



ITIS LEONARDO DA VINCI
Indirizzo Informatica e Telecomunicazioni
Tesina

ScholAR Applicazione Android sulla Realtà Aumentata

Autori:

Davrieux Agustin
Lanzi Giacomo
Zizzi Riccardo

Professori:

Prof. Ollari Paolo
Prof. Paganuzzi Alberto
Prof. Franco Francesc'Antonio

Anno Scolastico 2015/2016

Indice

| | |
|--|-----------|
| Introduzione | 2 |
| 1 Analisi del problema | 6 |
| 1.1 Idea | 7 |
| 1.2 Tutorial | 8 |
| 1.3 Riconoscimento del laboratorio | 8 |
| 1.3.1 Marker | 8 |
| 1.4 Visualizzazione delle informazioni | 10 |
| 2 Tecnologie utilizzate | 11 |
| 2.1 Wikitude SDK | 11 |
| 2.1.1 Requisiti di Wikitude | 12 |
| 2.2 Android Studio | 12 |
| 2.3 Progettazione | 13 |
| 2.3.1 RACI | 13 |
| 2.3.2 WBS | 14 |
| 2.3.3 Diagramma di Gantt | 17 |
| 3 Realizzazione | 18 |
| 3.1 SDK Manager | 18 |
| 3.2 Configurazione Android Studio | 20 |
| 3.3 ARchitectView nella Activity | 21 |
| 3.4 Riconoscimento Client | 22 |
| 3.4.1 Riconoscimenti multipli | 23 |
| Problemi riscontrati | 24 |
| Conclusioni | 25 |
| Riferimenti bibliografici | 26 |

Introduzione

Realtà Aumentata

La *realtà aumentata* (AR) è un mezzo tramite il quale vengono aggiunte informazioni al mondo fisico. Viene applicata in ambiti differenti come ad esempio i sistemi di mappatura GPS, la manipolazione dei codici a barre e le applicazioni mobili che aggiungono oggetti virtuali al mondo reale tramite dispositivi quali gli smartglass.

Le caratteristiche principali della realtà aumentata sono le seguenti:

- combinazione fra mondo reale e virtuale: vengono mostrate informazioni come se appartenessero al mondo fisico;
- interattività con l'utenza e con l'ambiente: alcune applicazioni mostrano informazioni dipendenti dalla geolocalizzazione e dalla prospettiva fisica dell'utente;
- rappresentazione di oggetti 3D: vengono inseriti oggetti digitali nella visione del mondo reale.

Informazioni per aumentare la realtà

L'informazione può essere di qualsiasi tipo: un testo, un'immagine, un modello 3D, un file audio.

A differenza di altre tecnologie simili, come la realtà virtuale, la realtà aumentata non preclude il contatto con il mondo fisico. Ad esempio, con i più moderni smartglass è possibile avere una visione ed un udito sulla realtà completamente intaccata, ottenendo nel frattempo le informazioni digitali richieste.

La realtà virtuale, invece, prevede un display montato sulla testa dell'utente, che lo preclude completamente dal mondo reale.

Interattività delle informazioni

L'idea chiave della realtà aumentata, è quella di far corrispondere gli oggetti digitali alle loro controparti fisiche: ad esempio se un oggetto 3D venisse situato su un tavolo, è essenziale che esso rimanga immobile salvo decisione attiva dell'utente. Andando all'altro capo del tavolo, dovrà essere possibile vedere il retro dell'oggetto.

Con la realtà aumentata è anche possibile far agire gli oggetti digitali in una maniera poco realistica (es. farli lievitare).

Il problema principale della visualizzazione di questi oggetti nella realtà è la precisione nello spazio. In alcuni casi potrebbe andare bene un'imprecisione di qualche centimetro, ma nel caso l'applicazione richiedesse estrema precisione, si creerebbero dei seri problemi. Questa problematica nasce dal fatto che l'oggetto digitale deve essere renderizzato nuovamente nello spazio reale ad ogni cambio di punto di vista.

Modalità di visione della realtà

Lo strumento più diffuso per l'utilizzo della tecnologia della realtà aumentata è la fotocamera, attraverso la quale possiamo ottenere immagini del mondo reale e inserire oggetti digitali all'interno di queste. A questo punto, è necessario un software che analizzi l'immagine inquadrata, in modo da poter riconoscere dei punti caratteristici naturali o artificiali e permettere all'applicazione di reagire di conseguenza: questi punti si chiamano *marker*.

I marker hanno l'utilità di fornire informazioni. Utilizzando un marker che faccia riferimento ad un oggetto specifico, tale oggetto può essere visualizzato nel punto richiesto in maniera digitale.

Ci sono infinite possibilità per i marker. I più diffusi sono i codici QR, che possono contenere una grande quantità di informazioni e sono facili da analizzare; ma persino un volto può essere considerato marker, essendo asimmetrico e fondamentalmente unico.

Affiancato ai marker, si utilizza anche il sistema GPS (Global Positioning System), il quale consente di ottenere informazioni circa la geolocalizzazione del dispositivo che si sta usando, e potrebbe potenzialmente ridurre il numero di marker da cercare, rendendo il software più leggero e performante.

