

Laboratorio di realtà virtuale e realtà aumentata
Laurea Magistrale in Informatica
Università di Bologna

Realtà Aumentata - Dress Fitting su dispositivi IOS

Keran Jegasothy - 1007457



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Contents

1	Introduzione	2
2	Progettazione	3
2.1	Funzionalità dell'applicazione	3
3	Implementazione	4
3.1	Architettura	4
3.2	Tecnologie utilizzate	4
3.3	Script	5
3.3.1	ScaleInOutXY, ScaleInOutXZ e ScaleInOutYZ	5
3.3.2	HideUnhide	5
3.3.3	ClothesSize	6
3.3.4	DestroyButton	6
4	Use Case	7
5	Commento	8
6	Schermate	9

1. Introduzione

Al giorno d'oggi, i sistemi smart hanno un impatto importante sulla vita quotidiana. Ad esempio, lo shopping online si è sviluppato rapidamente. Tuttavia, settori come quello dell'abbigliamento, non riescono a sfruttare tale sviluppo; in quanto il cliente ha la necessità di sapere come gli vestirà il capo d'abbigliamento. In particolare, un articolo scritto dalla Cornell University, Barret (2020) ha affermato che il 70 % degli indumenti acquistati online, vengono restituiti. Per poter risolvere tale problema, è necessario l'utilizzo di un *camerino virtuale*, in cui l'utente ha la possibilità di vedere la vestibilità dell'articolo prima di effettuare l'acquisto. In letteratura, sono presenti due approcci per la creazione di un camerino virtuale:

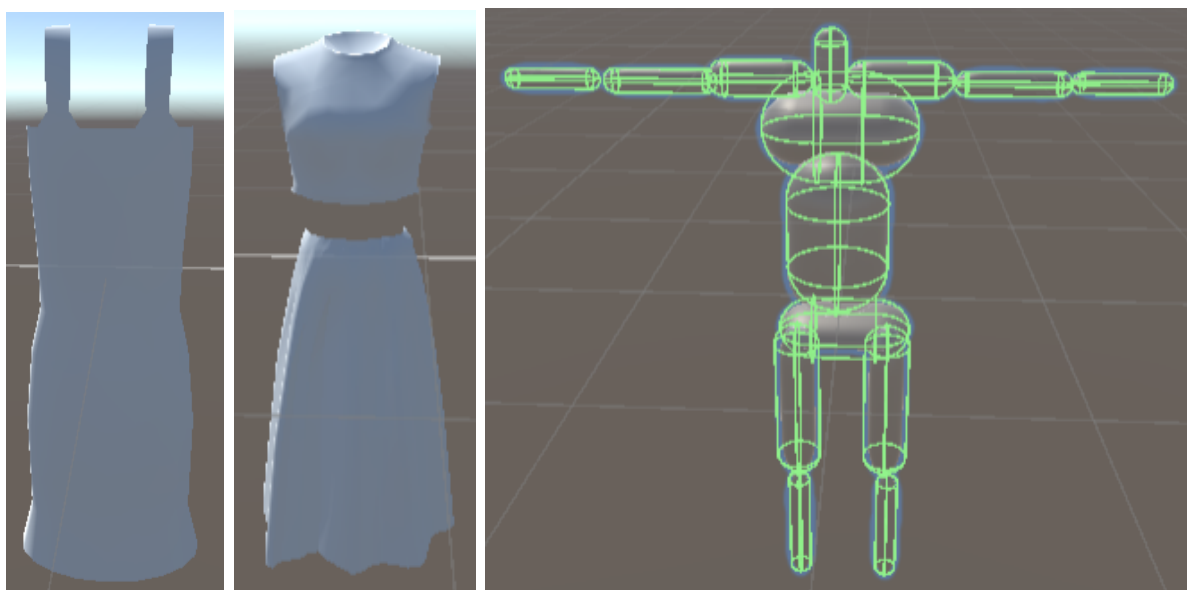
- utilizzare una fotocamera con il sensore di profondità, in modo tale da poter mappare l'utente.
- utilizzare un avatar che abbia una corporatura simile a quella dell'utente e che segua i suoi movimenti.

In questo progetto, la scelta è ricaduta sull'utilizzo del secondo approccio. Con tale approccio non è richiesto l'utilizzo di dispositivi particolari da parte dell'utente, in quanto è sufficiente uno smartphone. In particolare, il progetto consiste in un'applicazione per il sistema operativo IOS.

