Universitatea “Ştefan cel Mare” Suceava

Facultatea Inginerie Electrică şi Ştiinţa Calculatoarelor

Proiect la Prelucrarea Numerică a Imaginilor

Tema: Realitatea augmentată pe dispozitive Android

Îndrumător: s.l. dr. ing. Prodan Remus

Realizat de: Plotean Mihail

Suceava,

2012

Cuprins

Introducere

Abstract

Termenul „Realitate Augmentată” a fost introdus în 1990 de către Thomas Caudell, pe atunci angajat al Boeing. Realitatea augmentată este considerată o extensie a realităţii virtuale – un spaţiu virtual în care timpul, fizica şi materialele din lumea reală pot fi depăşite. Mai jos este prezentată celebra ilustraţie a continuumului Real-Virtual.

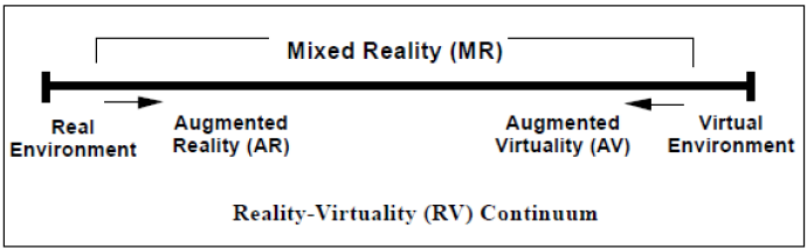


Fig.1 Continuumul Real-Virtual, Milgram, Takemura, Utsumi şi Kishino

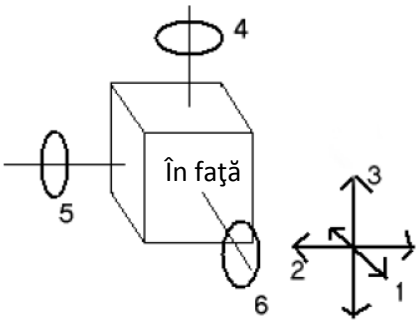
Astfel, realitatea augmentată este o tehnologie ce suprapune o lume virtuală generată de către computer peste lumea reală, înregistrată de către sensor-ul video al dispozitivului. Lumea virtuală poate fi formată din obiecte bi- sau tri-dimensionale de orice tip.

Întreaga tehnologie se bazează pe câţiva termeni, descrişi în continuare:

- Vederea reală – se referă la fluxul video produs de camera telefonului; este aceeaşi vedere obţinută atunci când este utilizată camera video a dispozitivului; aplicaţia captează imaginilie din fluxul video şi augmentează în timp real cu obiecte virtuale pentru a crea o realitate extinsă.
- Înregistrare şi observare – descrie metodele disponibile pentru a alinia obiectul virtual la vederea reală; aceasta implică senzorii de locaţie, cum este GPS-ul, compasul digital şi accelerometrul (observare bazată pe locaţie) sau un sistem de recunoaştere a imaginii (observare optică); unele aplicaţii folosesc ambele metode.
- Punctul de interes – este un item individual, de obicei, asociat cu o locaţie geografică(longitudine, latitudine, altitudine) sau un model vizual (marker, coperta unei cărţi, o imagine, etc.) ce poate fi redată de către aplicaţie; datele punctului de interes conţin descrierea locaţiei sau referinţa imaginii folosite pentru observare şi tipul de conţinut pentru a fi generat. De obicei, conţinutul însuşi nu este parte a punctului de interes, dar în schimb este furnizată o legătură către acesta.

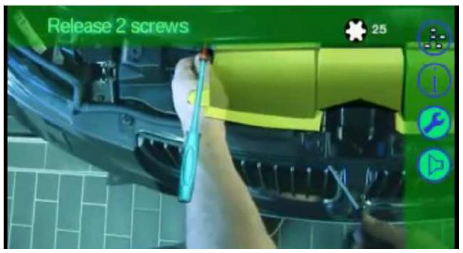
- Obiectul virtual – conţinutul digital generat de către aplicaţie şi suprapus peste vederea reală; acest conţinut poate include modele 3D, 2D, pictograme şi text.
- Canale, nivele şi lumi – se referă la grupurile ce publică obiectele virtuale ale punctului de interes asociat.
- Augmentare bazată pe marker – este cazul în care pentru observarea optică sunt folosite imagini artificiale intenţionat create pentru a servi aplicaţiei de realitate augmentată.

- Augmentare fără marker-e - este situaţia în care la observarea optică sunt utilizate imagini „naturale", nemodificate.
- Observare bazată pe locaţie – obiectul virtual este suprapus peste fluxul real, luând în calcul informaţiile geo-locaţionale obţinute de la senzorii terminalului.
- 6 grade de libertate – se referă la capabilitatea sistemului de observare de a menţine alinierea la un obiect real într-un spaţiu tridimensional; senzorii de locaţie sunt capabili să ofere informaţii referitoare la direcţiile „înainte/înapoi"(1), „stânga/dreapta"(2), „sus/jos"(GPS-ul)(3), „giraţie"(compasul)(4), „înclinare"(5) şi „rostogolire"(accelerometrul)(6).



- Near Field Communication – tehnologie bazată pe wireless de rază scurtă (până în 4 cm); implică un chip „iniţiator" activ şi unul „ţintă" pasiv care poate fi activat de către câmpul magnetic al iniţiatorului; aşadar, ţinta nu are nevoie de baterii şi poate avea forme flexibile şi portabile(carduri, stick-ere, tag-uri); NFC-ul se presupune a fi înlocuitorul marker-elor.

Datorită dispozitivelor mobile tot mai performante, au apărut diverse aplicaţii ce implementează realitatea augmentată. Dintre acestea, cele mai răspândite sunt aşa-numitele browser-e: Junaio, Layar, Sekai Camera, Wikitude Worlds Browser, LibreGeoSocial Open Source Browser ş.a. Există şi o multitudine de alte aplicaţii în diverse domenii cum ar fi cel educaţional, distractiv, publicitar, ş.a.m.d.



← Aplicaţia BMW de asistenţă a tehnicianului la reparaţia motorului.

Aplicaţie de modelare virtuală a inimii umane. →

Descriere

Prezenta lucrare descrie modul în care poate fi realizată o aplicaţie de realitate augmentată pentru un dispozitiv cu sistemul de operare Android.

Pentru realizarea aplicaţiei a fost utilizat instrumentul de dezvoltare a jocurilor, Unity3d cu add-on-ul pentru Android(disponibil gratis până pe 08 aprilie 2012) şi extensia pentru acesta - Vuforia a celor de la Qualcomm. Unity3d a fost ales pentru simplitatea cu care se poate crea o aplicaţie de realitate augmentată. Pentru extensia Vuforia, realizată de Qualcomm, s-a optat datorită perfomanţelor şi suportului pe care comunitatea Qualcomm îl oferă. Astfel, a fost posibilă crearea aplicaţiei fără a scrie nicio linie de cod.

Vuforia permite diverse implementări ale realităţii augmentate: modelul peste care se suprapune lumea virtuală este o imagine – în acest caz suportă o ţintă unică; modelul este un marker de tipul Qualcomm – sunt posibile ţinte multiple. De menţionat că marker-ul Qualcomm este un marker specific şi trebuie obţinut de la Qualcomm(gratis). Proiectul în cauză foloseşte o imagine peste care va suprapune un model 3D. Acest obiect este creat cu setul de instrumente Blender, disponibil, la fel, gratis.