Oltre al GPS, ci sono ulteriori sensori quali i giroscopi, gli accelerometri e le bussole:

- i giroscopi forniscono informazioni sull'orientamento;

- le bussole forniscono informazioni sulla direzione in cui il dispositivo è rivolto;
- gli accelerometri misurano l'accelerazione.

Infine, esistono anche sensori che consentono l'input dell'utente, permettendo la partecipazione attiva all'esperienza della realtà aumentata. In un telefono, potrebbe essere il touch screen, o un pulsante. In un dispositivo come gli smartglass o nella realtà virtuale, invece, persino dei gesti con le mani possono essere utilizzati per mandare un segnale di input, a cui l'applicazione reagirà.

Applicazione

Viene utilizzata in numerosi campi:

- *Industria*: Permette di semplificare ogni passo del ciclo di vita del prodotto, la progettazione, la formazione dei lavoratori e la manutenzione.
- *Istruzione*: La combinazione di realtà fisica e virtuale consente un maggiore coinvolgimento, tramite il quale l'apprendimento è facilitato.
- *Turismo*: Proprio come la nostra applicazione, nel settore del turismo è possibile sfruttare la realtà aumentata per esplorare luoghi e ottenere informazioni a riguardo. Non solo, ma anche l'utilizzo della realtà aumentata per migliorare l'esperienza di chi visita (ad esempio, il *Royal Ontario Museum* utilizza la realtà aumentata per aggiungere la carne agli scheletri dei dinosauri).
- *Medicina*: I vantaggi del suo utilizzo comprendono la possibilità di ottenere maggiore accuratezza e precisione con minori rischi, la possibilità di diagnosticare le condizioni del paziente durante l'intervento chirurgico e di effettuare chirurgia guidata in minor tempo. Grazie alla visualizzazione 3D, la realtà aumentata può aiutare i medici ad effettuare interventi chirurgici meno invasivi, fornendo un'immagine anatomica del punto su cui intervenire.
- *Marketing*: La realtà aumentata garantisce una maggior sicurezza nel marketing. Alcuni negozi di abbigliamento hanno cominciato ad utilizzare i camerini virtuali, tramite i quali è possibile provare i vestiti come se fossero sovrapposti. Il rivenditore *American Appeal* ha addirittura consentito la possibilità di scegliere fra più colori e leggere le recensioni di altri clienti che hanno acquistato il prodotto precedentemente. Fuori

Introduzione

dal settore della moda, la realtà aumentata consente di "provare" un prodotto prima di acquistarlo: *IKEA*, ad esempio, ha sfruttato la tecnologia della AR per situare i mobili in una stanza, ed averne un primo assaggio senza necessariamente acquistare.

Capitolo 1

Analisi del problema



- A** Avvio tutorial dell'applicazione
- B** Visualizzazione dell'ambiente con la fotocamera
- C** Identificazione marker e sovrapposizione media

Figura 1.1: Funzionamento dell'applicazione

1.1 Idea

ScholAR nasce dall'idea di introdurre una nuova tecnologia nel nostro istituto. Dopo un periodo di ricerca del problema da svolgere come tesina, abbiamo valutato che fosse necessario un miglioramento dell'esperienza di orientamento formativo effettuato nelle scuole superiori.

Avendo analizzato le potenzialità offerte dalle tecnologie attuali, un'ottima idea sarebbe stata aggiungere contenuti multimediali alle attività di orientamento. Per effettuare ciò è stato deciso di utilizzare le potenzialità offerte dalla realtà aumentata, adatte al nostro scopo.

L'idea è stata quella di progettare un'applicazione per smartphone che consentisse agli studenti, partecipi dell'orientamento, di visualizzare dei brevi filmati che illustrassero il funzionamento dei vari laboratori. Questa applicazione permetterebbe all'utente di visualizzare i contenuti multimediali mentre effettua la visita della scuola.

Grazie alla realtà aumentata, infatti, è possibile disporre degli oggetti all'interno dell'istituto in modo che vengano riconosciuti automaticamente dallo smartphone dell'utente.

Il funzionamento dell'applicazione è stato pensato nel seguente modo: lo studente entrerà all'interno dell'istituto e troverà un apposito cartello con il codice *QR* da inquadrare per scaricare l'applicazione dallo store; una volta scaricata, potrà iniziare la sua visita guidata all'interno dell'istituto e, quando si avvicinerà ai laboratori, l'applicazione tramite la fotocamera riconoscerà i cartelli situati davanti alle porte e caricherà in automatico il contenuto multimediale inerente al laboratorio inquadrato; visualizzato il video potrà continuare il suo percorso per la scuola e visualizzare altri contenuti di altri laboratori.

Il vantaggio principale introdotto da questo modello è il maggiore coinvolgimento dello studente nelle attività di orientamento, il quale presta una maggiore attenzione e grazie ai video può vedere il funzionamento delle strutture didattiche dell'istituto.

La prima decisione da prendere è stata la piattaforma su cui sviluppare l'applicazione. E' stato deciso lo sviluppo in Android dato che durante il percorso di studio era stato affrontato e approfondito.

