešenie bolo použité na testovanie funkčnosti aplikácie, v prípade reálneho nasadenia by bolo potrebné ošetriť bezpečnostné riziká súvisiace s PHP konštrukciou `$_GET`.

6.3.2. Použité prostriedky

PHP QR Code

Na generovanie QR kódu bola použitá knižnica PHP QR Code (<http://phpqrcode.sourceforge.net/>). Ako vidno v o výpise zdrojového kódu v kapitole 0, použitie knižnice je jednoduché. Inštalácia knižnice spočíva len v jej nakopírovaní do adresára, z ktorého ju následno môžeme importovať.

Na fungovanie tejto knižnice je potrebné mať v PHP nainštalovanú a povolenú knižnicu GD2.

Os Commerce

Pri našom riešení bolo potrebné otestovať funkcionality systému s obchodom. Na toto otestovanie sme si vybrali internetový obchod založený na OSCommerce (<http://oscommerce.com/>). Bola použitá verzia 2.3.1. Nasadenie internetového obchodu je bezproblémové a vyžaduje nasledujúce kroky:

7. Inštalácia webového servera (Apache)
8. Inštalácia databázového servera (MySQL)
9. Inštalácia skriptovacieho jazyka PHP
10. Nasadenie samotného OSCommerce

Prvým trom krokom sa venujeme bližšie v nasledujúcich častiach dokumentácie. Nasadenie OSCommerce spočíva v skopírovaní jeho súborov do adresára, ktorý ma webový server nastavený ako zdrojový. Po otvorení adresára OSCommerce v internetovom prehliadači nasleduje nastavenie internetového obchodu pomocou webového rozhrania.

Počas inštalácie je nutné zadať (okrem iného) nasledujúce údaje:

- SQL server
- Prihlasovacie údaje na SQL server
- Meno a heslo správcu internetového obchodu

Na základe týchto údajov inštalčný skript vytvorí kompletnú štruktúru SQL databázy.

Ďalším krokom inštalácie OSCommerce bola úprava obchodu pre naše potreby. Po inštalácii bol vytvorený ukázkový obchod s elektronikou. Bolo potrebné tieto produkty odstrániť a vytvoriť obchod s nábytkom.

MySQL

Použitá bola databáza MySQL. MySQL bolo použité na základe požiadavky použitého internetového obchodu. Inštalácia MySQL servera prebehla počas inštalácie samotného operačného systému Ubuntu Server. Po nainštalovaní databázy a internetového obchodu bola k databázovej štruktúre obchodu pridaná ďalšia tabuľka. Išlo o tabuľku prepájajúcu produkty a modely. Jej štruktúra je nasledujúca:

Models	
PK	model_id
	product_id model_file texture_file image added comment

Obr. 51.: Tabulka prepojenia modelov a produktov.

Položka *product_id* slúži na prepojenie s databázou produktov. Položky *model_file* a *texture_file* obsahujú názvy pridelených súborov.

PHP

PHP bolo použité na základe požiadavky použitého internetového obchodu. Inštalácia prebehla počas inštalácie samotného operačného systému Ubuntu Server. Následne bolo potrebné kvôli knižnici PHP QR Code pridať knižnicu na generovanie obrázkov GD2 pomocou *apt-get install php5-gd*

Apache

Apache bol vybraný ako webový server z dvoch hlavných dôvodov. Prvým dôvodom bola podpora Apache priamo v rámci operačného systému, kde inštalácia Apache, rovnako ako PHP a MySQL prebehla automaticky. Druhým dôvodom boli naše predchádzajúce skúsenosti s týmto webovým serverom. Do webového servera nebolo po inštalácii nijako zasahované.

Linux

Ako operačný systém pre náš server sme zvolili po predchádzajúcich skúsenostiach opravný systém Linux, konkrétne edíciu Ubuntu Server 10.10 64bit. Výhodou je podpora tohto operačného systému smerom k nami používaným prostriedkom ako je PHP,MySQL a Apache.

6.4. Databáza klientskej časti

Na vytvorenie databázy mobilnej aplikácie tohto projektu je používaná technológia SQLite. Jedná sa o systém na správu relačných databáz, ktorý funguje aj na mobilných zariadeniach so systémom Android.