2. Progettazione

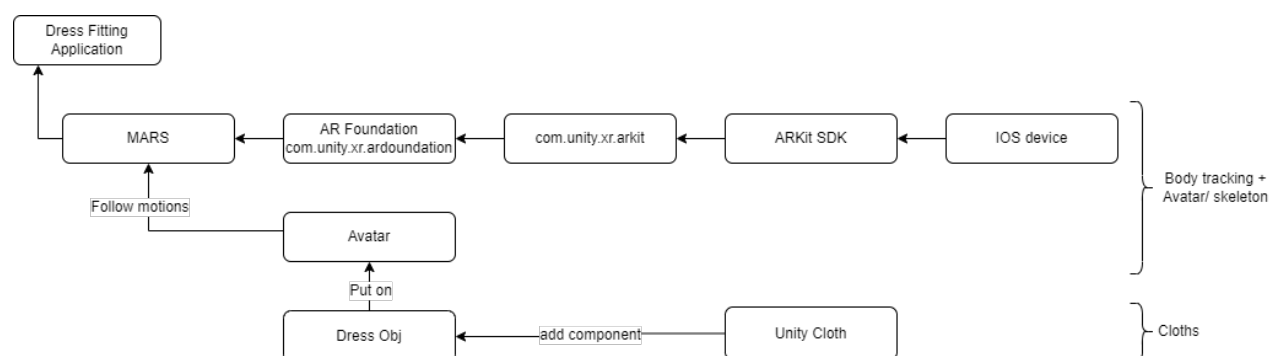
2.1 Funzionalità dell'applicazione

All'interno dell'applicazione, tramite l'apposito menù, l'utente ha la possibilità di modificare le dimensioni delle varie parti del corpo dell'avatar in modo tale che quest'ultimo abbia una costituzione fisica simile a quella dell'utente. Successivamente, l'utente ha la possibilità di provare 2 outfit. Per ciascun outfit, è possibile "indossare" diverse taglie (S, M, L, XL, XXL). In seguito sono riportati i due outfit disponibili nell'applicazione e l'avatar.



3. Implementazione

3.1 Architettura



3.2 Tecnologie utilizzate

Le tecnologie utilizzate per la realizzazione di questo progetto sono:

- **Unity** (2019.4.40): è un motore grafico multiplatforma sviluppato da Unity Technologies che consente lo sviluppo di videogiochi e altri contenuti interattivi, quali visualizzazioni architettoniche o animazioni 3D in tempo reale. Inoltre, per poter dare le proprietà fisiche corrette ai vestiti è stato utilizzato il componente *cloth* disponibile in Unity.
- **MARS**: è una libreria che mette a disposizione strumenti per lo sviluppo di applicazioni riguardanti la realtà aumentata. In particolare, mette a disposizione uno strumento per il body tracking.

In seguito vengono riportate alcune tecnologie alternative a MARS:

- **ARFoundation** (<https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/index.html>)
- **Barracuda** (<https://docs.unity3d.com/Packages/com.unity.barracuda@1.0/manual/index.html>)

3.3 Script

Per lo sviluppo delle varie operazioni eseguibili all'interno dell'applicazione, vengono utilizzati degli script (C++). In seguito è riportato l'elenco di tutti gli script:

- ScaleInOutXY
- ScaleInOutXZ
- ScaleInPutYZ
- HideUnhide
- ClothesSize
- DestroyButton

3.3.1 ScaleInOutXY, ScaleInOutXZ e ScaleInOutYZ

Gli script *ScaleInOutXY*, *ScaleInOutXZ* e *ScaleInOutYZ* permettono di modificare le dimensioni di ciascuna parte dell'avatar.

```
1 if (_ZoomIn)
2 {
3     //make a bigger object
4     Object.transform.localScale += new Vector3(Scale, Scale, 0);
5 }
6 if (_ZoomOut)
7 {
8     //make a small object
9     Object.transform.localScale -= new Vector3(Scale, Scale, 0);
10 }
```

3.3.2 HideUnhide

Lo script *HideUnhide* permette di visualizzare un determinato outfit, nascondendo l'altro.

```
1 public void Hide(){
2     Object.SetActive(false);
3 }
4 public void Unhide(){
5     Object.SetActive(true);
6 }
```