1.2 Tutorial

Per facilitare l'uso dell'applicazione si è pensato di introdurre un tutorial al primo avvio. Il tutorial è composto da 3 pagine che spiegano nel modo più esaustivo il funzionamento (figura 1.2).



Figura 1.2: Tutorial iniziale dell'applicazione

1.3 Riconoscimento del laboratorio

Il riconoscimento del laboratorio in cui si trova l'utente avviene attraverso i marker.

Ogni marker possiede un identificativo, al quale è associato uno o più contenuti multimediali. Dopo averlo inquadrato, l'informazione virtuale corrispondente al codice viene sovrapposta alla realtà.

1.3.1 Marker

Per il problema si è ideato un marker corrispondente a un cartello nel quale è scritto il nome del laboratorio e le informazioni sull'applicazione (figura 1.3). In questo modo il marker, oltre a servire per il funzionamento dell'applicazione, fornisce informazioni utili anche agli studenti.



Figura 1.3: Marker per la palazzina informatica

L’idea è stata di renderlo ben visibile e la grafica dietro al testo è stata utilizzata per renderli unici e per soddisfare le caratteristiche richieste per essere un buon marker. Tali caratteristiche sono:

- Dimensioni comprese tra 500 e 1000 pixel per ogni lunghezza. Le immagini piccole non contengono abbastanza informazioni per estrarre i punti caratteristici. Viceversa immagini troppo grandi possono non essere tracciate in modo corretto.
L’unicità, il numero e la distribuzione dei punti caratteristici sono l’elemento più importante per poter riconoscere e tracciare le immagini.
- Contrasto elevato. Le immagini con molto contrasto e strutturate in modo molto ricco e caratteristico sono le migliori per un riconoscimento del tracker.
I colori che appaiono molto contrastati all’occhio umano non sono gli stessi per il computer, dal momento che utilizza un algoritmo basato sulle scale del grigio.
- Contenuti distribuiti omogeneamente e strutture complesse. Una struttura abbastanza complessa rende le immagini uniche e facili da rilevare. Strutture ripetitive constano di informazioni grafiche ripetitive, le quali rendono più difficile il riconoscimento.
Inoltre, aree con colori singoli o transizioni leggere spesso non sono utili al riconoscimento, dal momento che non danno punti caratteristici per il tracker.

È sconsigliato l’utilizzo di:

- testi molto grossi;
- immagini con struttura ripetitiva;

- vaste aree monocromatiche;
- colori contrastanti, perché tutte le immagini sono riconosciute su scale di grigio.

1.4 Visualizzazione delle informazioni

Per la visualizzazione di video, immagini e collegamenti è stata pensata un'interfaccia semplice e intuitiva, in modo da avere una buona curva di apprendimento.

Mentre il marker è inquadrato le informazioni appaiono ancorate al marker (figura 1.4), permettendo lo spostamento dell'utente e la visualizzazione del contenuto da differenti angolature.

Appena il marker smette di essere inquadrato le informazioni si dispongono a schermo intero (figura 1.5) per permettere all'utente di visualizzarle in un formato più comodo.



Figura 1.4: Marker inquadrato



Figura 1.5: Marker non inquadrato

Capitolo 2

Tecnologie utilizzate

2.1 Wikitude SDK

Il primo SDK utilizzato è stato *ARtoolkit*, un kit di sviluppo open source. Questo risultò molto complicato e con una documentazione abbastanza ridotta, il che non facilitò il lavoro.

Così si è passati a *Vuforia*, un SDK disponibile in versione gratuita ma orientato alla sovrapposizione di oggetti 3D, che non era necessario per il progetto. Dopo aver sperimentato con i precedenti, Wikitude, considerato tra i migliori al momento e utilizzato da più di 10.000 applicazioni, è risultato ottimale per le necessità del progetto. Il suo SDK permette di riconoscere immagini 2D e sovrapporre video, testi, immagini e collegamenti a risorse esterne.

Wikitude SDK è basato su tecnologie web (HTML, JavaScript e CSS) e ogni esperienza virtuale viene chiamata *Architect World*: esse consistono in pagine HTML che utilizzano API per sovrapporre oggetti virtuali alla realtà.

Essendo basato su tecnologie web, l'Architect World può essere implementato su altri sistemi, come iOS o Windows phone.

Il framework, illustrato nella figura 2.1, permette differenti approcci per la creazione di applicazioni basate sulla realtà aumentata. Quello adottato per il progetto è stato lo sviluppo di un'applicazione Android e la creazione degli oggetti virtuali utilizzando le API JavaScript.