Naša aplikácia pri každom spustení overí, či už je vytvorená tabuľka databázy určená na ukladanie údajov o jednotlivých modeloch. Ak nie, vytvorí novú, prázdnu tabuľku, ak sa nejedná o prvé spustenie aplikácie, používa sa nájdená, uložená tabuľka.

6.4.1. Tabuľka modelov

Tabuľka s dátami o modeloch obsahuje napríklad tieto údaje:

- ID modelu
- Opis modelu
- Cesta k príslušnému OBJ súboru
- Cesta k príslušnému MTL súboru
- Cesta k obrázku

Do tabuľky sa pridávajú údaje hneď ako sa stiahne nový model do pamäti telefónu. Údaje o modeli ostanú uchované aj po skončení aplikácie.

6.4.2. Funkcie nad tabuľkou modelov

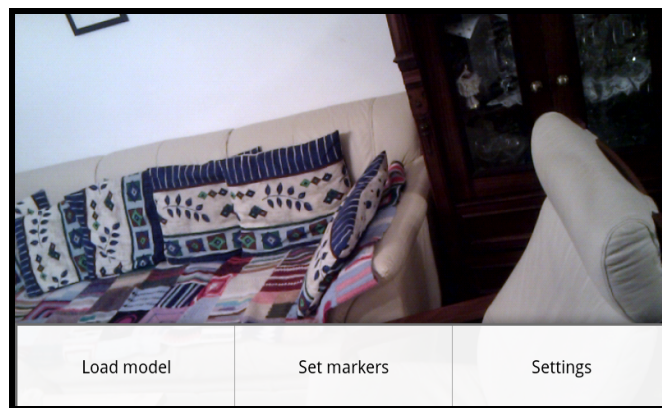
Najdôležitejšie operácie a teda aj funkcie pracujúce s tabuľkou modelov v databáze na mobilnom zariadení sú tieto:

- Get model by ID
(vráti model podľa jeho ID)
- Get model basic info
(ID, MTL path, OBJ path)
- Delete by ID
(vymaže model s daným ID)
- Update ID
(zmení model s daným ID)
- Add new model
(pridá nový záznam do tabuľky)

6.5. Grafické používateľské rozhranie

Vyvíjaná aplikácia Furniture Viewer je implementovaná na platformu Android. Väčšina zariadení, ktoré fungujú s operačným systémom Android disponujú dotykovými displejmi. Práve tento fakt existencie dotykových displejov určuje, že tomu musí byť prispôsobený aj vzhľad vyvíjaných aplikácií. Nami implementovaná aplikácia poskytuje funkcionality obohatenej reality, čo je opísané v predošlých kapitolách. No z hľadiska grafického používateľského rozhrania, v prípade obohatenej reality, sme k tomu prispôbili vzhľad aplikácie. Pre maximalizáciu možného viditeľného obrazu počas behu aplikácie je obraz snímaný kamerou zobrazovaný na celú plochu obrazovky zariadenia.

Aplikácia po spustení zobrazí obraz z kamery a zároveň sa nachádza v režime, v ktorom detekuje značky nad ktorými zobrazí načítané modely. Menu v tomto režime je znázornené na obrázku 1. Poskytuje možnosti načítania modelu z QR kódu, priradenie modelu ku značke a tiež voľbu, nastavení.



Obr. 52.: Menu.

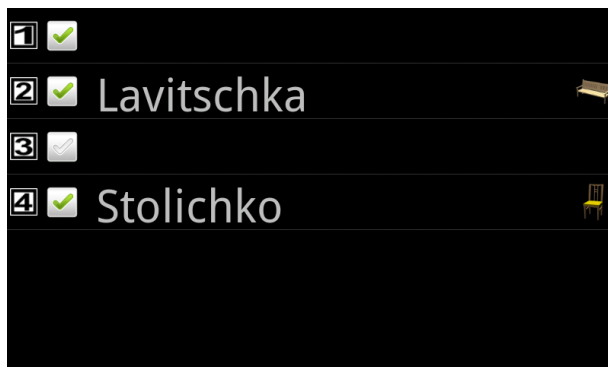
Load model

Načítavanie modelov (Load model) spustí pomocou Android Intent aplikáciu BarCode scanner. Táto aplikácia načíta z QR kódu linku, pomocou ktorej sa stiahne zo servera vybraný model nábytku do mobilu.