Modelul, denumit de Qualcomm Image Targets, este imaginea pe care Vuforia o poate detecta şi urmări. Sunt folosiţi algoritmi sofisticaţi pentru a detecta şi urmări proprietăţile imaginii. Ţinta este recunoscută prin compararea acestor caracteristici naturale cu proprietăţile imaginii păstrate într-o bază de date. Odată recunoscută, ţinta va fi urmărită atât timp cât ea se află, cel puţin parţial, în câmpul de vizibilitate al camerei. Aceasta este una din performanţele Vuforia – obiectul 3D este redat chiar şi la unghiuri apropiate de 180^0 sau la distanţe, relativ, mari faţă de imaginea model. Aceste target-uri sunt create on-line pe site-ul Vuforia(necesită crearea unui cont) din fişiere .jpg sau .png(sunt suportate doar imagini RGB sau în gamă de gri), utilizând un sistem de management al ţintelor – Target Management System. Caracteristicile extrase sunt păstrate într-o bază de date şi folosite pentru comparaţiile din runtime. Toate ţintele sunt organizate în seturi de date – datasets. Vuforia permite încărcarea, activarea, dezactivarea şi eliminarea dataset-urilor în runtime. O aplicaţie poate conţine mai multe dataset-uri, care pot fi interschimbate în timpul execuţiei aplicaţiei.

Prezentul proiect utilizează un set de date format dintr-o singură imagine. Aplicaţia urmăreşte şi recunoaşte doar o ţintă unică în scenă.

În calitate de *development phone* este utilizat dispozitivul Samsung Galaxy S cu sistemul de operare Android Ice-Cream Sandwich versiunea 4.0.4.

Principii de funcţionare

Pachetul unity „Image Target”, oferit de Qualcomm, conţine deja script-ul necesar pentru funcţionarea aplicaţiei. Astfel, fişierele C# cu script-ul dorit se ataşează obiectelor scenei 3D pentru a determina comportamentul acestora. Proiectul dat specifică 3 tipuri de comportament pentru obiectul cameră „ARCamera” şi 3 pentru obiectul imagine ţintă „ImageTarget”. Script-ul pentru închiderea aplicaţiei a fost ataşat obiectului „ARCamera”. Celelalte fişiere C# sunt păstrate pentru dezvoltări ulterioare.

Diagrama claselor

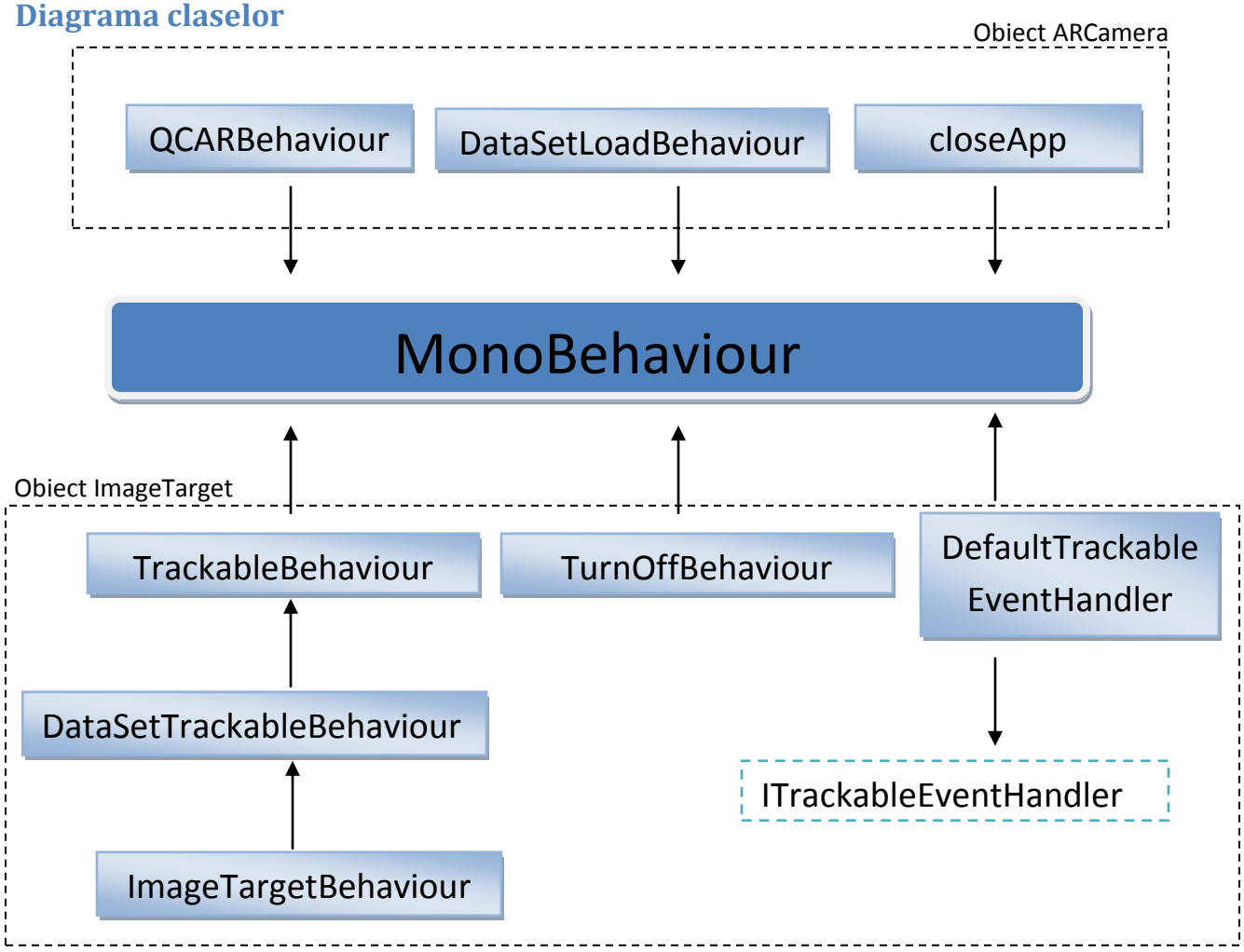


Fig. 2 Organizarea claselor

Comportament pentru ARCamera

Obiectul are ataşate următoarele fişiere:

- QCARBehaviour;
- DataSetLoadBehaviour;
- closeApp.