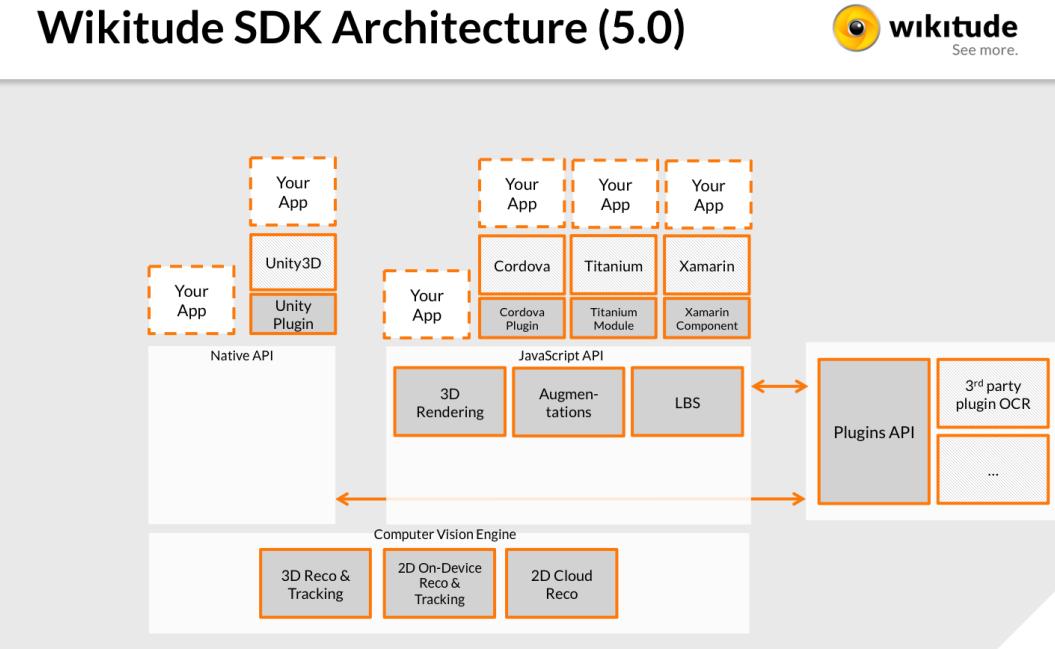


Figura 2.1: Architettura Wikitude SDK

2.1.1 Requisiti di Wikitude

Wikititude SDK supporta i dispositivi che soddisfano questi requisiti:

- Android 4.0.3+ (API 15+)
- Device ad alta risoluzione (hdpi)
- Fotocamera posteriore
- CPU sufficiente, ad esempio:
 - Samsung Galaxy S2+
 - Nexus 4+
 - Nexus 10+

2.2 Android Studio

Android Studio fornisce tutto ciò che serve per iniziare a sviluppare applicazioni Android (Ambiente di sviluppo + SDK). Per installarlo basta scaricarlo

dal sito ufficiale [4].

Una volta completato il download, eseguire e completare l'installazione guida-
ta; in particolare fare attenzione a spuntare l'opzione ‘Android SDK’ quando
ci viene chiesto che componenti installare.

2.3 Progettazione

Per l'assegnazione dei ruoli, le attività del progetto e la suddivisione del
lavoro in un tempo ben definito si è deciso di stilare la tabella *RACI*, le varie
strutture *WBS* e il diagramma di *Gantt*.

In questo modo abbiamo ottenuto un lavoro diviso equamente e il lavoro di
gruppo è risultato leggero con ottimi risultati.

2.3.1 RACI

La RACI o matrice delle responsabilità (figura 2.2) è la rappresentazione
tabellare che identifica i ruoli e le responsabilità per ogni attività. I ruoli
previsti sono:

- *Responsible* (R), persona coinvolta attivamente nell'attività e respon-
sabile della parte operativa;
- *Accountable* (A), responsabile dell'attività e approva il lavoro dei ruoli
R, solitamente il ruolo è unico per l'attività;
- *Consulted* (C), aiuta e collabora con R per l'esecuzione dell'attività;
- *Informed* (I), persona che deve essere informata sullo stato dell'attività.

| | Progettazione | Interfaccia | Programmazione Android | Programmazione JavaScript | Verbalizzazione |
|------------------|---------------|-------------|------------------------|---------------------------|-----------------|
| Agustin Davrieux | A/R | R | A/R | R | R |
| Giacomo Lanzi | R | A/R | R | A/R | I/C |
| Riccardo Zizzi | R | I/C | I | I | A/R |

Figura 2.2: RACI di scholAR

2.3.2 WBS

La Work Breakdown Structure (WBS) è la rappresentazione in forma grafica della composizione gerarchica di tutte le attività di un progetto.

