



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

PROGETTO DI TESI TRIENNALE

«MODELLAZIONE MULTIMODALE DI
AMBIENTI INDOOR E TESTA UMANA
PER APPLICAZIONI DI REALTA'
AUMENTATA»

Laureandi: Giulio Meda, Leonardo Egidati

Relatore: Prof. Michele Geronazzo

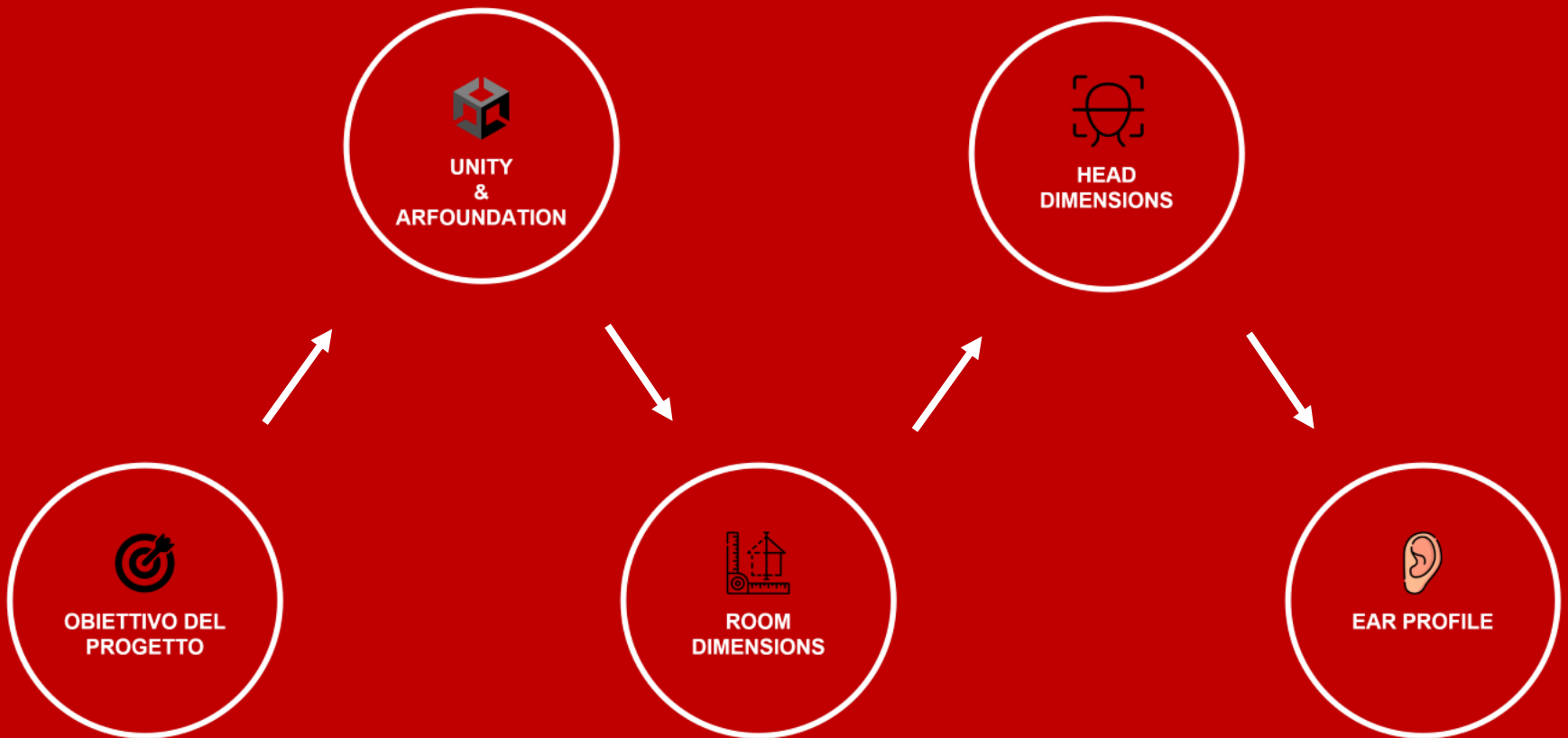
Correlatore: Prof. Alessandro Giuseppe Privitera

Anno Accademico 2022-2023



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

PROGETTO DI TESI TRIENNALE : ROAD MAP





**OBIETTIVO DEL
PROGETTO**

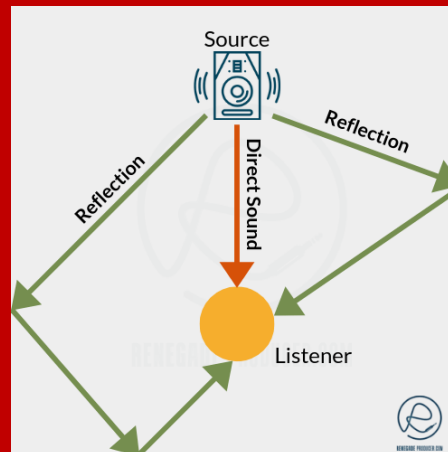


OBIETTIVO DEL PROGETTO

Start Point

SDN library è una libreria per la spazializzazione/riverberazione del suono, funzionante con Unity. In particolare si occupa di ricreare un ambiente acustico virtuale simile a quello reale.

Necessita di alcuni dati di input come, ad esempio, le dimensioni dell'ambiente indoor da ricreare.





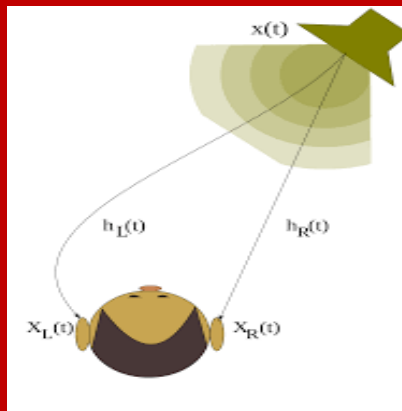
OBIETTIVO DEL PROGETTO

Start Point

HRTF: Head-Related Transfer Function:

Risposta che caratterizza il modo in cui un orecchio percepisce un suono da un punto nello spazio.

Questa risposta è influenzata dalle dimensioni della testa, dalle orecchie e dal canale uditivo.

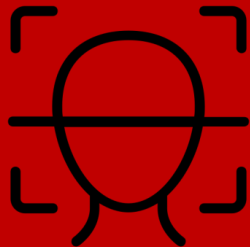




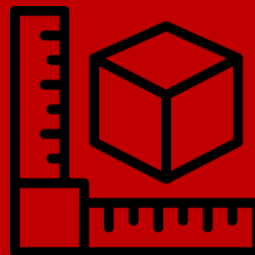
OBIETTIVO DEL PROGETTO

Sviluppo di un'applicazione contenente le seguenti feature:

Head Dimensions



Room Dimensions



Ear Profile





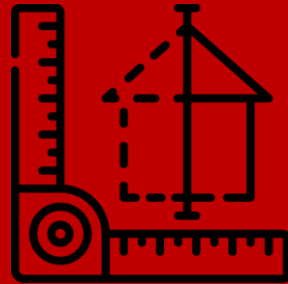
**UNITY
&
ARFOUNDATION**



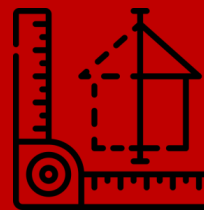
UNITY & ARFOUNDATION

Unity è un game engine
multiplatforma che
consente lo sviluppo di
videogiochi e altri
contenuti interattivi
(wikipedia)

AR Foundation è un
framework
multiplatforma che
consente di realizzare
esperienze di realtà
aumentata.