QCARBehaviour este responsabilă cu urmărirea imaginii ţintă şi generarea obiectului virtual pe suprafaţa acesteia. Funcţia moştenită *Update* este apelată la fiecare cadru şi actualizează scena cu noile valori de urmărire.

```
void Update()
{
    if (!mIsInitialized)
        return;

    // Get the current orientation of the surface:
    ScreenOrientation surfaceOrientation = (ScreenOrientation)getSurfaceOrientation();

    // Check if we need to update the video background configuration and projection matrix:
    if (QCAR.IsRendererDirty() || mProjectionOrientation != surfaceOrientation)
    {
        ConfigureVideoBackground();
        UpdateProjection(surfaceOrientation);
    }

    // Bind a simple material to clear the OpenGL state
    mClearMaterial.SetPass(0);

    // QCARManager renders the camera image and updates the trackables
    QCARManager.Instance.Update(mProjectionOrientation);

    // Tell Unity that we may have changed the OpenGL state behind the scenes
    GL.InvalidateState();

    // Update the camera clear flags
    UpdateCameraClearFlags();

    // Let the trackable event handlers know that all trackables have been updated
    foreach (ITrackerEventHandler handler in mTrackerEventHandlers)
    {
        handler.OnTrackablesUpdated();
    }
}
```

Fig. 3 Actualizarea scenei cu noile valori de urmărire

Clasa DataSetLoadBehaviour realizează încărcarea setului de date corepsunzător imaginii ţintă ce se doreşte a fi detectată. Awake() este, asemeni funcţiei Update(), o metodă a clasei *MonoBehaviour*.

```
void Awake()
{
    if (Application.isEditor)
    {
        return;
    }

    if (TrackerManager.Instance.GetTracker(Tracker.Type.IMAGE_TRACKER) == null)
    {
        TrackerManager.Instance.InitTracker(Tracker.Type.IMAGE_TRACKER);
    }

    if (mDataSetsToLoad.Count <= 0)
    {
        Debug.LogWarning("No data sets defined. Not loading any data sets.");
        return;
    }

    foreach (string dataSetName in mDataSetsToLoad)
    {
        if (!DataSet.Exists(dataSetName))
        {
            Debug.LogError("Data set " + dataSetName + " does not exist.");
            continue;
        }

        ImageTracker imageTracker = (ImageTracker)TrackerManager.Instance.GetTracker(Tracker.Type.IMAGE_TRACKER);
        DataSet dataSet = imageTracker.CreateDataSet();

        if (!dataSet.Load(dataSetName))
        {
            Debug.LogError("Failed to load data set " + dataSetName + ".");
            continue;
        }

        // Activate the data set if it is the one specified in the editor.
        if (mDataSetToActivate == dataSetName)
        {
            imageTracker.ActivateDataSet(dataSet);
        }
    }
}
```

Fig. 4 Încărcarea setului de date

CloseApp a fost adăugată pentru a permite utilizatorului să părăsească aplicaţia prin apăsarea butonului *Back*. De remarcat, dacă se apasă butonul *Home* aplicaţia nu este părăsită, ci doar pusă în modul *sleep*.

La fiecare frame se verifică dacă nu a fost apăsat butonul *Back*.

```
void Update () {
if (Input.GetKeyDown(KeyCode.Escape))
    {
        Application.Quit();
    }
}
```

Fig. 5 Ieşirea din aplicaţie

Comportament pentru ImageTarget

Următoarele clase sunt ataşate obiectului ImageTarget:

- ImageTargetBehaviour;
- TurnOffBehaviour;
- DefaultTrackableEventHandler.

ImageTargetBehaviour este clasa ce reprezintă imaginea ţintă. DataSetTrackableBehaviour reprezintă setul de date încărcat cu imaginile ţintă corespunzătoare, iar TrackableBehaviour este clasa de bază ce implementează detecţia şi urmărirea imaginii ţintă. TurnOffBehaviour este utilizat pentru a întrerupe generarea unui obiect virtual în timpul execuţiei. Clasa DefaultTrackableEventHandler implementează handler-ul ce tratează evenimentul de urmărire a ţintei.

Când imaginea ţintă se află în câmpul vizual al camerei, obiectul 3D este generat pe suprafaţa acesteia, în caz contrar – redarea obiectului este întreruptă.

```
private void OnTrackingFound()
{
    Renderer[] rendererComponents = GetComponentsInChildren<Renderer>();

    // Enable rendering:
    foreach (Renderer component in rendererComponents) {
        component.enabled = true;
    }

    Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " found");
}
```

Fig. 6 Depistarea ţintei

```
private void OnTrackingLost()
{
    Renderer[] rendererComponents = GetComponentsInChildren<Renderer>();

    // Disable rendering:
    foreach (Renderer component in rendererComponents) {
        component.enabled = false;
    }

    Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " lost");
}
```

Fig. 7 Pierderea ţintei

Extinderea realităţii

Principiul operaţional este foarte simplu:

- se realizează captura video;
- se adaugă obiectele 3D pe scenă;
- se afişează cadrul obţinut într-un flux video.