3.3.3 ClothesSize

Ogni volta che l'utente cambia la taglia dei vestiti, viene richiamato lo script *ClotheSize*, che modifica le dimensioni degli abbigliamento.

```
1 public void OnPressSDress(){
2     Object.transform.localScale = new Vector3(0.65f, 0.75f, 1f);
3 }
4
5 ...
6
7 public void OnPressXXLDress(){
8     Object.transform.localScale = new Vector3(0.85f, 0.95f, 1.05f);
9 }
```

3.3.4 DestroyButton

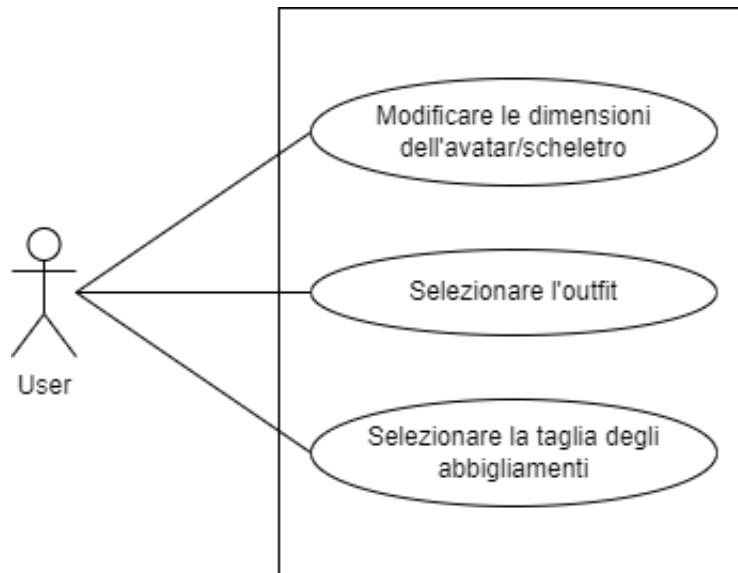
All'avvio dell'applicazione , è presente il pulsante start. Nel momento in cui l'utente tocca tale pulsante, vengono richiamati *DestroyButton* e *HideUnhide*. *DestroyButton* viene richiamato per eliminare il pulsante start, mentre *HideUnhide* permette di nascondere tutti i vestiti.

```
1 public void Destroy(){
2     Destroy(startButton);
3 }
```

4. Use Case

All'interno dell'applicazione, l'utente ha la possibilità di:

- Modificare le dimensioni dell'avatar. In particolare, è possibile modificare i parametri dei seguenti elementi: collo, petto, busto, bacino, spalla destra e sinistra, braccio destro e sinistro, avambraccio destro e sinistro, coscia destra e sinistra, gamba destra e sinistra.
- Selezionare uno tra i due outfit disponibili
- Selezionare la taglia dei vestiti



5. Commento

In seguito è riportato la lista delle possibili criticità nella soluzione proposta:

- l'applicazione non è stato testato in quanto non ho la possibilità di caricare l'applicazione su un dispositivo IOS. Il problema sorge in quanto non possiedo un PC con la possibilità di installare XCode.
- per ciascuna taglia, è difficile identificare il corretto ridimensionamento dell'oggetto che simula il vestito.
- l'avatar/scheletro deve essere uguale all'utente. Per ottenere questa uguaglianza, bisogna dare la possibilità all'utente di modificare ogni singolo punto dell'avatar. Questa operazione richiederebbe un dispendio di tempo esagerato per l'utente.

Come già detto nell'introduzione, in letteratura, sono presenti due approcci per poter creare un camerino virtuale:

- tramite l'utilizzo di fotocamere particolari con sensori di profondità
- tramite l'utilizzo di un avatar

Tenendo in considerazione sia l'andamento del mercato che le criticità del secondo approccio, si può concludere che il primo approccio può essere quello più interessante.

Andamento del mercato: Nell'ambito IOS, i dispositivi top di gamma (iPhone 12 pro, iPhone 12 pro max, iPhone 13 pro e iPhone 13 pro max) hanno già il Lidar. Mentre nell'ambito Android, i top di gamma hanno il sensore ToF.

Sia il Lidar che il ToF permettono di scannerizzare oggetti 3D, quindi con l'utilizzo di tali sensori, si potrà creare l'avatar automaticamente tramite una scansione 3D eseguita sull'utente. In tal modo, si può risolvere l'ultima criticità della lista soprastante.

6. Schermate