ROOM DIMENSIONS



ROOM DIMENSIONS

La scena è composta da 4 fasi principali :



Posizionamento
dell'origine



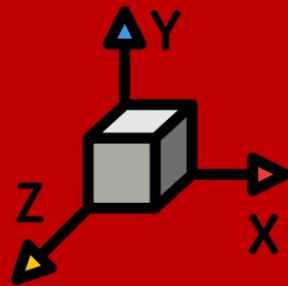
Scansione della
stanza



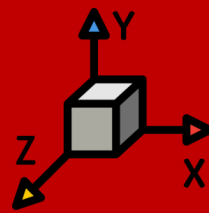
Calcolo delle
distanze



Salvataggio e
scrittura su file

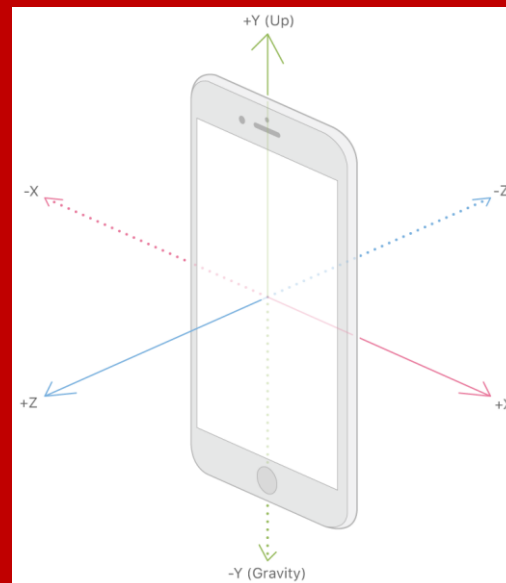


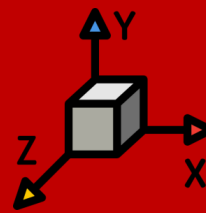
**Posizionamento
dell'origine**



Posizionamento dell'origine

Appena la sessione AR viene avviata, il dispositivo utilizzato riporterà come sua posizione il punto $(0,0,0)$. Le caratteristiche rilevate, come i piani, saranno relative a questa posizione iniziale del device.

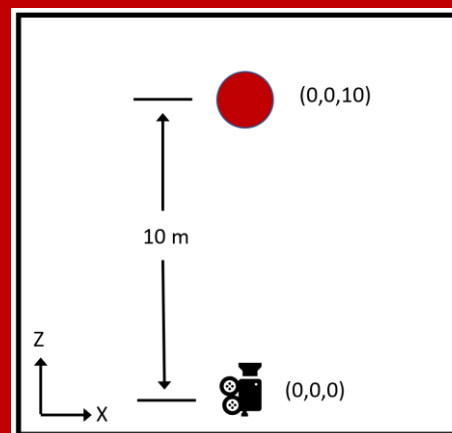


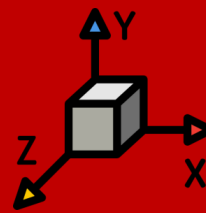


Posizionamento dell'origine

L'idea è quindi quella di far riposizionare l'origine della scena all'utente in modo tale da poter sfruttare:

- La sola componente z dei piani per il calcolo della lunghezza
- La sola componente x dei piani per il calcolo della larghezza
- La sola componente y dei piani per il calcolo dell'altezza



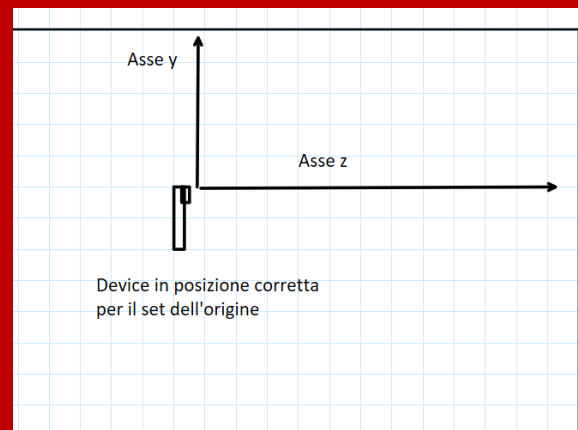


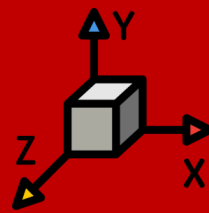
Posizionamento dell'origine

All'utente viene quindi chiesto premere il pulsante per il riposizionamento una volta che il device è in posizione verticale e con la camera che punta verso il primo piano da rilevare.

In questo modo si ottiene:

- Asse z perpendicolare alla prima coppia di piani verticali
- Asse x perpendicolare alla seconda coppia di piani verticali
- Asse y perpendicolare a soffitto e pavimento.





Posizionamento dell'origine

Come avviene il riposizionamento?

```
MakeContentAppearAt(m_ARSessionOrigin.transform,  
m_ARCameraManager.transform.position, m_ARCameraManager.transform.rotation);
```

```
public void MakeContentAppearAt(Transform content, Vector3 position, Quaternion rotation)
```

Parameters

Type	Name	Description
Transform	<i>content</i>	The <code>Transform</code> of the content you wish to affect.
Vector3	<i>position</i>	The position you wish the content to appear at. This could be a position on a detected plane, for example.
Quaternion	<i>rotation</i>	The rotation the content should appear to be in, relative to the <code>Camera</code> .



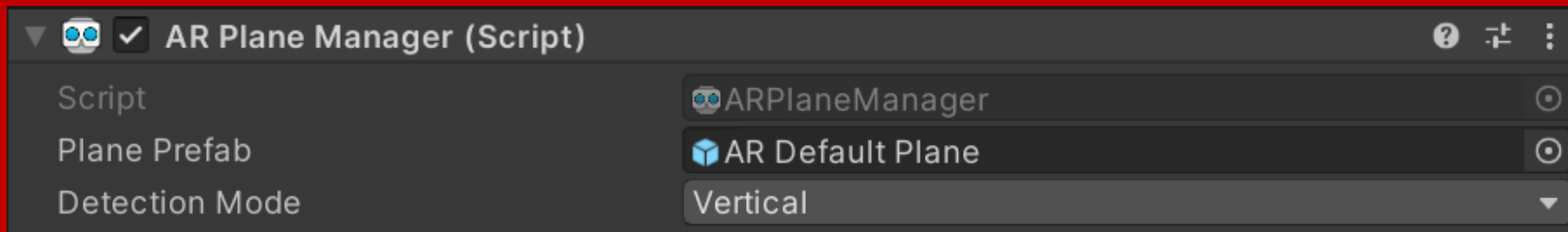
**Scansione della
stanza**



Scansione della stanza

Come avviene la scansione?

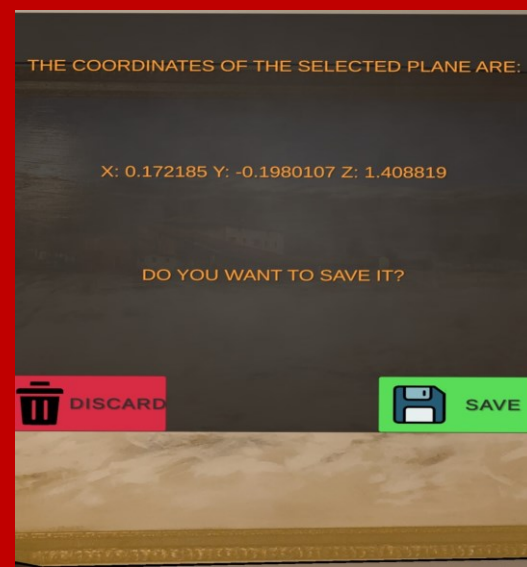
Tramite la feature «PlaneDetection» messa a disposizione da ARFoundation vengono rilevati i principali piani (verticali e orizzontali) che costituiscono la stanza.





Scansione della stanza

Sta all'utente scegliere uno tra i vari piani virtuali rilevati. Non ha importanza che il piano virtuale rappresenti interamente quello reale, basta solo che una piccola parte venga rilevata correttamente.





Scansione della stanza

Come viene rilevata la selezione dell'utente?

La selezione del piano da parte dell'utente viene rilevata tramite Ray casting (hit testing).

```
Raycast(touchPosition, hits, UnityEngine.XR.ARSubsystems.TrackableType.PlaneWithinPolygon)
```

In particolare il metodo sopracitato restituisce «true» se l'hit testing fatto partire dalla posizione «touchPosition» va ad intersecare un piano virtuale rilevato dall'applicazione («PlaneWithinPolygon»).

```
TrackableId selectedPlaneID = hits[0].trackableId;  
currentSelectedPlane = m_ARPlaneManager.GetPlane(selectedPlaneID);
```



Scansione della stanza

ANCHOR POINTS

«Posa nell'ambiente fisico che viene tracciata da un dispositivo XR. Mano a mano che il dispositivo raffina la comprensione dell'ambiente, gli anchors vengono aggiornati, aiutando a mantenere i contenuti virtuali connessi ad una posizione ed orientazione del mondo reale»
(Documentazione Unity)

```
ARAnchor anchor = m_ARAnchorManager.AttachAnchor(currentSelectedPlane, new Pose  
(currentSelectedPlane.transform.position, currentSelectedPlane.transform.rotation));
```



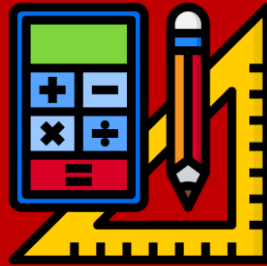


Scansione della stanza

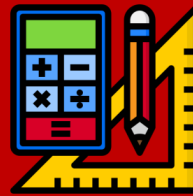
La procedura viene ripetuta e i piani selezionati dall'utente vengono salvati in una lista.