Set markers

Táto časť aplikácie slúži na priradenie načítaných modelov k jednotlivým značkám. Otvorí sa zoznam dostupných značiek. Pre tento projekt sme definovali 4 značky, no množstvo značiek je teoreticky neobmedzené.

Ako je vidno na obrázku 2, značke 2 sme priradili načítaný model lavičky, podobne značke 4 bol priradený model stoličky. Následne po návrate späť a načítaní scény sú vybrané modely zobrazené na displeji nad vybranými značkami.

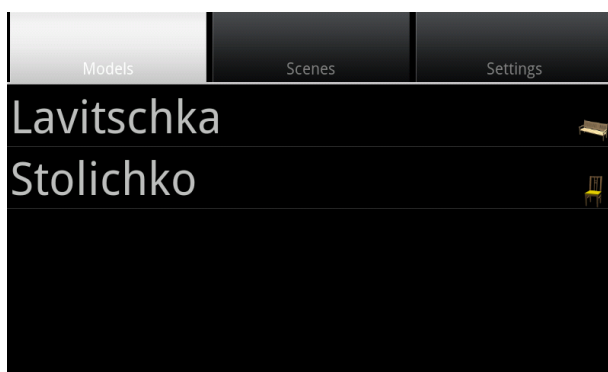


Obr. 53.: Prirad'ovanie modelov k značkám

Settings

Do podmenu nastavení (Settings) sme umiestnili formou záložiek možnosti výberu modelov, o ktorých je možné po kliknutí pozrieť detailné informácie, a taktiež možnosť vymazávania načítaných modelov.

Zvyšné 2 záložky scenes a settings súvisia so spravovaním scén a nastaveniami samotnej aplikácie.



Obr. 54.: Nastavenia aplikácie.

7. Zhodnotenie

Predmet Tímový projekt nás sprevádzal uplynulým rokom štúdia a dovoľme si tvrdiť, že bol časovo spomedzi ostatných predmetov najnáročnejší ale zároveň i najdynamickejší. Bol to čas, pri ktorom sme sa postupne oboznamovali najprv so schopnosťou pracovať v tíme, neskôr s analýzou témy až nakoniec samotnou implementáciou riešenia. V rámci prvých týždňov sme sa zoznamovali s našimi pridelenými rolami v tíme a učili sa navzájom počúvať a akceptovať. Všetci členovia tímu sa však zhodneme, že aj napriek nezhodám a rozdielnym názorom, to bolo veľmi prínosné obdobie, ktoré nám poskytlo viac ako základ nielen pre našu budúcu prácu ale i prínos pre orientovanie sa v živote.

Prvý semester prebiehal v znamení analyzovania obohatenej reality. Táto spočiatku pre nás neobjavená technológia sa stala často vyhľadávaným pojmom v našich internetových prehliadačoch rovnako ako i často skloňovanou témou v rámci tímových obedoch. Postupne sme si vytvárali obraz budúceho riešenia pomocou diskusií s vedúcim projektu. Analyzovali sme množstvo technológií, ktoré nás neraz inšpirovali a rozhodli sme sa ich zapojiť do projektu. Koncom úvodného semestra sme vytvorili prototyp aplikácie. Daný prototyp už bol schopný tvorby obohatenej reality. Tento fakt nás motivoval pre prácu na projekte i v druhom semestri.

Druhá polovica predmetu predstavovala implementáciu analýzy, ktorú sme vykonávali počas prvého semestra. Jednalo sa o obdobie, kedy sme neraz vďaka problémom pri programovaní zvažovali správnosť výberu našej témy. Vytrvali sme však až do konca a počas neľahkých troch mesiacoch, sme prototyp aplikácie zmenili na takmer profesionálne riešenie, ktoré zaujalo nejedného návštevníka konferencie IITSRC.

Predmet nám dal okrem spoznania spomínanej tímovej práce i nové pohľady v oblasti moderných technológií. Počas riešenia sme sa neustále zoznamovali a pracovali s niečím novým, pre nás ešte neprebádaným.

Pri analýze toho, čo nám tento predmet zobrať, prichádzame všetci s rovnakou odpoveďou, a síce voľný čas. Bolo to obdobie kedy sme zanedbávali naše záujmové krúžky, rovnako ako i spoločenský život ako taký. No aj napriek tomu môžeme všetci prikývnuť, že bol to správne investovaný čas.