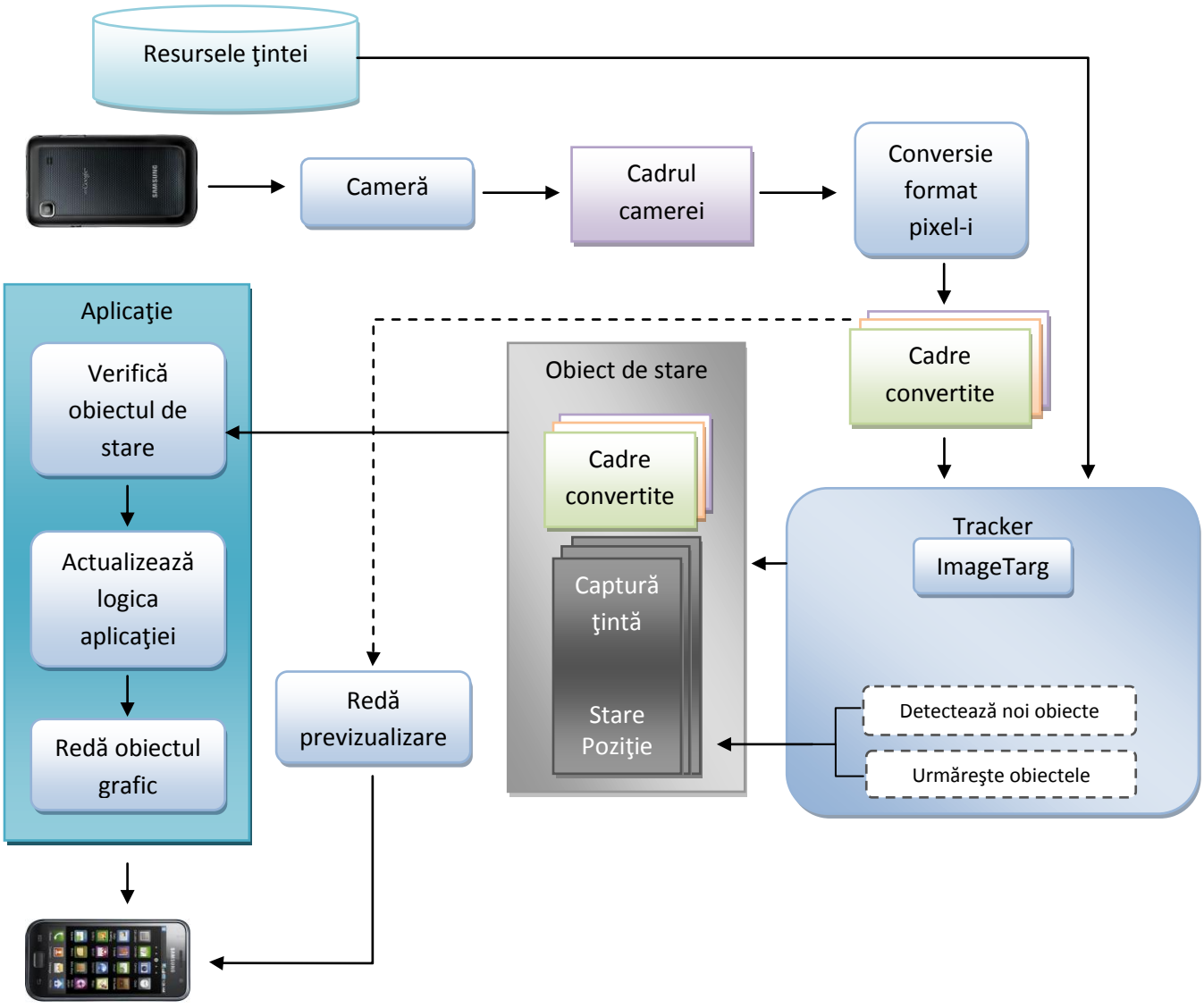


Fig. 8 Principiul „extinderii realităţii” cu Vuforia

Obiectul cameră asigură că fiecare cadru video este captat şi transmis către tracker. Cadrul este transmis în mod automat, într-un format şi dimensiune dependente de dispozitiv. Convertorul de imagine este necesar pentru a converti din formatul cameră(cum ar fi, YUV12) într-un format acceptabil pentru redarea OpenGL ES(de exemplu, RGB565) şi pentru urmărire. Această conversie, de asemenea, include subeşantionarea pentru ca imaginile furnizate de cameră să fie disponibile în diverse rezoluţii în stiva cadrelor convertite. Tracker-ul conţine algoritmii ce permit detecţia şi urmărirea obiectelor reale în cadrele video. Rezultatele recunoaşterii sunt păstrate într-un obiect de stare, folosit de către modulul ce generează imaginea. Acest modul redă imaginea ce trebuie să apară pe ecranul terminalului.

Pentru fiecare cadru procesat, obiectul de stare este actualizat şi este apelată metoda de generare a conţinutului grafic. Întregul proces se rezumă la paşii:

1. verificarea obiectului de stare pentru apariţia unor imagini detectate sau actualizări ale imaginilor(schimbarea poziţiei);
2. actualizarea logicii cu noi date de intrare;
3. redarea graficii augmentate.

Desfăşurarea proiectului

Alegerea şi crearea imaginii ţintă

După cum a fost menţionat în descriere, imaginea ţintă împreună cu setul de date necesar pentru detecţie şi urmărire este creat prin intermediul sistemului celor de la Qualcomm - Target Management System. Dimensiunea ţintei se referă la dimensiunea reală a imaginii în unităţi ale scenei 3D. Această dimensiune este foarte importantă, întrucât poziţionarea lumii virtuale va fi realizată respectând aceeaşi scară. Astfel, dacă modelul are în lăţime 16 unităţi, atunci, la deplasarea camerei din partea stângă a modelului spre dreapta, obiectul virtual îşi va modifica poziţia cu 16 unităţi.

Dimensiunea actuală a imaginii imprimate se recomandă a fi cuprinsă între 20 şi 100 cm însă, sunt posibile şi valori mai mari sau mai mici. De asemenea, se recomandă ca dimensiunea ţintei, în unităţi ale scenei 3D, să fie raportată la dimensiunea ţintei imprimate. Astfel, se poate presupune că unei unităţi îi corespunde 1 centimetru din imaginea imprimată.

Pentru a beneficia de performanţele pe care Vuforia le poate oferi, este necesar să fie ales un model ţintă cât mai bun, uşor detectat şi urmărit de către aplicaţie. Pentru a obţine o ţintă ideală, sunt recomandate imaginile:

- Bogate în detalii(o scenă din stradă, un grup de persoane, colaje şi amestecuri de obiecte, scenă din sport, etc.).
- Cu un contrast bun, dispunând atât de regiuni întunecate cât şi de regiuni luminoase.
- Bine iluminate.
- Care nu au zone şterse în luminozitate sau culoare.
- Care nu au structuri repetitive cum ar fi un câmp cu iarbă, faţada unui bloc, tabla de şah ş.a.

Pentru imprimare, se recomandă ca rezoluţia imaginii să depăşească 200-300 dpi. Ţinta trebuie să aibă aceleaşi proporţii cu ale imaginii imprimate şi să fie într-o oarecare măsură identică cu aceasta, în sensul că luminozitatea, contrastul ş.a.m.d. să fie corelate. Pentru crearea ţintelor, sunt acceptate imagini în format .png sau .jpg mai mici de 2MB.

Crearea modelelor se poate face fie într-un nou proiect, fie într-un proiect existent. Pentru noua ţintă se specifică numele, lăţimea şi tipul – „Single Image”. Celelalte două sunt „Multi Target” – reprezintă mai multe imagini combinate din care se creează o ţintă unică. Mutli Target este folosit în cazul în care imaginea model reprezintă un obiect tridimensional.

În figura 9 este prezentată fereastra pentru setarea numelui, tipului şi dimensiunii modelului.

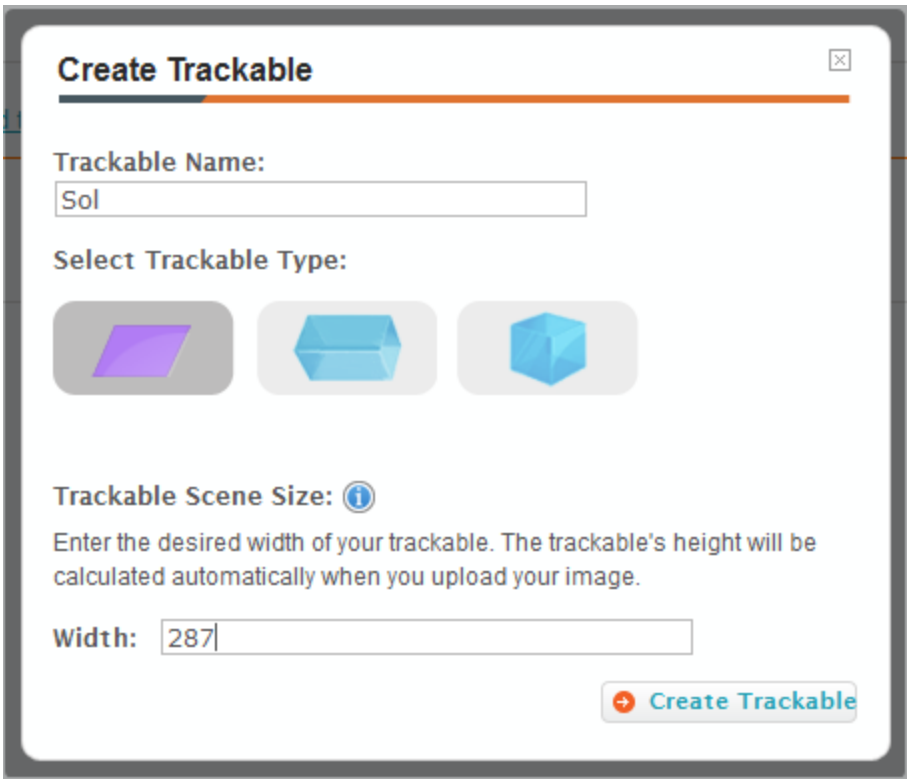


Fig. 9 Crearea imaginii ţintă

După încărcarea imaginii, sistemul o analizează şi întoarce setul de date necesar pentru detecţia şi urmărirea ei. Calitatea ţintei, din punctul de vedere al recunoaşterii şi detecţiei, este apreciată pe o scară de 5 stele. Figura 10 prezintă o imagine evaluată cu 5 stele.