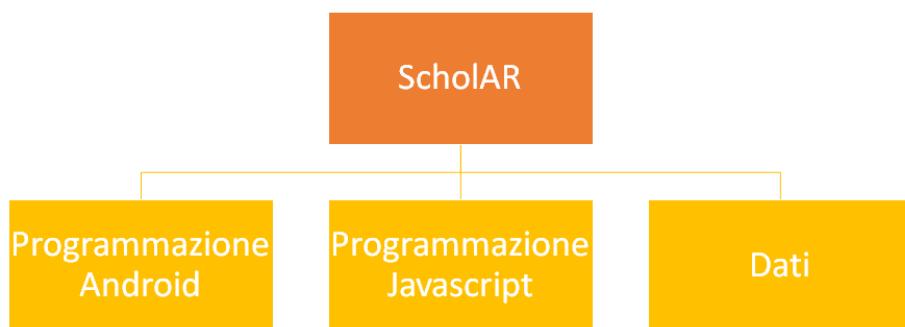
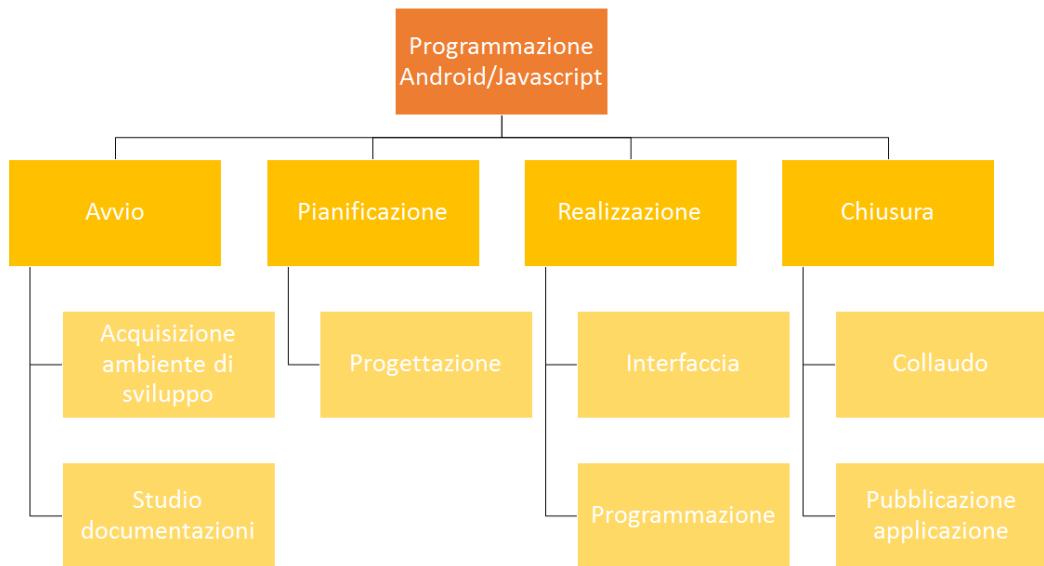


Figura 2.3: WBS del progetto

Programmazione Android e Javascript



| WBS | Descrizione | Durata (gg) |
|---------------------------|-----------------------------------|-------------|
| 1.1 AVVIO | | |
| 1.1.1 | Acquisizione ambiente di sviluppo | 1 |
| 1.1.2 | Studio documentazioni | 5 |
| 1.2 PIANIFICAZIONE | | |
| 1.2.1 | Progettazione | 6 |
| 1.3 REALIZZAZIONE | | |
| 1.3.1 | Interfaccia | 5 |
| 1.3.2 | Programmazione | 18 |
| 1.4 CHIUSURA | | |
| 1.4.1 | Collaudo | 2 |
| 1.4.2 | Pubblicazione applicazione | 1 |

Figura 2.4: WBS della programmazione Android/JavaScript

Dati



| WBS | Descrizione | Durata (gg) |
|---------------------------|-----------------------------------|-------------|
| 1.1 Pianificazione | | |
| 1.1.1 | <i>Scelta zone</i> | 1 |
| 1.1.2 | <i>Scelta contenuto mediatico</i> | 2 |
| 1.3 REALIZZAZIONE | | |
| 1.3.1 | <i>Creazione Marker</i> | 2 |
| 1.3.2 | <i>Realizzazione Media</i> | 4 |
| 1.4 CHIUSURA | | |
| 1.4.1 | <i>Collaudo</i> | 1 |
| 1.4.2 | <i>Installazione</i> | 1 |

Figura 2.5: WBS dei dati

2.3.3 Diagramma di Gantt

Il diagramma di Gantt è la rappresentazione delle tempistiche del progetto.