Una volta acquisiti i 4 piani verticali, si passa alla scansione di pavimento e soffitto, cambiando la modalità di detection mode.

```
m_ARPlaneManager.requestedDetectionMode = PlaneDetectionMode.Horizontal;
```

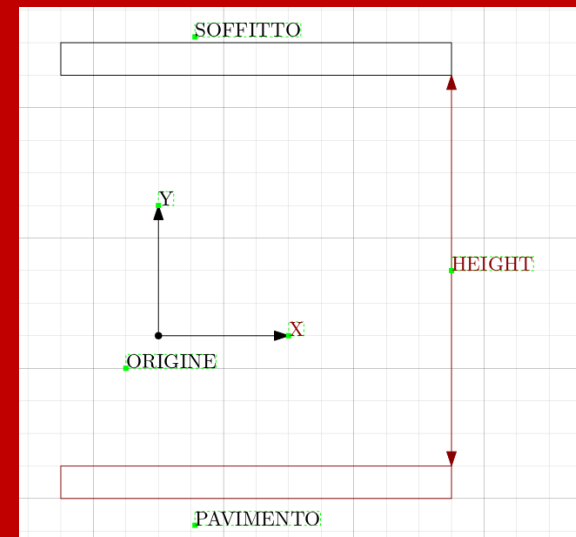


Calcolo delle distanze



Calcolo delle distanze

Accedendo alle coordinate di ogni piano e, come accennato in precedenza, sfruttando la componente lungo l'asse ortogonale al piano stesso, si calcolano le distanze tra coppie di piani opposti.





Calcolo delle distanze

Esempio di calcolo della lunghezza

```
struct RoomDimensions  
{  
    public float length;  
    public float width;  
    public float height;  
}
```

```
dimensions.lenght = Mathf.Abs(walls[0].transform.position.z)  
+ Mathf.Abs(walls[1].transform.position.z);
```




**Salvataggio e
scrittura su file**



Salvataggio e scrittura su file

Una volta calcolate le dimensioni, l'applicazione genera un file .csv sul quale vengono stampate le dimensioni e le coordinate di ogni piano.

Questo file quindi può essere successivamente utilizzato come input per applicazioni che utilizzano la SDN library.

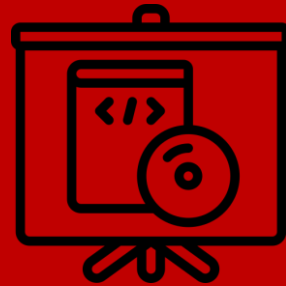




Salvataggio e scrittura su file

Esempio di output generato

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Wall1(x)	Wall1(y)	Wall1(z)	Wall2(x)	Wall2(y)	Wall2(z)	Wall3(x)	Wall3(y)	Wall4(z)	Wall4(x)	Wall4(y)	Wall4(z)	Floor(x)	Floor(y)	Floor(z)	Ceiling(x)	Ceiling(y)	Ceiling(z)	RoomLengh	RoomWid	RoomHeight
2	-0,75926	-0,06786	1,43652	0,00368	-0,13516	-3,01191	-1,27071	-0,88543	-1,3235	3,47206	0,05713	-1,31001	0,44675	-1,70475	-0,28074	0,25933	1,31081	-0,54006	4,448429	4,74277	3,015553
3	-0,82261	-0,15329	1,62651	-0,0921	-0,13191	-2,84597	-1,23667	-0,94387	-1,22246	3,40449	0,09998	-1,2466	0,55062	-1,75432	-0,31396	0,18001	1,24143	-0,47012	4,472484	4,64117	2,995749
4	-0,40764	-0,06559	1,53667	0,24722	-0,23194	-2,82747	-1,10329	-0,97394	-1,1982	3,55934	0,0228	-1,28347	0,46064	-1,61508	-0,39919	0,51489	1,21923	-0,52654	4,364141	4,66263	2,834316
5	-0,84398	-0,11633	1,4846	-0,03533	-0,17831	-2,96303	-1,32262	-0,99227	-1,3356	3,29401	-0,0095	-1,39647	0,21729	-1,62937	-0,51334	0,21253	1,32544	-0,6392	4,447629	4,61663	2,954802
6	-0,64654	-0,27045	1,57985	-0,01209	-0,06643	-2,90499	-1,13258	-0,91004	-1,26943	3,67855	0,0893	-1,11634	0,65923	-1,61427	-0,41143	0,34766	1,33176	-0,5056	4,484841	4,81113	2,946032
7	0,08821	-0,16261	1,54394	1,18497	-0,2765	-2,80596	-3,21399	0,02894	-0,32906	1,31471	-0,94686	-0,08702	-0,72216	-1,55614	-1,38786	0,01564	1,33841	-0,68401	4,349895	4,5287	2,894557
8	0,02851	-0,06823	1,493	1,05402	-0,34773	-2,92902	-3,33312	0,08335	-0,23871	1,39704	-0,9723	-0,13148	-0,40832	-1,58265	-0,95076	0,04584	1,29283	-0,87129	4,422015	4,73016	2,875483
9	0,04511	-0,11181	1,44468	1,01786	-0,35907	-2,99585	-3,23411	-0,03242	-0,28055	1,52749	-1,00628	-0,25795	-0,30771	-1,70576	-1,2026	0,17951	1,31814	-0,44244	4,440526	4,7616	3,023896
10	-0,76964	-0,1883	1,58338	0,1395	-0,18901	-2,83292	-1,14139	-0,96822	-1,19824	3,59802	-0,2495	-0,98853	0,59488	-1,66097	-0,43503	0,37259	1,31401	-0,28616	4,416303	4,73941	2,974979
11	-0,04102	-0,23825	1,37517	0,95698	-0,14005	-2,94621	1,42438	-0,92779	-0,158	-3,32883	-0,15715	-0,47066	-0,67127	-1,6437	-1,77083	-0,08934	1,17207	-0,69259	4,321381	4,7532	2,815769