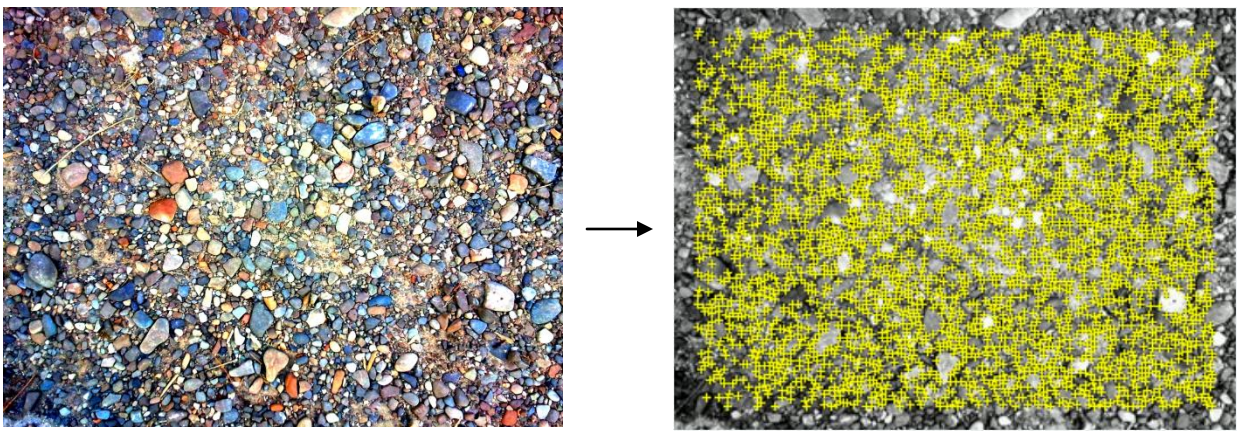
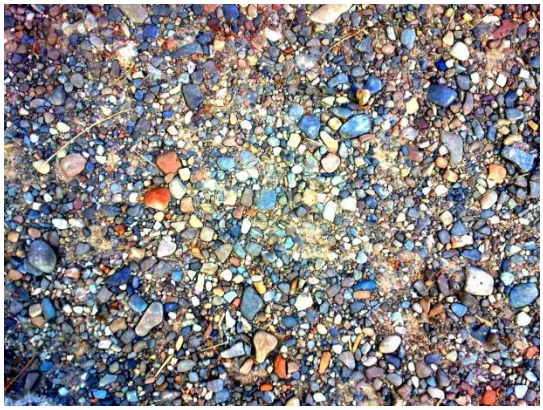


Fig. 10 Imaginea şi formele de recunoaştere şi detecţie

Pentru a ajunge la acest rezultat au fost necesare câteva transformări asupra imaginii originale.

Pentru comparaţie, în figura 11 este prezentată diferenţa între cele două imagini.

Imagine originală



Imagine obţinută în urma transformărilor

Fig.11 Comparaţie: imagine originală – imagine transformată

Au fost aplicate următoarele transformări, în ordinea în care urmează:

- Mărire contrast;
- Mărire luminozitate;
- Micşorare midtone;
- Mărire saturaţie;
- Micşorare hue;
- Primele 3 transformări pentru imaginea obţinută.

Acum se poate descărca imaginea ţintă în format pachet unity(.unitypackage).

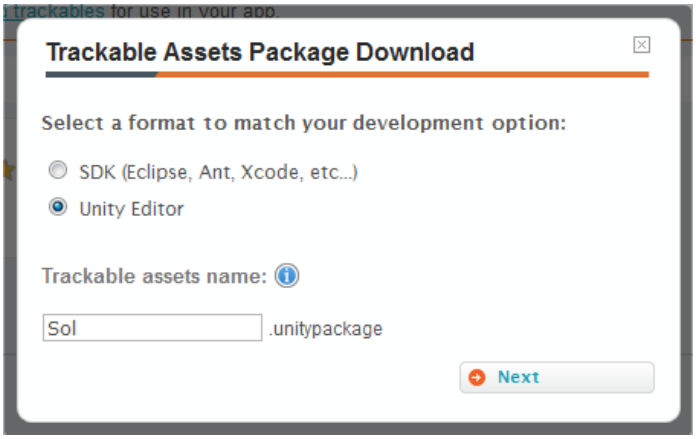


Fig. 12 Descărcare imagine ţintă

Modelarea lumii virtuale

Obiectul 3D este o altă provocare a aplicaţiilor de realitate augmentată. Dispozitivele mobile nu utilizează librăria grafică OpenGL standard, ci versiunea ES. Aceasta impune ca obiectele 3D să fie triangulate, în caz contrar ele nu vor fi redate. Triangularea reprezintă procesul de divizare a tuturor poligoanelor în triunghiuri. Blender ştie să facă acest lucru. O altă problemă care poate să apară este legată de utilizarea texturilor prost dimensionate, întrucât majoritatea terminalelor mobile recunosc doar pătrate şi dreptunghiuri 2x1 sau 1x2 pentru fişierele texturi. Astfel, 512x512, 32x16 şi 64x128 sunt dimensiuni acceptate, în timp ce 1024x16 sau 512x64 – nu. O altă incomoditate legată de texturi este faptul că există limite pentru numărul total de fişiere texturi ce pot fi încărcate în acelaşi timp de o aplicaţie şi, la fel, există limite pentru dimensiunea lor(1024, 512, şi 256 pixeli).

Lumea virtuală constă dintr-un obiect 3D ce reprezintă o casă din buşteni.

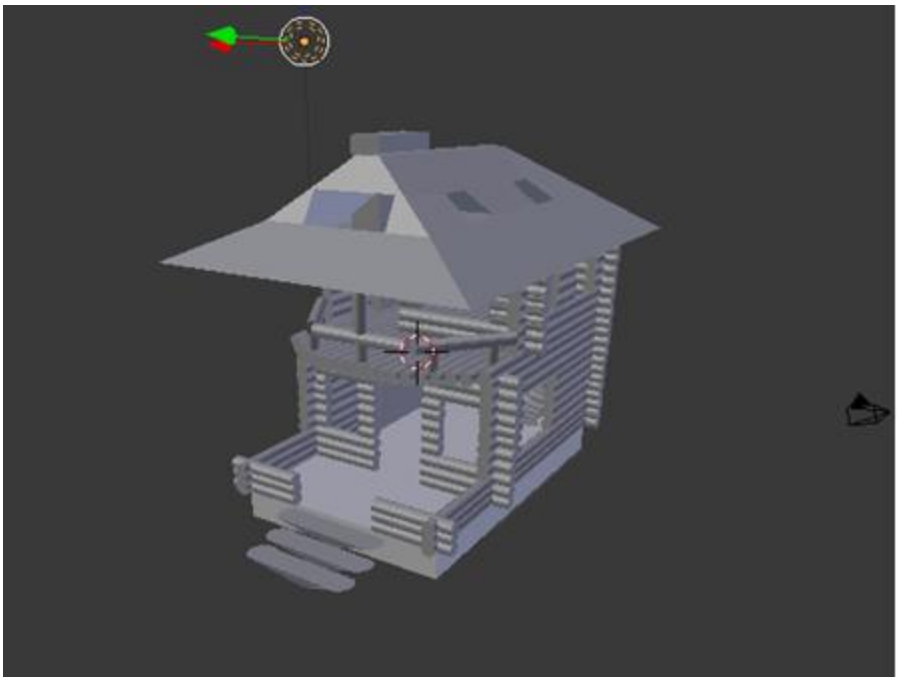


Fig. 13 Obiectul 3D modelat cu Blender

Aplicaţia pentru realitate augmentată

La instalarea extensiei Vuforia pentru Unity3d vor fi create 7 fişiere noi:

- **vuforia-android-xx-yy-zz.unitypackage:** extensia QCAR de bază;
- **vuforia-imagetargets-android-xx-yy-zz.unitypackage:** exemplu de aplicaţie cu imagini ţintă;
- **vuforia-framemarkers-android-xx-yy-zz.unitypackage:** exemplu de aplicaţie ce utilizează marker-i;
- **vuforia-multitargets-android-xx-yy-zz.unitypackage:** exemplu de aplicaţie ce utilizează *Multi Targets*;
- **vuforia-virtualbuttons-android-xx-yy-zz.unitypackage:** exemplu de aplicaţie ce implementează butoane virtuale;
- **vuforia-backgroundtextureaccess-android-xx-yy-zz.unitypackage:** exemplu ce utilizează texturi de fundal cu secvenţă video;
- **vuforia-occlusionmanagement-android-xx-yy-zz.unitypackage:** aplicaţie ce implementează efectul de ocluzie.

Pentru proiectul de faţă a fost folosit exemplul de aplicaţie cu imagine ţintă. Astfel, pachetul unity respectiv este încărcat în proiect, iar apoi se înlocuiesc sau se şterg componentele neutilizate.

Figura 14 prezintă modul în care se adaugă pachetele necesare într-un proiect „.unity”.

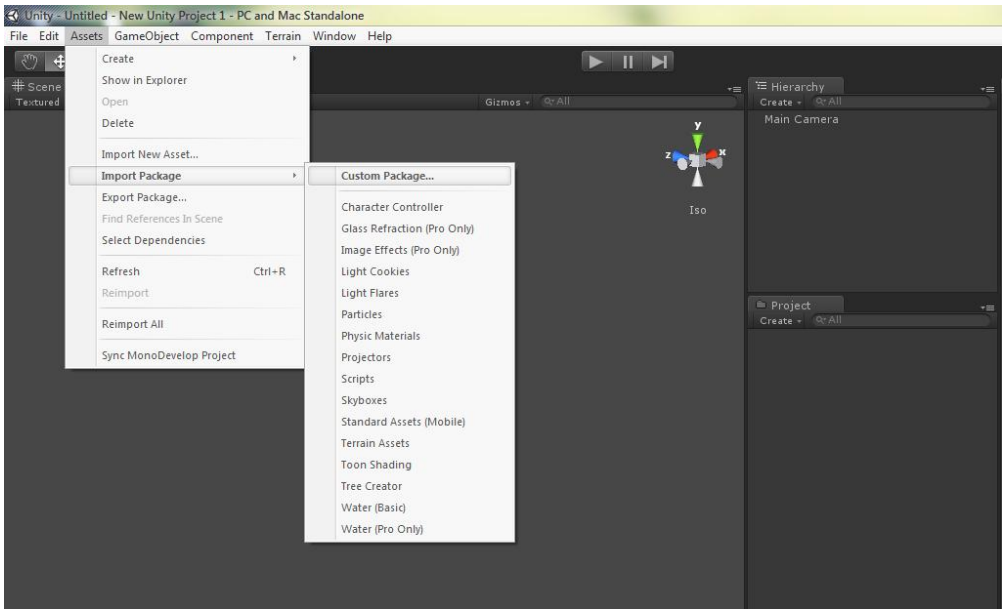


Fig. 14 Adăugare pachete în proiect unity

În continuare se înlocuieşte camera principală cu camera QCAR, se adaugă obiectul *ImageTarget* şi lumea virtuală care se doreşte să apară peste imaginea ţintă. Primele două se găsesc în directorul „Qualcomm Augmented Reality” -> „Prefabs” din cadrul proiectului, în timp ce lumea virtuală trebuie încărcată în proiect. Adăugarea obiectului 3D se face într-un mod asemănător cu adăugarea pachetului unity.

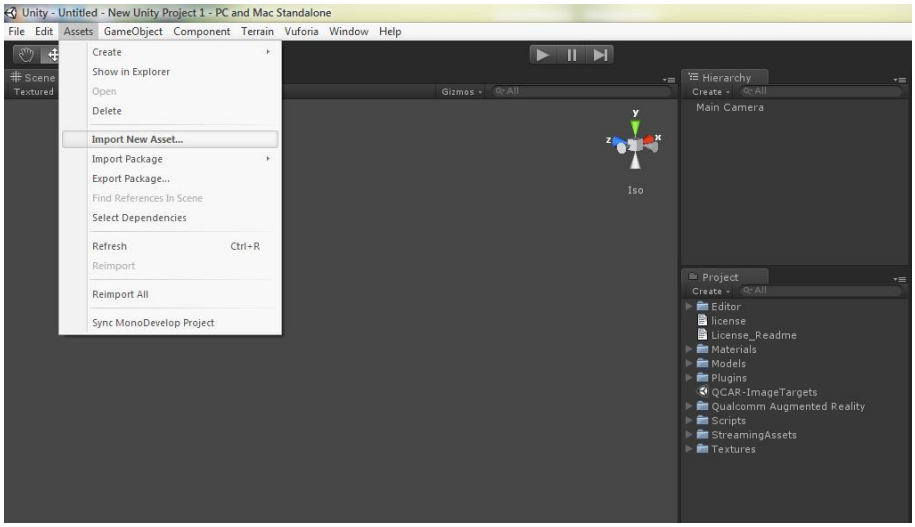


Fig. 15 Adăugare obiect 3D în proiect unity

Apoi, prin manevre drag-and-drop se construieşte ierarhia proiectului, descrisă mai sus. De remarcat faptul că va fi necesară şi adăugarea pachetului unity ce conţine setul de date pentru imaginea ţintă creată.