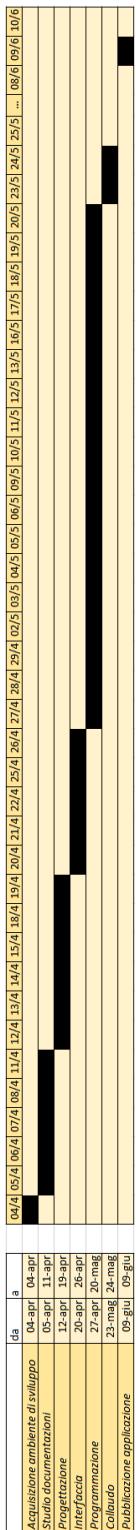


Figura 2.6: Diagramma della programmazione

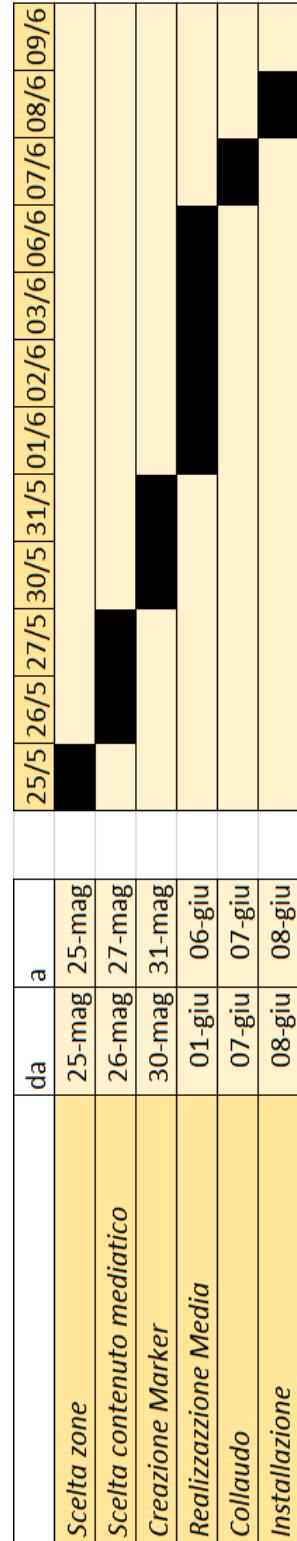


Figura 2.7: Diagramma dei dati

Capitolo 3

Realizzazione

3.1 SDK Manager

Installando Android Studio, si installa anche la versione più recente di Android SDK e i tools più rilevanti. In ogni caso, l'SDK Manager permette di installare componenti opzionali e di aggiornare l'SDK Android e i relativi tools all'ultima versione disponibile; per questo motivo è consigliabile eseguirlo periodicamente.

È consigliato eseguirlo almeno una volta dopo l'installazione di Android Studio, per essere certi di avere tutti i componenti necessari per iniziare a sviluppare in Android.

1. Dopo aver eseguito Android Studio, cliccare sull'icona dell'SDK Manager nella toolbar (figura 3.1).



Figura 3.1: Toolbar di Android Studio

2. Nella finestra di dialogo che si apre, cliccare su ‘Launch standalone SDK Manager’ (figura 3.2).

SDK Manager

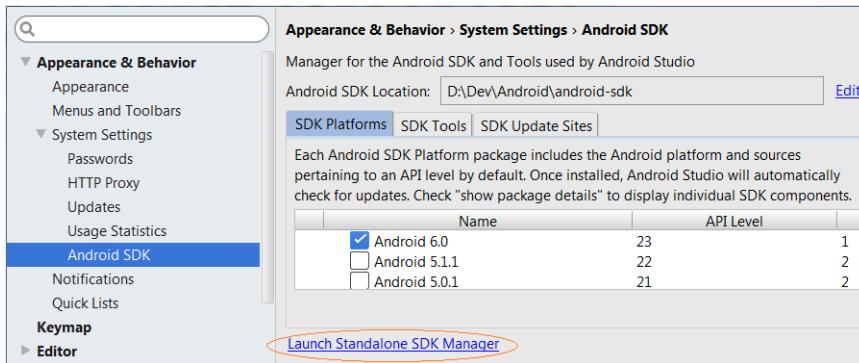


Figura 3.2: Impostazioni SDK

3. Selezionare i pacchetti da installare (figura 3.3):

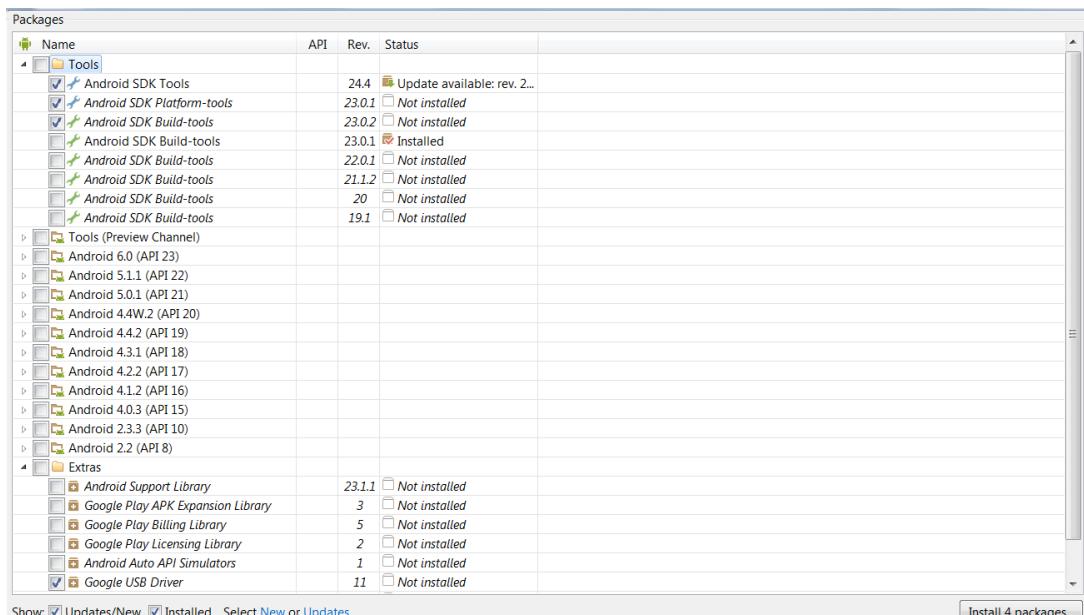


Figura 3.3: SDK Manager

- Tools:
 - Android SDK Tools;
 - Android SDK Build Tools;
 - Android SDK Platform-Tools.