Použité zdroje a literatúra

1. Tatra Banka. *Balik služieb*. Citované 22.10.2010. Dostupné z <www.baliksluzieb.sk>
2. Augmented reality encyclopedia [video]. Citované 23.10.2010. Dostupné z <<http://www.youtube.com/watch?v=oHkUOpYNhoM&feature=related>>
3. Sueddeutsche Zeitung is the first augmented reality enhanced magazine! [online]. Publikované 23.8.2010. Citované 23.10.2010. Dostupné z <<http://www.intomobile.com/2010/08/23/sueddeutsche-zeitung-is-the-first-augmented-reality-enhanced-magazine/>>
4. SnapShop - Finding, Visualizing and Sharing Furniture from your iPhone [video]. Citované 25.10.2010. Dostupné z <<http://www.youtube.com/watch?v=62fLkCZLnm0>>
5. Christian Dopler Laboratory. *Studierstube tracker* [online]. Publikované 29.5.2010. Citované 30.10.2010. Dostupné z <http://studierstube.icg.tu-graz.ac.at/handheld_ar/>
6. Augmented Environments Laboratory. *KHARMA – KML/HTML Augmented Reality Mobile Architecture* [online]. Georgia Institute of Technology School of Interactive Computing, USA. 2010. Citované 30.10.2010. Dostupné z <<https://research.cc.gatech.edu/polaris/content/home>>
7. Layar. *An overview of the Layar platform* [online]. 2010. Citované 30.10.2010. Dostupné z <<http://site.layar.com/create/platform-overview/>>
8. Wikitude. *Wikitude World Browser*. Publikované 30.9.2010. Citované 30.10.2010. Dostupné z <http://www.wikitude.org/category/02_wikitude/world-browser>
9. Wikitude. *Wikitude Drive*. Publikované 9.8.2010. Citované 30.10.2010. Dostupné z <http://www.wikitude.org/category/02_wikitude/wikitude-drive>
10. Tagwhat. *About Tagwhat* [online]. 2010. Citované 30.10.2010. Dostupné z <<http://www.tagwhat.com/>>
11. Craic Design. *Pocket Universe*. 2010. Citované 30.10.2010. Dostupné z <<http://pocketuniverse.info/>>
12. Hunter Research & Technology. *Theodolite*. 2010. Citované 30.10.2010. Dostupné z <<http://hunter.pairsite.com/theodolite/>>
13. Schmalstieg, D., Wagner, D. Experiences with Handheld Augmented Reality. In: *ISMAR '07: Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. 2007. ISBN 978-1-4244-1749-0
14. ARToolKit. Citované 28.10.2010. Dostupné z <<http://www.hitl.washington.edu/artoolkit/>>
15. ARToolKitPlus. Citované 28.10.2010. Dostupné z <http://studierstube.icg.tu-graz.ac.at/handheld_ar/artoolkitplus.php>
16. NyARToolKit Citované 28.10.2010. Dostupné z <<http://www.virtualworldlets.net/ResourcesHosted/Resource.php?Name=NyARToolKit>>
17. FLARToolKit, Citované 28.10.2010. Dostupné z <<http://www.virtualworldlets.net/Resources/Hosted/Resource.php?Name=FLARToolKit>>
18. Denso-Wave Incorporated. *About QR Code* [online]. Citované 30.10.2010. Dostupné z <<http://www.denso-wave.com/qrcode>>
19. GS1. *DataMatrix*. Verzia 1.17. 2010. Dostupné z <http://www.gs1.org/docs/barcodes/GS1_DataMatrix_Introduction_and_technical_overview.pdf>
20. Morovia Corporation. *PDF417 Specification* [online]. Publikované 3.5.2004. Citované 30.10.2010. Dostupné z <<http://www.morovia.com/education/symbology/pdf417.asp>>