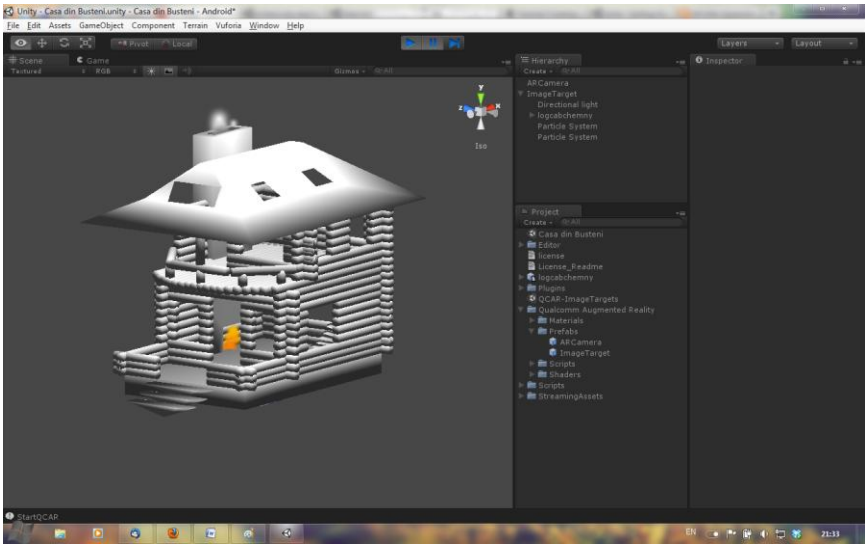


Fig. 16 Creare ierarhie proiect

Este important să fie adăugat un nou obiect numit „Directional Light” (din meniul GameObject->Create Other) pentru a ilumina suprafaţa imaginii ţintă, altfel ea va fi întunecată şi nu va corespunde cu imaginea ţintă imprimată.

Imaginea ţintă trebuie să fie părinte pentru toate obiectele 3D ce vor fi generate peste modelul ţintă imprimat.

Setările proiectului unity

Setarea corespunzătoare a proiectului este la fel de importantă precum şi paşii descrişi mai sus. O setare incorectă nu va genera obiectul virtual pe suprafaţa imaginii ţintă imprimate.

Se va selecta obiectul *ImageTarget* din ierarhie şi se vor seta din fereastra *Inspector* următoarele:

- în secţiunea Image Target Behaviour, Data Set cu setul de date încărcat;
- în secţiunea Image Target Behaviour, Image Target cu imaginea ţintă din cadrul dataset-ului încărcat.

Din ierarhie, se va selecta obiectul ARCamera şi se vor realiza următoarele:

- în secţiunea Data Set Load Behaviour, se va selecta un set de date existent pentru *Activate Data Set*.

Pe lângă aceste setări, pot fi şi altele, precum *Camera Device Mode*, *Synchronous Video* ş.a.

Acum se poate trece la faza de compilare şi execuţie, care va crea fişierul *.apk* şi va instala aplicaţia pe dispozitiv.

Prezentarea rezultatului

După ce imaginea ţintă a fost imprimată(poate fi găsită în anexă), se poate lansa în execuţie aplicaţia. Camera telefonului trebuie orientată către imaginea imprimată. Ca efect, pe ecranul telefonului se observă obiectul 3D modelat, adică o casă din buşteni.

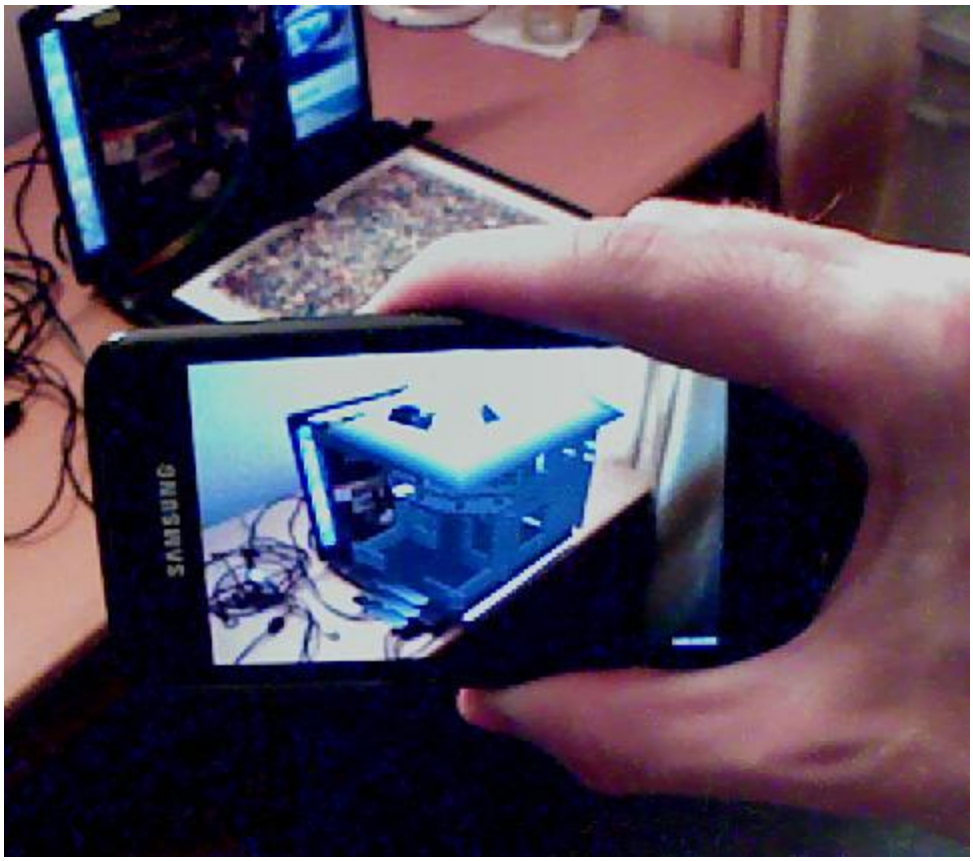


Fig. 17 Rezultatul obţinut

Concluzie

Realizarea unei aplicaţii ce implementează realitatea augmentată implică un proces complex prin care se specifică fiecare detaliu privind extinderea realităţii. În general, acest proces poate fi divizat în 3 etape:

1. setarea imaginii ţintă, fie ea o imagine oarecare sau un marker;
2. modelarea anturajului virtual;
3. descrierea funcţionalităţii – prezentul proiect utilizează limbajul C#(iar Vuforia pentru SDK-ul Android foloseşte Java).