Demo



Demo

MULTIMODAL MODELING OF INDOOR ENVIRONMENTS AND HUMAN HEAD

00:00



ROOM DIMENSIONS



HEAD DIMENSIONS



EAR PROFILE



QUIT

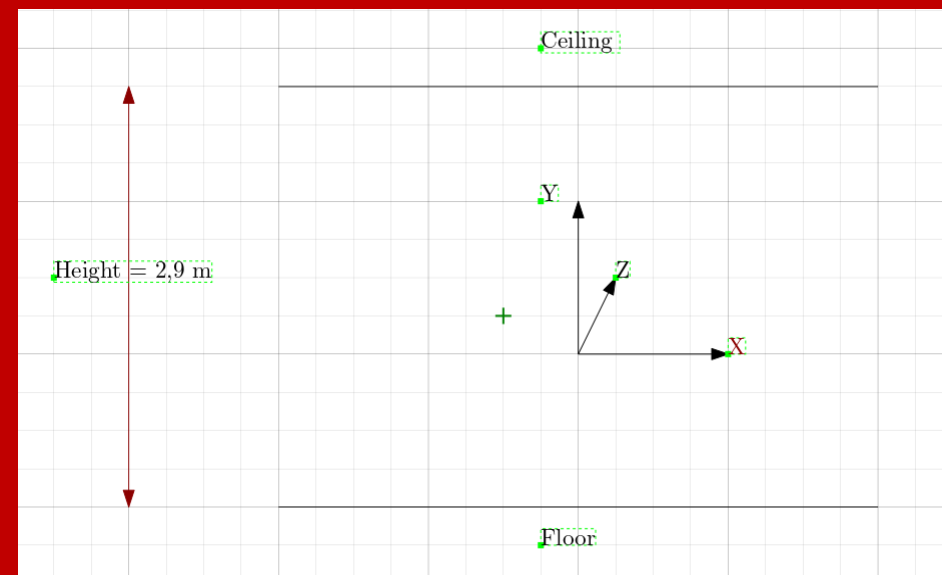
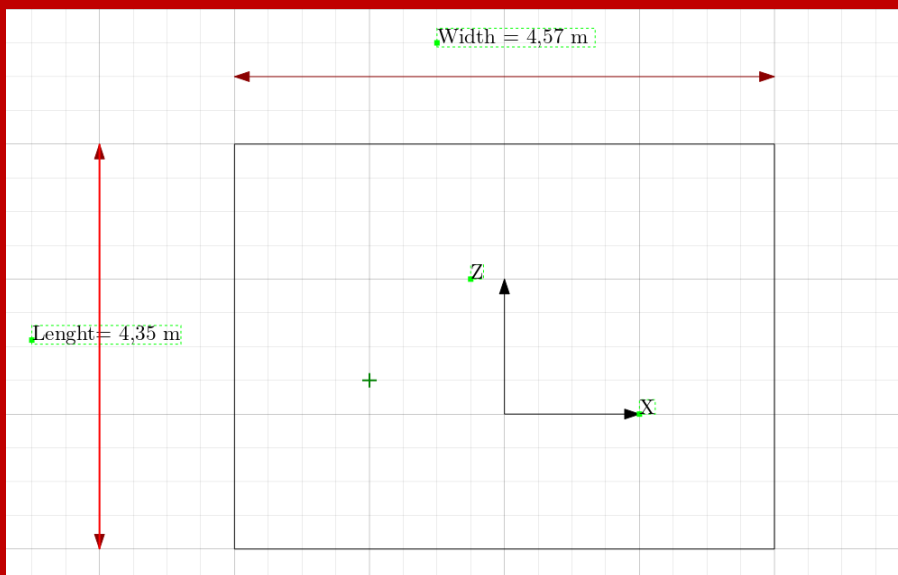


Validazione



Validazione

Per la validazione dei dati, l'applicazione è stata testata in una stanza con le seguenti caratteristiche:



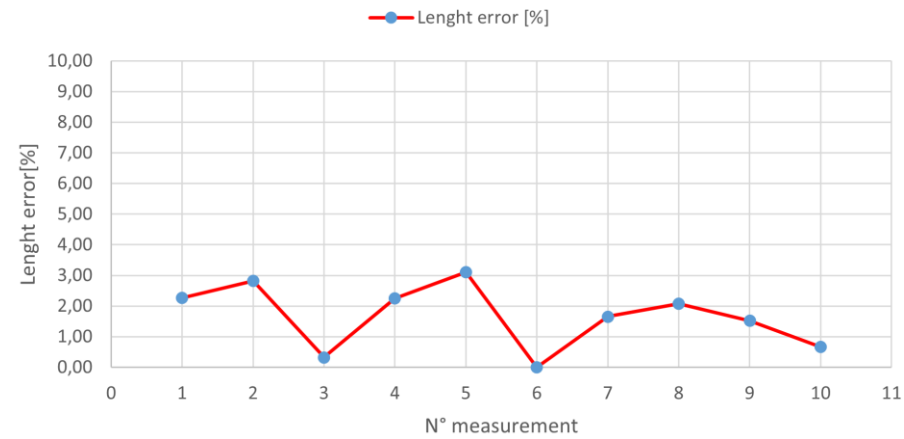
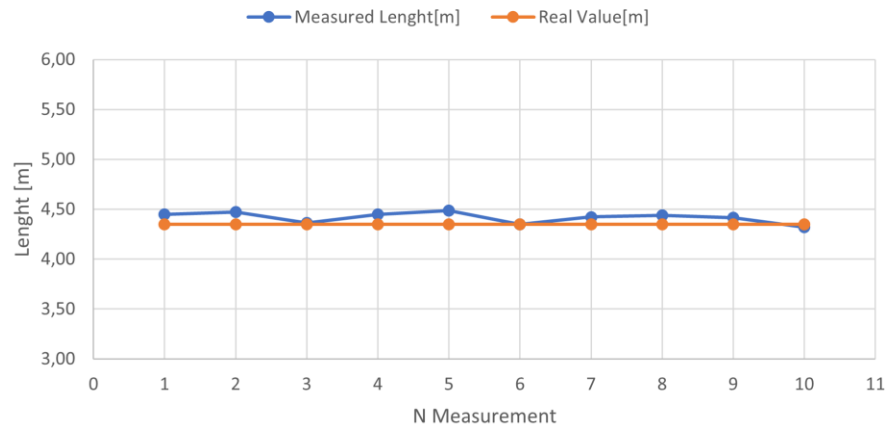


Validazione

N° misura	RoomLength [m]	RoomWidth [m]	RoomHeight [m]	Length error [%]	Width error [%]	Height error [%]
1	4,4484	4,7428	3,0156	2,2627	3,7805	3,9846
2	4,4725	4,6412	2,9957	2,8157	1,5572	3,3017
3	4,3641	4,6626	2,8343	0,3251	2,0269	2,2650
4	4,4476	4,6166	2,9548	2,2443	1,0204	1,8897
5	4,4848	4,8111	2,9460	3,0998	5,2764	1,5873
6	4,3499	4,5287	2,8946	0,0024	0,9036	0,1877
7	4,4220	4,7302	2,8755	1,6555	3,5046	0,8454
8	4,4405	4,7616	3,0239	2,0811	4,1926	4,2723
9	4,4163	4,7394	2,9750	1,5242	3,7071	2,5855
10	4,3214	4,7532	2,8158	0,6579	4,0088	2,9045
ELABORAZIONE DATI						
media [m]	4,4168	4,6987	2,9331			
σ [m]	0,0544	0,0846	0,0743			
Valore reale [m]	4,3500	4,5700	2,9000			
Errore medio [%]	1,6669	2,9978	2,3824			

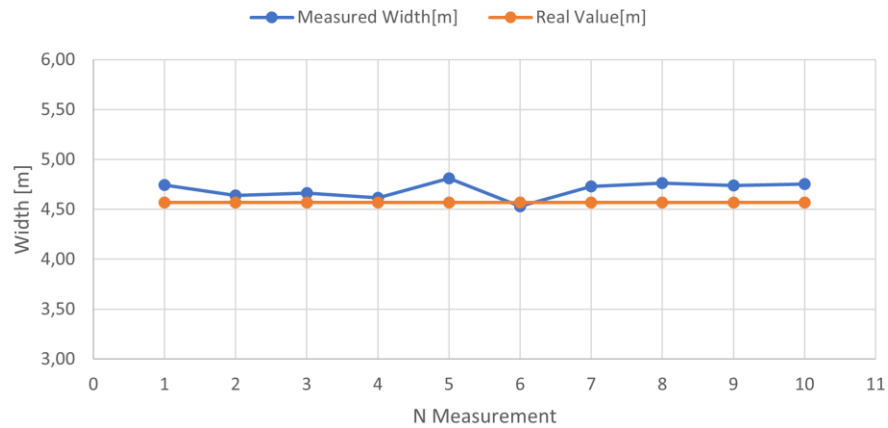


Validazione



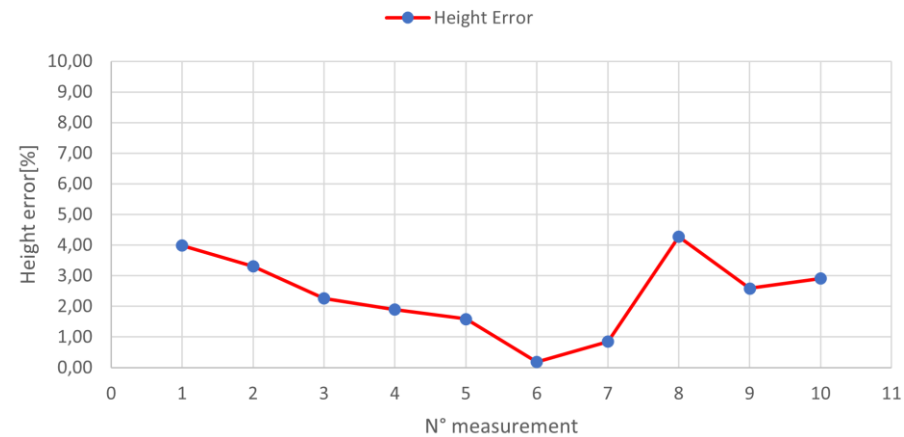
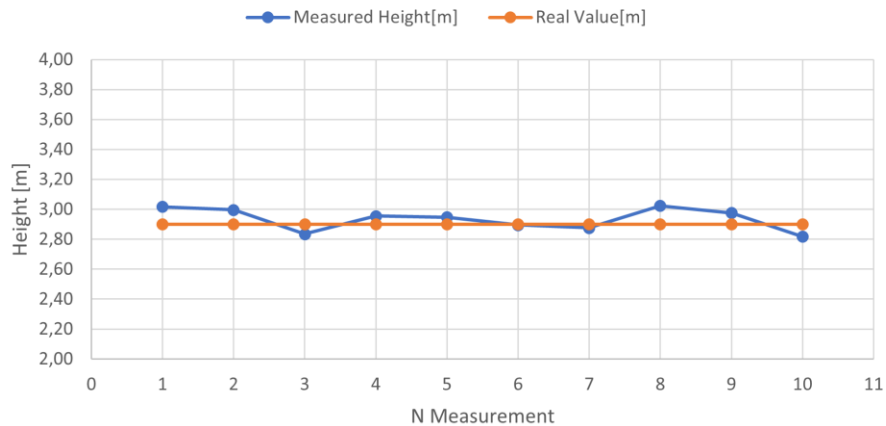