- Android 6.0 (API 23)¹
 - SDK Platform;
 - Documentation for Android SDK.
4. Cliccare ‘Install X packages’.

3.2 Configurazione Android Studio

Dopo aver scaricato Wikitude SDK dal loro sito ufficiale [5] (in questo progetto viene utilizzata la versione 5.1.4 della JavaScript API) estrarre il contenuto dell’archivio e successivamente:

- creare un nuovo progetto Android (nel SDK sono inclusi progetti d’esempio nei quali non è necessario svolgere i seguenti passaggi);
- copiare il file `wikitudesdk.aar` (cartella `libs` del SDK) nella cartella delle librerie del progetto creato precedentemente (es: `nomeProgetto\app\libs`);
- aprire il file `build.gradle` e aggiungere `wikitudesdk.aar` alle dipendenze, come nel codice sottostante:

```
android {
    ...
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile (name: 'wikitudesdk', ext:'aar')
    compile 'com.android.support:appcompat-v7:21.0.3'
}

repositories {
    flatDir{
        dirs 'libs'
    }
}
```

- Aggiungere i permessi necessari nel file `AndroidManifest.xml`:

```
<uses-permission android:name="android.permission.
    INTERNET" />
<uses-permission android:name="android.permission.
    ACCESS_COARSE_LOCATION" />
```

¹Può essere utilizzata qualsiasi versione di Android compatibile con Wikitude

```

<uses-permission android:name="android.permission.
    ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.
    ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.
    ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.
    ACCESS_GPS" />
<uses-permission android:name="android.permission.CAMERA"
    />
<uses-permission android:name="android.permission.
    WRITE_EXTERNAL_STORAGE" />
<uses-feature android:name="android.hardware.camera"
    android:required="true" />
<uses-feature android:name="android.hardware.location"
    android:required="true" />
<uses-feature android:name="android.hardware.sensor.
    accelerometer" android:required="true" />
<uses-feature android:name="android.hardware.sensor.
    compass" android:required="true" />
<uses-feature android:glEsVersion="0x00020000" android:
    required="true" />
<uses-sdk android:targetSdkVersion="19" android:
    minSdkVersion="9" />

```

- L'activity con l'AR-View (denominata `architectView` nell'esempio che segue) deve avere `android:configChanges="screenSize|orientation"` nel Manifest. Per esempio:

```

<activity android:name="com.scholar.AppProva.MiaActivity"
    android:configChanges="screenSize|orientation"/>

```

3.3 ARchitectView nella Activity

Bisogna tenere a mente che Wikitude SDK non è nativo di Android come gli altri SDK. Il concetto alla base è aggiungere una `architectView` al progetto e notificarlo sugli eventi. L'architectView gestisce la fotocamera e gli eventi scatenati dai sensori.

Architect World, invece, è implementato in JavaScript e impacchettato nella cartella assets del progetto. Tutti gli eventi sono scritti in HTML e JavaScript e servono a richiamare i metodi di Wikitude.

Bisogna includere:

```
<script src="architect://architect.js"></script>
```

nei file HTML per utilizzare l'applicazione AR e `architectView` si occuperà del resto.

È consigliabile occuparsi della realtà aumentata in un'Activity separata. Dichiare `architectView` dentro al file .xml del layout. Per esempio, aggiungendo questa riga dentro al `FrameLayout`:

```
<com.wikitude.architect.ArchitectView android:id="@+id/
    architectView"
    android:layout_width="fill_parent" android:layout_height="
    fill_parent"/>
```

È fondamentale notificare `architectView` degli eventi dell'Activity (attraverso i metodi `onCreate()`, `onPostCreate()`, `onPause()`, `onResume()`, `onDestroy()`). Il metodo migliore è definire una variabile di riferimento alla `architectView` subito dopo `setContentView` nel `onCreate()` così da poter accedervi direttamente.

```
this.architectView = (ArchitectView) this.findViewById(R.id.
    architectView);
final StartupConfiguration config = new StartupConfiguration
    (* licenza *);
this.architectView.onCreate( config );
```

Dopodiché caricare l'Architect World attraverso il metodo `architectView.load()`:

```
architectView.load('FilePerAR.html')
```

L'argomento è il percorso del file HTML. Può essere situato nella cartella `assets` oppure in un server web.

3.4 Riconoscimento Client

I marker (nel nostro caso immagini) devono essere raggruppati in una `target collection`, che deve essere creata online [7] e ha un formato speciale `.wtc`. Nel file html si deve includere il file JavaScript:

```
<script src="assets/scholar.js"></script>
```

Un esempio di codice potrebbe essere:

```
// Inizializzazione ClientTracker
this.tracker = new AR.ClientTracker("assets/scholar.wtc");

// Creazione overlay per la prima pagina
var imgLabInf = new AR.ImageResource("assets/labinf.png");
var overlayLabInf = new AR.ImageDrawable(imgLabInf, 1, {
    offsetX: -0.15,
    offsetY: 0
```

```

});;

var pageLabInf = new AR.Trackable2DObject(this.tracker, ""
    labinf", {
    drawables: {
        cam: overlayLabInf
    }
});

```

Per prima cosa un `AR.ClientTracker` deve essere creato per poter riconoscere i marker. Deve essere inizializzato con un URL specifico per la target collection desiderata.

