

# 1. Anchor and Trackable

ARCore definisce gli Anchor per assicurare che gli oggetti virtuali rimangano nella stessa posizione e vengano tracciati nel tempo. L'ambiente circostante può cambiare, ed è necessario che la posizione di questi oggetti rimanga stabile. Gli Anchor sono disposti in insieme di punti o piani rappresentati da oggetti di tipo Trackable.

Gli oggetti Trackable rappresentano la forma geometrica sulla quale verranno definiti gli anchor. Su questi oggetti possono essere invocati 3 metodi:

- *createAnchor(*pose: Pose*)* crea un anchor in una posa che è definita nel Trackable corrente. Il tipo di oggetto Trackable definirà il modo con cui l'anchor verrà disposto e la modalità di aggiornamento della sua posa mentre il modello del mondo varia.
- *getAnchors()* restituisce tutti gli anchor presenti nel dato Trackable.
- *getTrackingState()* restituisce un oggetto TrackingState che rappresenta lo stato del Trackable. Questo stato può essere: PAUSED quando il rilevamento viene perso ma potrebbe riprendere in futuro; STOPPED quando viene fermato e non verrà più ripreso; TRACKING se viene tracciato il determinato Trackable.

La definizione di anchor e lo stato di tracciamento sono stati molto importanti per la definizione delle funzionalità principali della nostra applicazione.

In base alla modalità l'applicazione offre funzionalità differenti:

- *Plane Detection*: ARCore rileva dei piani (Trackable) sui quali è possibile posizionare degli animali virtuali. In particolare, quando l'utente tocca un punto preciso del piano viene definito un anchor sul quale verrà renderizzato il modello 3d dell'animale. (Codice in 1.1 nella pagina seguente)
- *Augmented Images*: in ciascun frame viene controllato se lo stato di un'immagine aumentata è TRACKING; in questo caso l'immagine viene riconosciuta e viene definito un anchor nel suo centro nel quale verrà renderizzato il modello del pianeta corrispondente. (Codice in 1.2 nella pagina successiva)

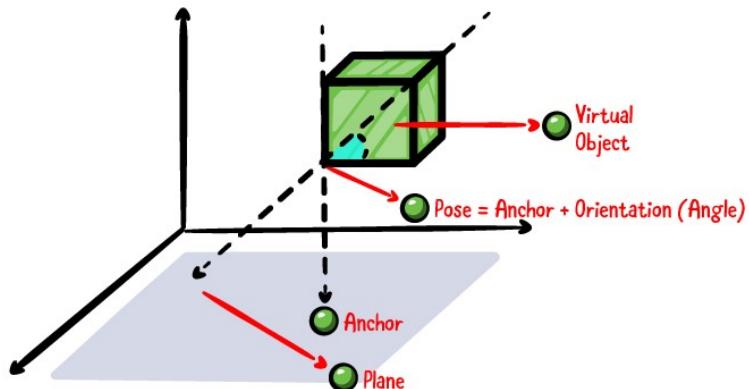
```
1 //Evento che si verifica quando viene toccato un piano
2     arFragment.setOnTapArPlaneListener { hitResult, plane, motionEvent ->
3
4         //Se siamo nella modalità place model
5         if (!switchButton.isChecked) {
6
7             arFragment.arSceneView.scene.addChild(AnchorNode(hitResult.createAnchor()))
8                 .apply {
9
10                 // Crea il transformable model e lo aggiunge all'anchor
11                 addChild(TransformableNode(arFragment.transformationSystem).apply {
12
13                     setModel()
14                     renderable = objRenderable
15
16                     // ...
17
18                 })
19             }
20         }
21     }
22 }
```

Listing 1.1: Definizione Anchor in Plane Detection

```
1 // Per ogni immagine tracciata se non è presente il modello allora viene immediatamente costruito e instanziato
2     for (AugmentedImage in augmentedImages) {
3
4         if (augmentedImage.trackingState == TrackingState.TRACKING) {
5
6             for (i in 0 until namesobj.size) {
7
8                 if (augmentedImage.name.contains(namesobj[i]) && !renderobj[i]) {
9
10                     Toast.makeText(this, ""+namesobj[i]+ " rilevato", Toast.LENGTH_SHORT).show()
11
12                     if (namesobj[i]== "systemsolar"){
13                         renderObject(
14                             arFragment,
15                             augmentedImage.createAnchor(augmentedImage.centerPose),
16                             "solar_system"
17                         )
18                     } else {
19                         renderObject(
20                             arFragment,
21                             augmentedImage.createAnchor(augmentedImage.centerPose),
22                             namesobj[i]
23                         )
24                     }
25                     renderobj[i] = true
26                 }
27             }
28         }
29     }
```

Listing 1.2: Definizione Anchor in Augmented Images

Di seguito è riportata una rappresentazione di come un oggetto virtuale viene disposto in un piano.