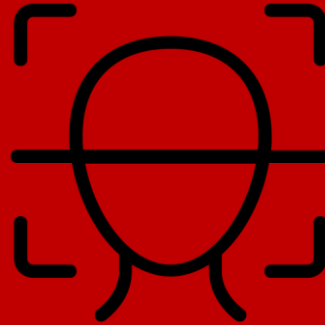
Validazione



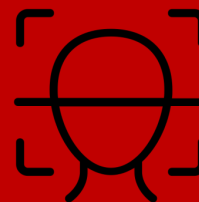


Validazione





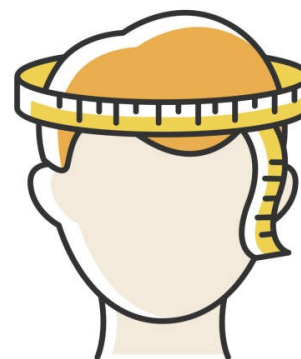
**HEAD
DIMENSIONS**



HEAD DIMENSIONS

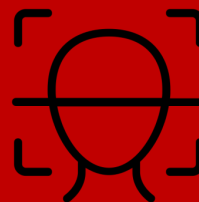
Obiettivo della feature:

STIMARE DIMENSIONI VOLTO UMANO
RILEVATO TRAMITE SMARTPHONE



DISTANZA ORECCHIO DX-
ORECCHIO SX

DISTANZA NUCA-NASO



HEAD DIMENSIONS

Fasi di sviluppo:

STUDIARE SCENA
ARFOUNDATION SAMPLES

COMPRENDERE LOGICA
UNITY E C#

GESTIRE L'ALGORITMO



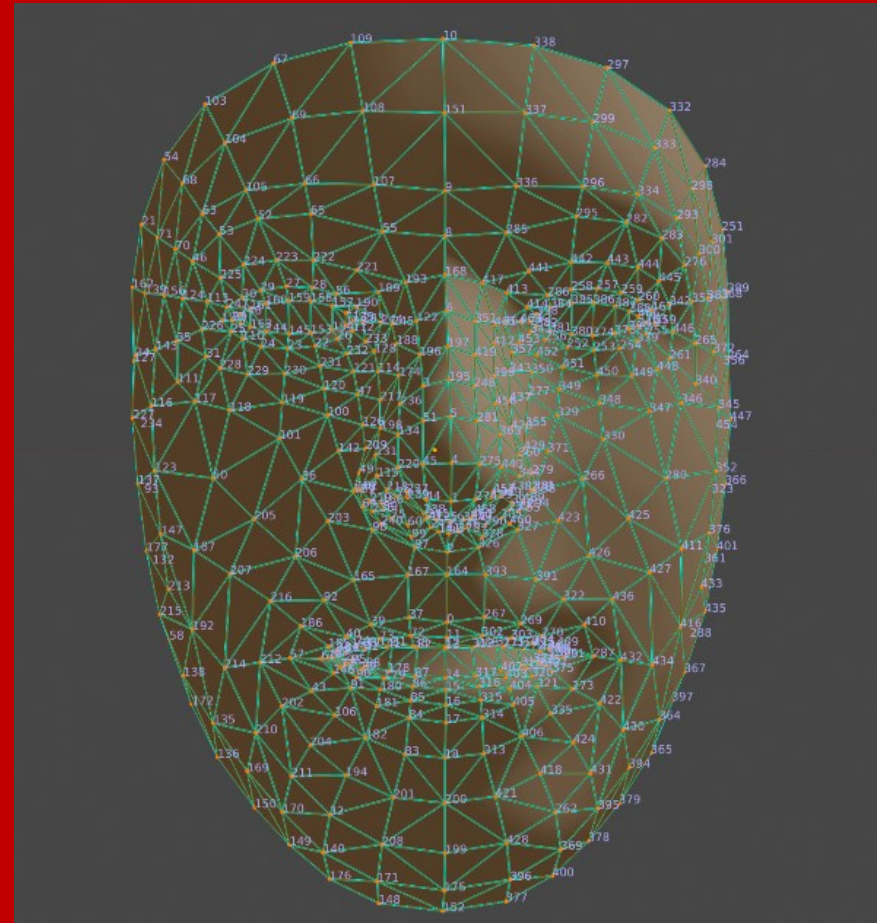
HEAD DIMENSIONS



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Idea di progetto:

- Il tracciamento del volto avviene grazie alla sovrapposizione sul viso di una maschera (face-prefab) costituita da 468 punti (vertici). Pertanto come primo passo bisognerà prendere la faccia rilevata e successivamente ricavare da questa i vertici.
- L'idea per il calcolo delle distanze si basa quindi sulla possibilità di accedere ai vertici della maschera sovrapposta al viso.

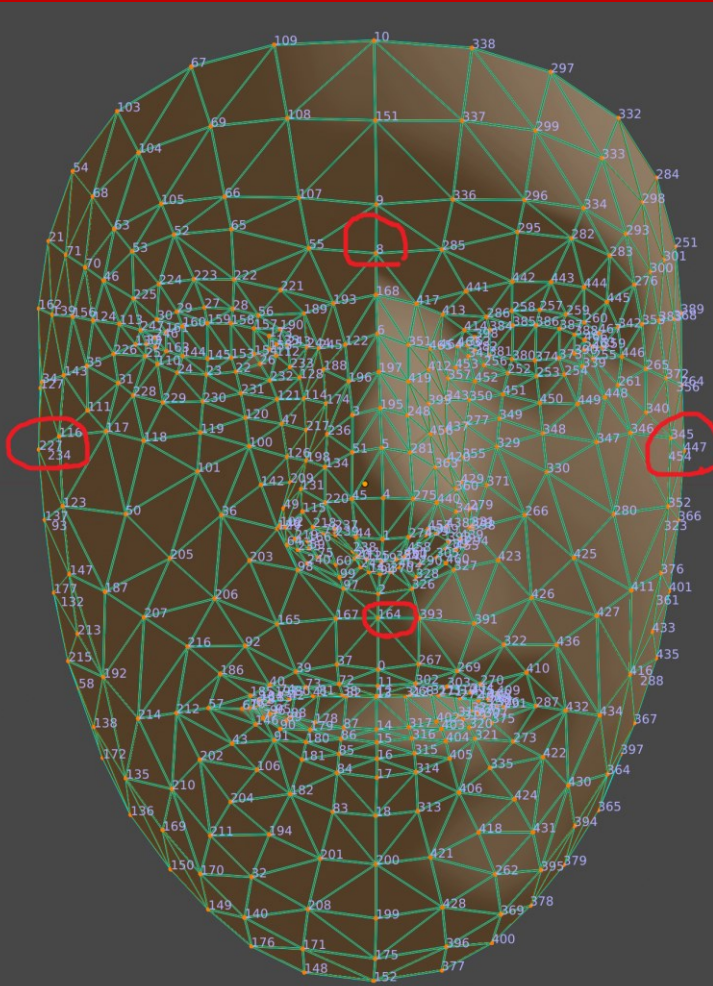


HEAD DIMENSIONS



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Idea di progetto:



```
foreach (ARFace f in m_arFaceManager.trackables)
{
    m_face = f;

    up = m_face.vertices[8];
    center = m_face.vertices[164];

    left1 = m_face.vertices[345];
    left2 = m_face.vertices[447];
    left3 = m_face.vertices[454];

    right1 = m_face.vertices[116];
    right2 = m_face.vertices[227];
    right3 = m_face.vertices[234];
}
```