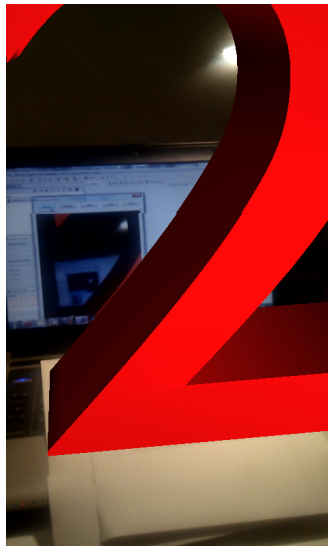
21. International Air Transport Association. *Bar Coded Boarding Pases (BCBP)* [online]. September 2010. Dostupné z <<http://www.iata.org/whatwedo/stb/bcbp/Pages/materials.aspx>>
22. Content Idea of China Co., Ltd.. *What is PM code?*. 2008. Dostupné z <http://ci-a.co.jp/pm/pm_eng_01.pdf>
23. ZXing. *Barcode contents* [online]. Publikované 28.6.2010. Citované 30.10.2010. Dostupné z <<http://code.google.com/p/zxing/wiki/BarcodeContents>>
24. IT arts. *QRPR augmented reality* [online]. Citované 30.10.2010. Dostupné z <http://it-arts.com/projects/augmented_reality.aspx>
25. Kan, T., Teng, Ch., Chou, W. Applying QR code in augmented reality applications. In: *VRCAL '09: Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry*. 2009. ISBN 978-1-60558-912-1
26. Freeman, R., Steed, A., Zhou B.: Rapid scene modelling, registration and specification for mixed reality systems. In: *Proceedings of the ACM symposium on Virtual reality software and technology*. 2005. s. 147-150
27. Apache server. Citované 29.10.2010. Dostupné z <http://httpd.apache.org/ABOUT_APACHE.html>.
28. ZXing. Citované 29.10.2010. Dostupné z <<http://code.google.com/p/zxing>>
29. Android SDK. Citované 29.10.2010. Dostupné z <<http://developer.android.com/sdk/index.htm>>

PRÍLOHA A: Používateľská príručka

Ovládanie aplikácie Furniture viewer je prispôsobené dotykovým displejom a je navrhnuté a implementované tak, aby bolo použiteľné aj pre technicky menej zdatných používateľov. Celá aplikácia sa ovláda niekoľkými kliknutiami na displeji zariadenia.

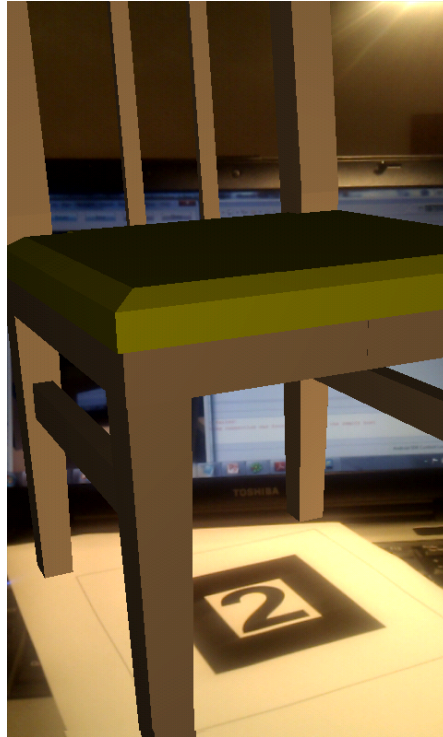
Spustenie a základný princíp aplikácie Furniture Viewer

Po spustení aplikácie FV je na celú plochu displeja zobrazený v reálnom čase snímaný obraz z kamery zariadenia. V tomto obraze sú detekované určené značky nad ktorými sa vizualizujú predurčené modely. Predurčený model bol zvolený 3D model čísla danej značky. Tieto modely sú zobrazované v prípade, ak snímanej značke nie je priradený iný, stiahnutý model. Na obrázku 4 je zobrazený predurčený model čísla 2.



Obr 1.: Zobrazenie preddefinovaného modelu nad značkou.

Po načítaní modelu (viď Načítanie modelu) je možné tento model priradiť niektorej zo štyroch dostupných značiek (viď Priradenie modelu k značkám). Ak priradíme napríklad načítaný model stoličky značke 2, tak po detekovaní značky číslo 2 kamerou zariadenia bude nad touto značkou zobrazený model stoličky. Táto skutočnosť je znázornená na obrázku 5.



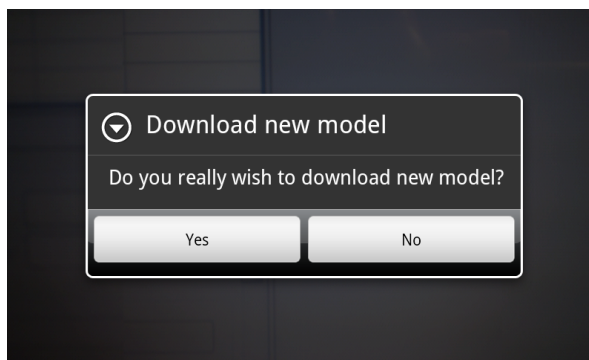
Obr 2.: Zobrazenie modelu nad značkou.