Il prossimo passo è creare l'overlay, ovvero ciò che andrà a coprire l'inquadratura della fotocamera (nel nostro caso, un'immagine o un video). Nel codice esempio, un'immagine dimostrativa viene creata e passata a `AR.ImageDrawable`. Un *drawable* è un componente visivo che può essere connesso a un target `AR.Trackable2DObject`.

`AR.ImageDrawable` è inizializzato con la risorsa e la rispettiva grandezza. I parametri opzionali possono indicare la posizione dell'immagine relativamente al target riconosciuto.

L'ultima parte combina tutto creando un `AR.Trackable2DObject` che usa il tracker precedentemente creato, il nome del target (equivale al nome del file caricato nella target collection) e il drawable da visualizzare.

3.4.1 Riconoscimenti multipli

Per multipli target si devono creare diverse risorse, drawable e pagine quanti sono i marker che si vogliono riconoscere. Ad esempio:

```

// Crea overlay per la pagina due
var imgLabSis = new AR.ImageResource("assets/labsis.png");
var overlayLabSis = new AR.ImageDrawable(imgLabSis, 0.5, {
    offsetX: 0.12,
    offsetY: -0.01
});
var pageLabSis = new AR.Trackable2DObject(this.tracker, ""
    labsis", {
    drawables: {
        cam: overlayLabSis
    }
});

```

Similmente al primo codice esempio, l'immagine sorgente è `AR.ImageDrawable` per il secondo overlay sono stati creati. `AR.Trackable2DObject` per la seconda pagina utilizza lo stesso tracker ma con un target name diverso.

Problemi riscontrati

Sono stati riscontrati diversi problemi nella realizzazione del progetto. Il prototipo è stato sviluppato più volte prima di trovare un SDK con le caratteristiche desiderate, utilizzando prima *ARtoolkit* e poi *Vuforia*.

L'intento iniziale era quello di riconoscere l'ambiente in cui l'utente si trova, ma è risultato irrealizzabile per la complessità del riconoscimento, dato che richiedeva la creazione di un modello 3D dell'ambiente.

Un altro problema è stata la visualizzazione *on demand* dei media, data la difficoltà di scaricare le informazioni relative ai laboratori da un database *MongoDB* [9] (una base di dati non relazionale orientata ai documenti).

Infine ci sono stati problemi legati al riconoscimento dei marker, nel caso le condizioni di luminosità fossero avverse.

Conclusioni

In questo progetto di tesina si è giunti alla creazione di un prototipo funzionante; nei test effettuati i marker sono stati riconosciuti nella maggior parte dei casi e non sono stati riscontrati problemi con la visualizzazione del contenuto mediatico.

L'interfaccia grafica è stata progettata in modo che possa risultare intuitiva e comoda per l'utente che utilizza l'applicazione. Inoltre, l'applicazione è stata sviluppata in modo modulare permettendo l'aggiunta o la rimozione dei media senza la necessità di essere modificata.

Uno sviluppo futuro potrebbe essere quello di creare un'applicazione per tutte le scuole superiori della provincia. Aggiungendo un meccanismo di riconoscimento della scuola, ad esempio utilizzando il GPS, i marker e i contenuti multimediali potrebbero essere caricati dinamicamente e il servizio sarebbe garantito per tutti.

C'è anche l'idea di sfruttare la portabilità della tecnologia web utilizzata da Wikitude SDK per sviluppare la stessa applicazione in iOS e Windows phone.

Bibliografia

- [1] Héctor Martínez, Danai Skournetou, Jenni Hyppölä, Seppo Laukkanen and Antti Heikkilä
Drivers and Bottlenecks in the Adoption of Augmented Reality Applications
<http://jmta.avestia.com/2014/004.html>
- [2] Ronald T. Azuma
A Survey of Augmented Reality
<http://www.cs.unc.edu/~azuma/ARpresence.pdf>
- [3] *Augmented Reality*
http://etec.ctlt.ubc.ca/510wiki/Augmented_Reality
- [4] Google
Android Studio
<http://developer.android.com/sdk/index.html>
- [5] Wikitude GmbH
Augmented Reality SDK
<http://www.wikitude.com>
- [6] Wikitude GmbH
Target Manager
<http://targetmanager.wikitude.com/>
- [7] Wikitude GmbH
Wikitude documentation for Android
<http://www.wikitude.com/external/doc/documentation/latest/android/>
- [8] MongoDB
NoSQL database
<http://targetmanager.wikitude.com/>
- [9] Paolo Ollari, Giorgio Meini, Fiorenzo Formichi
Gestione, progetto e organizzazione d'impresa