HEAD DIMENSIONS



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Idea di progetto:

Per rendere le misure più precise...

```
left = AverageV3(left1, left2, left3);  
right = AverageV3(right1, right2, right3);
```

```
//faccio una media dei 3 vertici di sx e dx  
private Vector3 AverageV3(Vector3 a, Vector3 b, Vector3 c)  
{  
    return (a + b + c) / 3;  
}
```

HEAD DIMENSIONS



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Calcolo delle distanze:

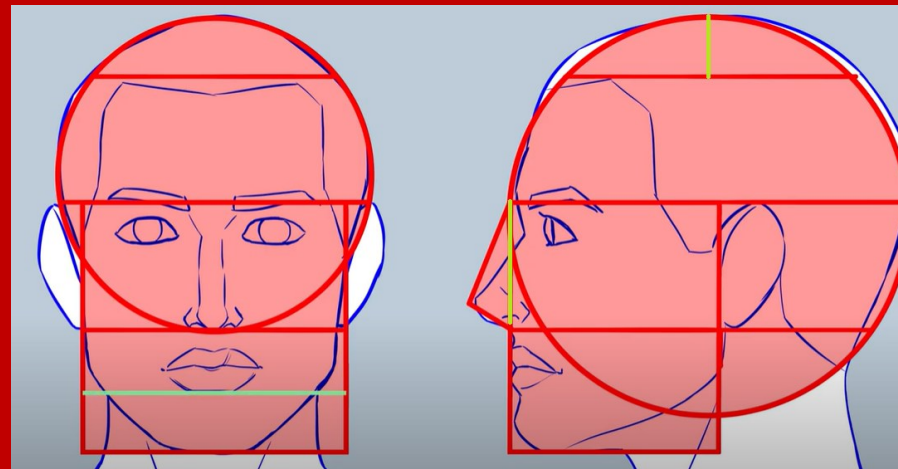
DISTANZA ORECCHIE:

calcolare la distanza tra due punti (coordinate)

PROFONDITA':

proporzioni di una testa umana (geometria-sferoide).

```
float dist = Vector3.Distance(other.position, transform.position);
```



HEAD DIMENSIONS



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Demo:



HEAD DIMENSIONS



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Validazione:

orecchie 1 (cm)	profondità 1 (cm)	reali orecchie 1 (cm)	reali profondità 1 (cm)	errore orecchie 1 (cm)	errore profondità 1 (cm)
13,97	20,57	15	19,5	1,03	1,07
14,185	20,79	15	19,5	0,815	1,29
15,1	18,51	15	19,5	0,1	0,99
14,91	17,69	15	19,5	0,09	1,81
14,35	19,8	15	19,5	0,65	0,3
14,77	21,01	15	19,5	0,23	1,51
13,77	20,86	15	19,5	1,23	1,36
13,98	19,3	15	19,5	1,02	0,2
14,27	19,42	15	19,5	0,73	0,08
14,11	20,18	15	19,5	0,89	0,68
14,3415	19,813			0,6785	0,929
orecchie 2 (cm)	profondità 2 (cm)	reali orecchie 2 (cm)	reali profondità 2 (cm)	errore orecchie 2 (cm)	errore profondità 2 (cm)
13,66	17,45	14,5	19	0,84	1,55
14	20,12	14,5	19	0,5	1,12
14,58	19,5	14,5	19	0,08	0,5
14,18	18,69	14,5	19	0,32	0,31
13,79	19,14	14,5	19	0,71	0,14
13,93	19,43	14,5	19	0,57	0,43
13,81	18,86	14,5	19	0,69	0,14
15,62	20,47	14,5	19	1,12	1,47
14,06	18,6	14,5	19	0,44	0,4
13,93	19,77	14,5	19	0,57	0,77
14,156	19,203			0,584	0,683

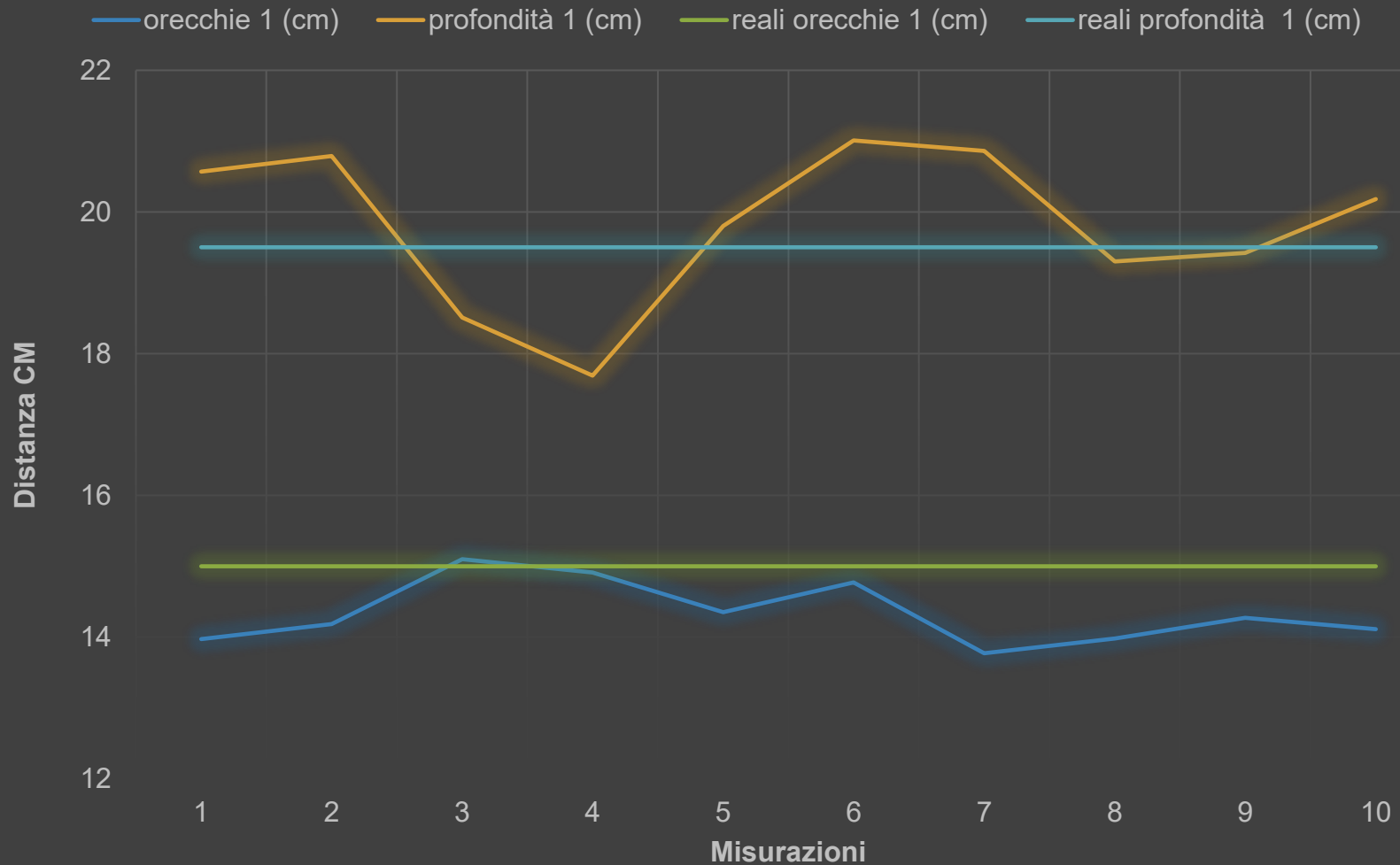
HEAD DIMENSIONS



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Validazione grafica:

Volto #1

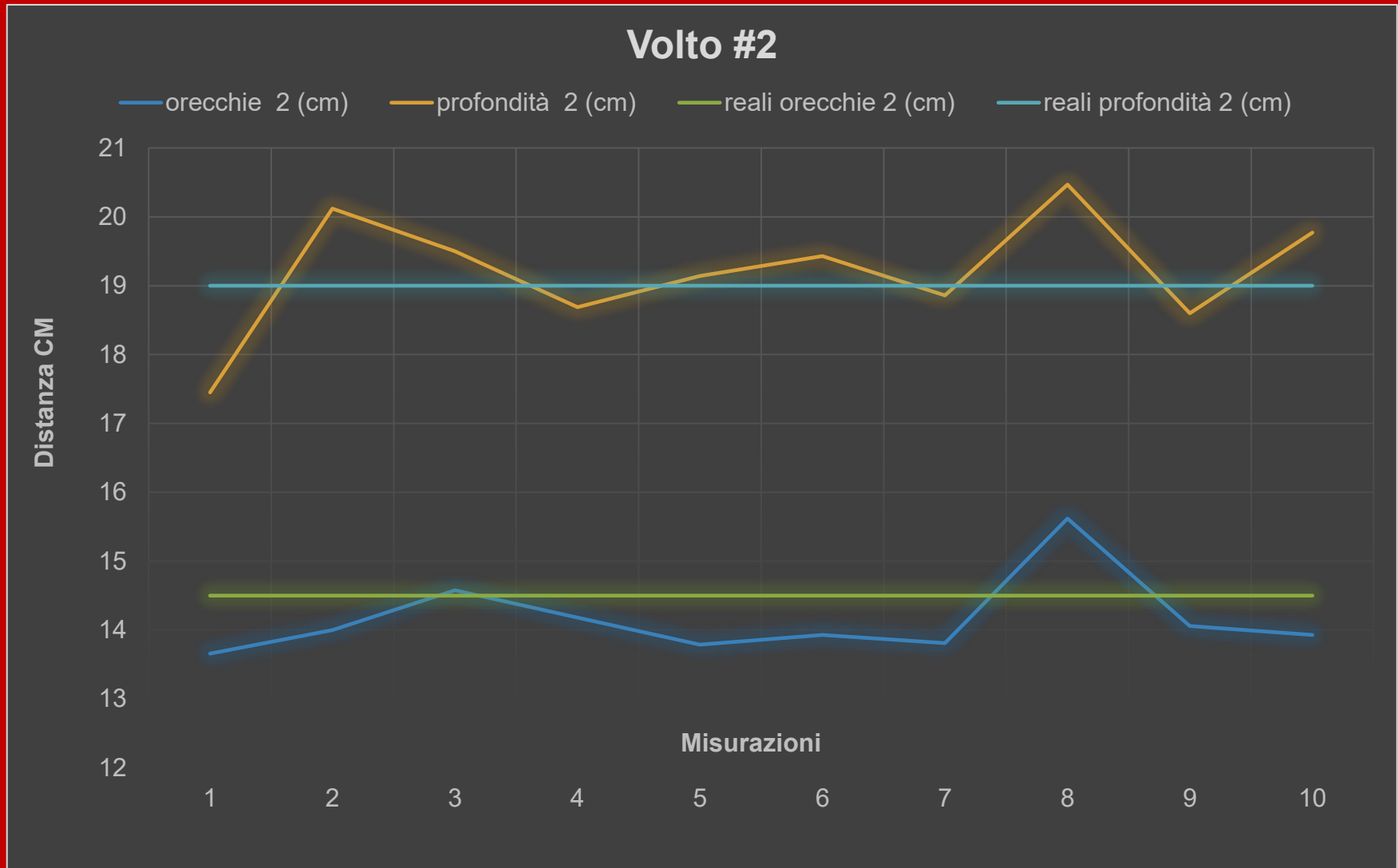


HEAD DIMENSIONS



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Validazione grafica:

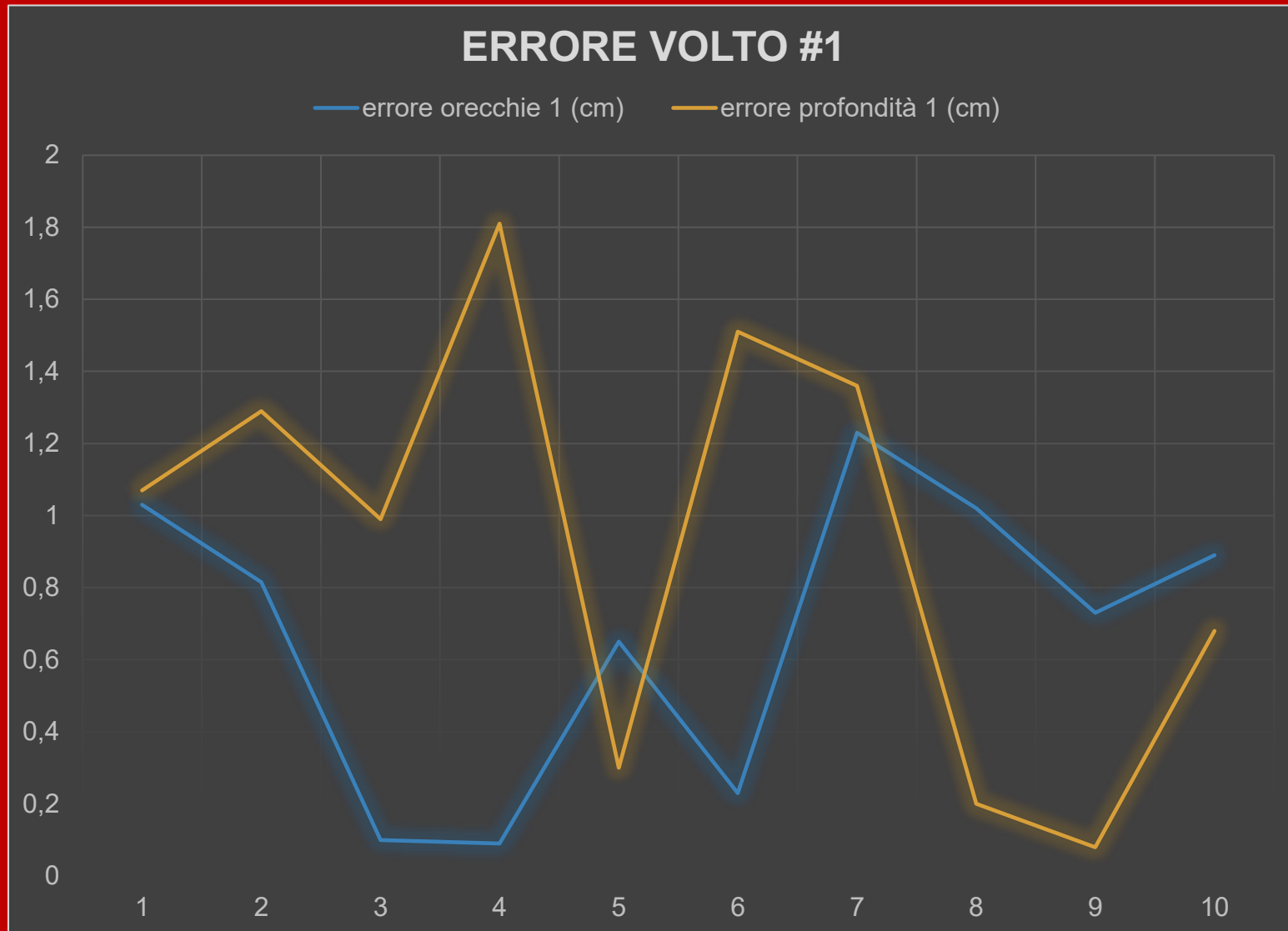


HEAD DIMENSIONS



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Validazione grafica:

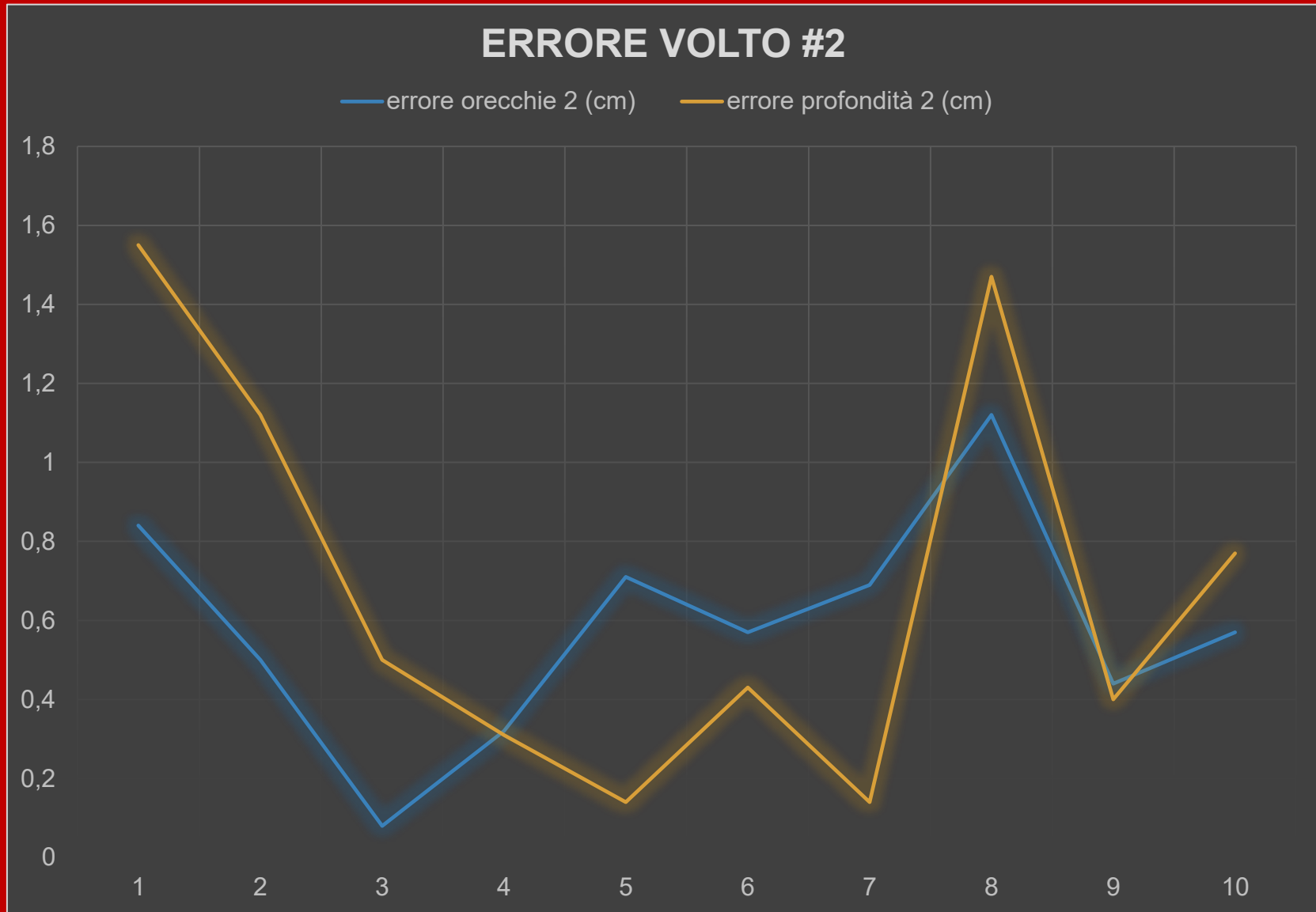


HEAD DIMENSIONS



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Validazione grafica:





EAR PROFILE



EAR PROFILE

La scena ha come scopo quello di ricavare le coordinate (in pixels) relative al profilo dell'orecchio



Acquisizione
immagine



Applicazione
distorzione



Controllo tocchi
utente



**Acquisizione
immagine**



Acquisizione immagine

Per questa feature sono stati utilizzati i plugin «Unity Native Camera» e «Unity Native Gallery», funzionanti con Unity e disponibili su github ai seguenti link:

- <https://github.com/yasirkula/UnityNativeCamera>
- <https://github.com/yasirkula/UnityNativeGallery>





Acquisizione immagine

In particolare

```
NativeGallery.getImageFromGallery( ( path ) );
```

consente l'accesso alla galleria del dispositivo, e

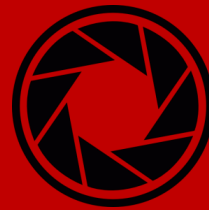
```
NativeCamera.takePicture( ( path ) );
```

permette all'utente di utilizzare la camera del device per scattare una foto.





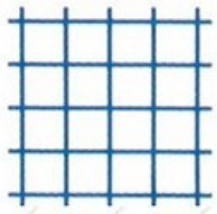
**Applicazione
distorsione**



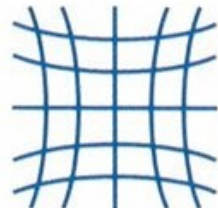
Applicazione distorsione

La distorsione della lente è un effetto che si verifica quando si utilizza una lente per catturare immagini o video.

In unity, è possibile utilizzare un componente di effetto di post-elaborazione per aggiungere la distorsione della lente alla fotocamera virtuale.



VISIONE NORMALE

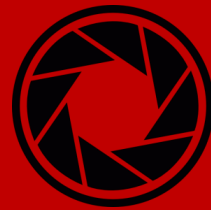


DISTORSIONE A CUSCINO



DISTORSIONE A BARILE





Applicazione distorsione

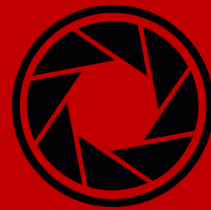
L'obiettivo è quello di modificare l'immagine importata che è soggetta a distorsione, dovuta dalla lente del device, in modo da visualizzarla correttamente (togliere la distorsione).

Properties

▼ Lens Distortion

All None

- Intensity
- X Multiplier
- Y Multiplier
- Center X Y
- Scale



Applicazione distorsione

Via script:

```
private LensDistortion m_lensdistortion;
```

```
m_lensdistortion.intensity.value = -0.5f;  
m_lensdistortion.yMultiplier.value = 0.5f;  
m_lensdistortion.xMultiplier.value = 0.2f;
```



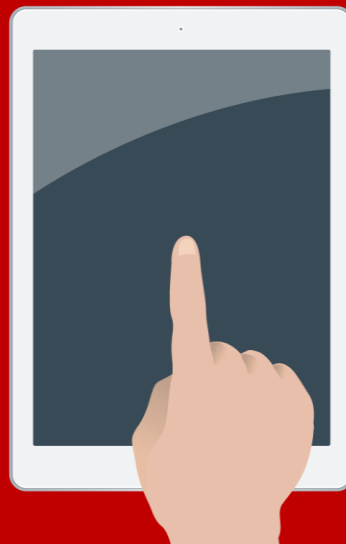
**Controllo tocchi
utente**



Controllo tocchi utente

GOAL : ricavare le coordinate relative al profilo dell'orecchio.

All'utente viene chiesto quindi di trascinare il dito sullo schermo, seguendo il profilo dell'orecchio della foto caricata.





Controllo tocchi utente

Attraverso la classe `Input`, si accede ad ogni tocco sullo schermo rilevato dall'applicazione:

```
currentTouch = Input.GetTouch(0);
```

Ogni posizione (x,y) dello schermo «toccata» viene salvata in una lista

```
if (currentTouch.phase == TouchPhase.Moved){  
    touchPositions.Add(currentTouch.position);  
}
```



Controllo tocchi utente

Quando viene rilevata la fine del tocco tramite il controllo :
`if(currentTouch.phase == TouchPhase.Ended)`

Viene abilitato il pulsante che permette di salvare su file le coordinate salvate nella lista «touchPositions»

```
saveDataIntoFile.gameObject.SetActive(true);
```

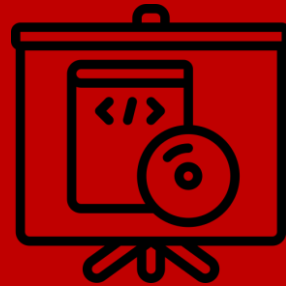


Controllo tocchi utente

Output

L'applicazione genera in maniera automatica un file di output .csv, dove vengono salvate tutte le coordinate (in pixels) relative ad ogni tracciamento del profilo.

	A	B	C
1	num. of touch	x	y
2	1	540,791	991,846
3	1	542,9004	989,949
4	1	545,5371	987,781
5	1	552,3926	984,8
6	1	592,4263	974,273
7	1	637,7694	973,387
8	1	663,5156	978,581
9	1	712,4344	998,829
10	1	757,4055	1030,59
11	1	794,0115	1071,92
12	1	823,058	1123,7
13	1	847,4252	1175,7
14	1	866,4199	1228,35



Demo



Demo

MULTIMODAL MODELING OF INDOOR ENVIRONMENTS AND HUMAN HEAD

00:00



ROOM DIMENSIONS



HEAD DIMENSIONS



EAR PROFILE



QUIT