Týmto spôsobom je možné zobraziť model aj nad viacerými značkami, po priradení modelu značkám.

Pre zobrazenie modelu v reálnej veľkosti je potrebné nastaviť reálnu veľkosť značky v cm. Toto nastavenie veľkosti značky sa realizuje vo voľbe Menu/Set Markers. Po dlhšom kliknutí na dotykovú obrazovku v časti vypísaných značiek (obrázok 2) značiek sa zobrazí okno, do ktorého je možné nastaviť reálnu veľkosť značky. Implicitne je veľkosť značky nastavená na hodnotu 18cm.

Načítanie modelu

Načítanie modelu funguje na základe zvolenia módu v menu Load model (Menu/Load Model). Toto je vidno na obrázku 1. Po zvolení danej možnosti sa spustí aplikácia načítavajúca QR kódy. Namierte kamerou zariadenia na QR kód nachádzajúci sa v katalógu nábytku pri vybranom kuse nábytku. Po načítaní QR kódu Vám zariadenie zobrazí voľbu, či chcete stiahnuť vybraný model do telefónu alebo nie. Toto je zobrazené na obrázku 6. Po odsúhlasení bude model stiahnutý zo servera do mobilu.



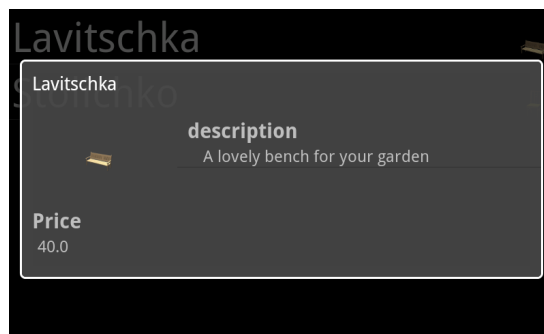
Obr 3.: Sťahovanie modelu.

Priradenie modelu k značkám

Načítané modely sa priradujú k predurčeným značkám vo voľbe Menu/Set Markers (obrázok 1). Po zvolení tejto voľby sa na obrazovke zariadenia zobrazí zoznam dostupných značiek (obrázok 2). Po kliknutí na niektorú zo značiek sa otvorí voľba s možnosťou výberu zo stiahnutých modelov. Následne zvolíte model, ktorý si želáte danej značke priradiť.

Získanie informácií o modely

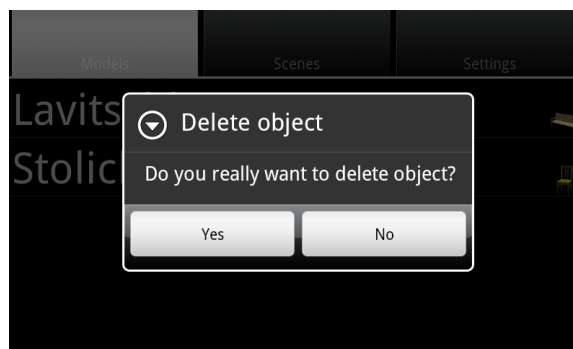
Spolu so stiahnutým samotným modelom sa stiahne do zariadenia aj opis tohto modelu, a jeho cena. Tieto údaje je možné zobraziť voľbou Menu/Settings (obrázok 1). Dostanete sa do podmenu nastavení (obrázok 3). Vo zvolenej záložke Models, po kliknutí dotykového displeja na požadovanom modely sa zobrazí informačné okno o danom modely (obrázok 7).



Obr 4.: Informácie o modely.

Odstránenie modelu

Odstránenie stiahnutého modelu je riešené voľbou Menu/Settings (obrázok 1). Dostanete sa do podmenu nastavení (obrázok 3). Vo zvolenej záložke Models, po dlhšom podržaní dotykového displeja na požadovanom modeli sa zobrazí potvrdzujúca otázka (obrázok 8), či si želáte daný model odstrániť. Po jej odsúhlasení bude daný model odstránený.



Obr 8.: Odstránenie modelu.