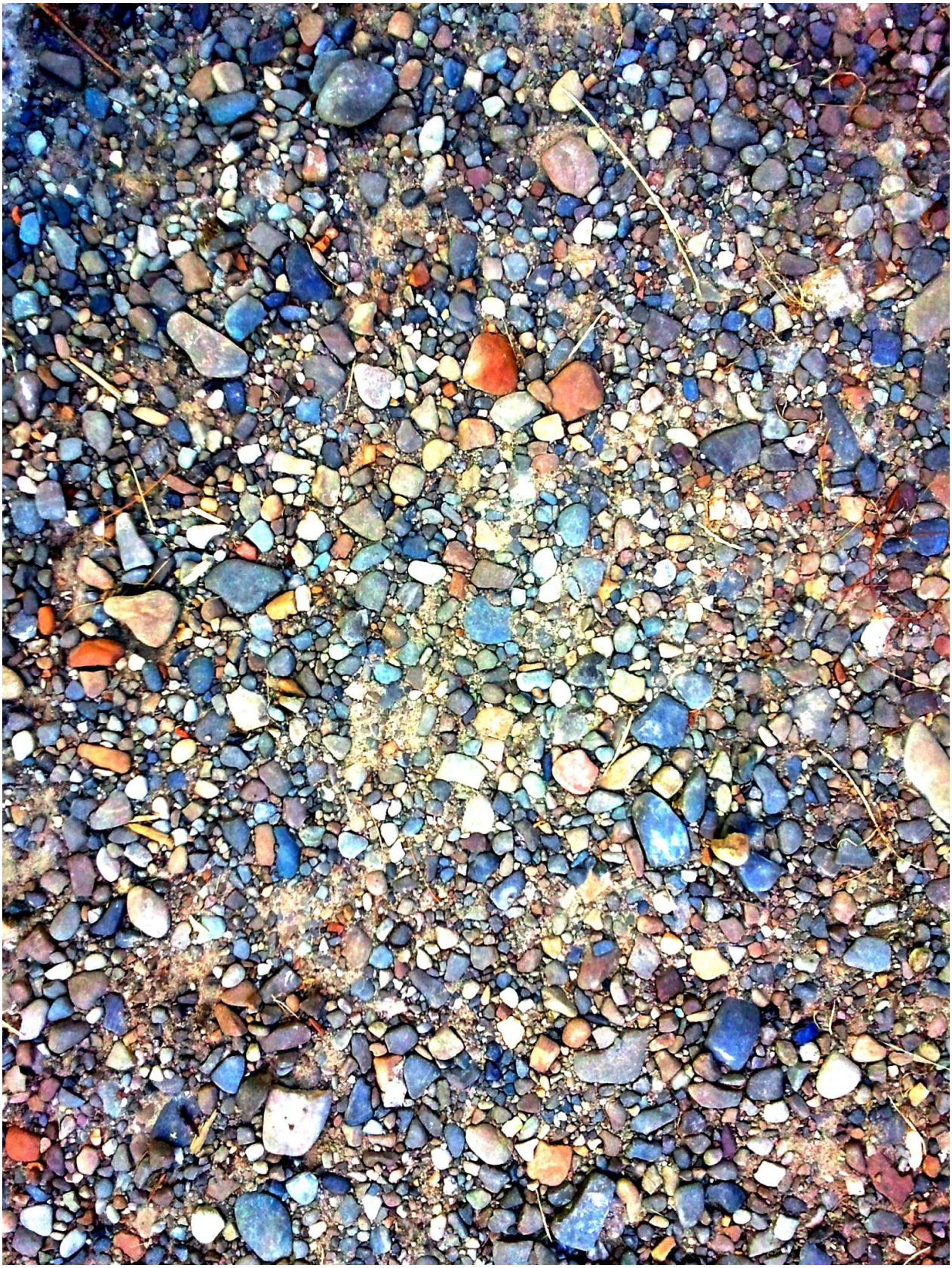
O provocare este realitatea augmentată pe dispozitivele mobile, întrucât acestea recunosc şi pot reda doar un anumit gen de modele grafice – cele triangulate, discutate în cadrul documentaţiei date.

Scopul aplicaţiei realizate a fost de a face o introducere în domeniul realităţii augmentate pe telefoane cu Android. Astfel, realizarea este departe de a fi perfectă(aici este menţionat şi obiectul 3D), eventuale îmbunătăţiri vizând modelarea lumii virtuale, adăugarea a noi funcţionalităţi aplicaţiei precum implementarea unor butoane virtuale, utilizarea mai multor imagini ţintă în cadrul aceleiaşi activităţi Android cu scopuri diferite ş.a.

Realitatea augmentată este o tehnologie aflată în etapa de dezvoltare. Din acest motiv, încă nu au fost elaborate standarde care să specifice modul în care utilizatorii interacţionează cu aplicaţia şi cum să fie realizată augmentarea în fiecare din domeniile de activitate umană. Interacţiunea utilizatorului cu realitatea augmentată este o altă problemă cu care se confruntă dezvoltatorii de aplicaţii. Deja există blogger-i care au ajuns în „valea dezamăgirii” precum Illya Vedrashko, care a concluzionat în analiza sa asupra microsite-urilor despre realitate augmentată că „wow” a degradat rapid la „meh”, şi David Klein, care a abordat tema ciudăţeniei sociale când un telefon este ţinut în faţă pentru câteva minute, în postarea de pe blog-ul său „You Look Ridiculous: The Other Augmented Reality Issue”. Totuşi, viitorul realităţii augmentate este unul luminos, având în vedere ritmul cu care se dezvoltă tehnologiile; aici trebuie menţionat proiectul GoogleEye al celor de la Google ce presupune realizarea unor ochelari, obişnuiţi la exterior, care să implementeze realitatea augmentată.

Apărută cu peste un deceniu în urmă, realitatea augmentată a fost disponibilă doar în laboratoarele de cercetare. Prezentul proiect demonstrează că acum această tehnologie este accesibilă unui public mai larg la costuri, relativ, reduse.

Anexă



Bibliografie

- History of Mobile AR, Christian Doppler Laboratory for Handheld Augmented Reality, https://www.icg.tugraz.at/~daniel/HistoryOfMobileAR/
- BMW Augmented Reality Workshop, BMW Research, http://www.bmw.com/com/en/owners/service/augmented_reality_workshop_1.html
- Imaginality “Building The Human Heart” demonstration, http://www.mindspacesolutions.com/imaginality/html/build_the_heart.html
- Augmented Reality Year in Review – 2010, Thomas Carpenter, http://thomaskcarpenter.com/2011/01/03/augmented-reality-year-in-review-2010
- Lester Madden, Nitin Samani “iPhone Augmented Reality Applications Report”, June 2010, Augmented Planet http://www.augmentedplanet.com/wpcontent/uploads/report/iPhoneApplicationReport_v1.pdf
- Foteini Valeonti , “Getting To Know Augmented Reality”, May 2010, http://valeonti.com/?p=331
- Feng Zhou, Henry Been-Lirn Duh, Mark Billinghurst, "Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR," Mixed and Augmented Reality, IEEE / ACM International Symposium on, pp. 193-202, 2008 7th IEEE/ACM International Symposium on
- Mixed and Augmented Reality, 2008.
- Douglas Dixon , February 2010, Augmented Reality Goes Mobile: Trends in Augmented
- Reality for Mobile Devices, http://www.manifest-tech.com/society/augmented_reality.htm
- Milgram, Paul; H. Takemura, A. Utsumi, F. Kishino (1994). "Augmented Reality: A class of displays on the reality-virtuality continuum" (pdf). Proceedings of Telemanipulator and Telepresence Technologies. pp. 2351–34.
- http://vered.rose.utoronto.ca/publication/1994/Milgram_Takemura_SPIE1994.pdf
- “Six Degrees of Freedom” Wikipedia, http://en.wikipedia.org/wiki/File:6_degrees_freedom.png
- Qualcomm AR, http://developer.qualcomm.com/ar
- Unity 3d, http://unity3d.com/unity/
- Gartner 2010 Emerging Technologies Hype Cycle, September 2010, http://blogs.gartner.com/hypecyclebook/2010/09/07/2010-emerging-technologies-hypecycle-is-here/
- Illya Vedrashko, May 2009, “Augmented Reality Microsites: First Impressions”, http://adverlab.blogspot.com/2009/05/augmented-reality-microsites-first.html
- David Klein, March 2010, “You Look Ridiculous: The Other Augmented Reality Issue”, http://gigaom.com/apple/you-look-ridiculous-the-other-augmented-reality-issue/
- The Case Against Augmented Reality, January 2010, Augmented Planet, http://www.augmentedplanet.com/2010/01/the-case-against-augmented-reality/
- http://www.readwriteweb.com/archives/code-free_augmented_reality_in_under_5_minutes_video.php