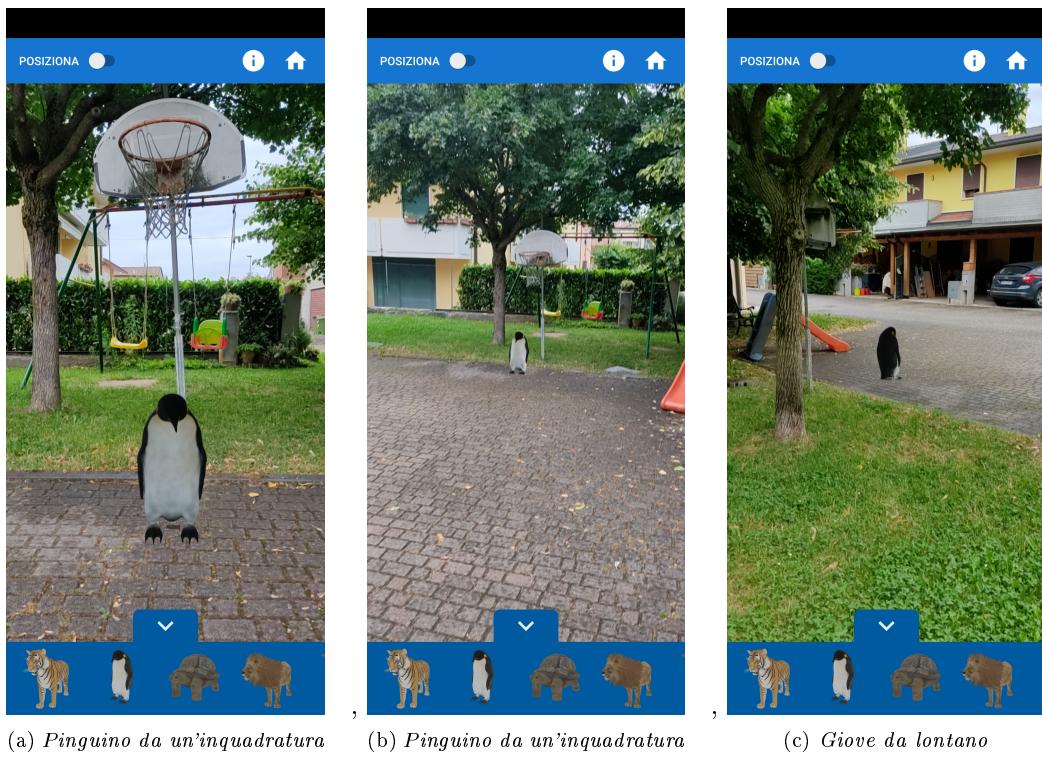


Fonte: <https://medium.com/@jaaveeth.developer/arcore-81528569eb2c>

Figura 1.1: Oggetto virtuale in un piano

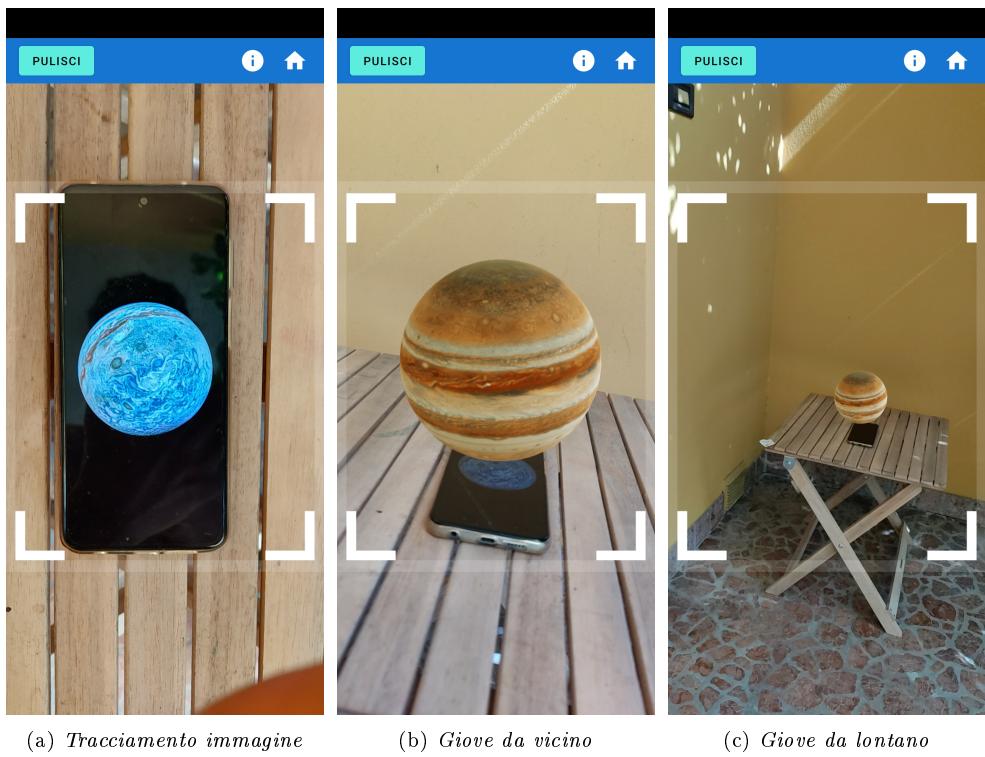
Nella modalità *Plane Detection* la posa (posizione e orientamento) di un animale rimane invariata anche se l'ambiente circostante cambia.

Qui sotto sono riportati degli esempi in cui si può notare che il pinguino rimane nello stesso punto da qualsiasi prospettiva e distanza.



Fonte: <https://developers.google.com/>

Figura 1.2: Esempio di inquadrature differenti in Plane Detection



(a) *Tracciamento immagine*

(b) *Giove da vicino*

(c) *Giove da lontano*

Fonte: <https://developers.google.com/>

Figura 1.3: Esempio di rilevamento in Augmented Images