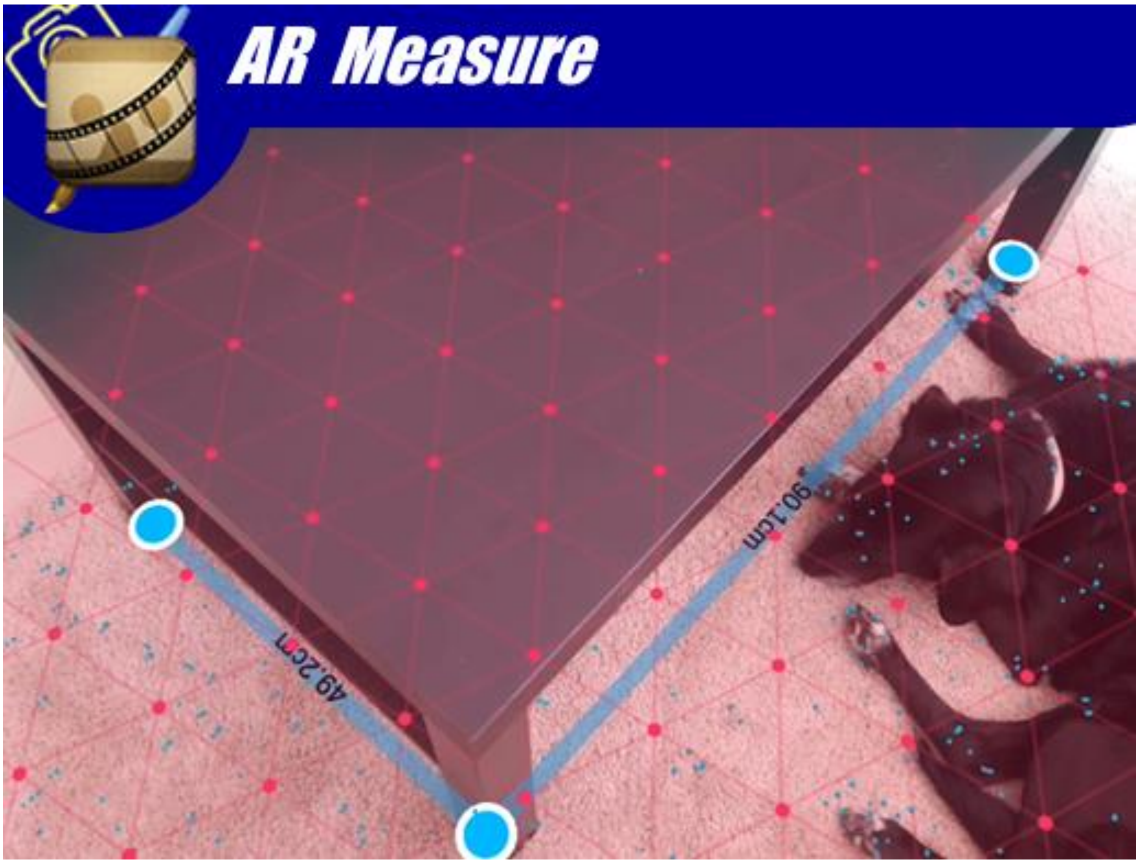


AUTHOR	Cardboard Buddies
CONTACT	michael.soler.beatty@gmail.com
Unity Ver.	2018.3.f1

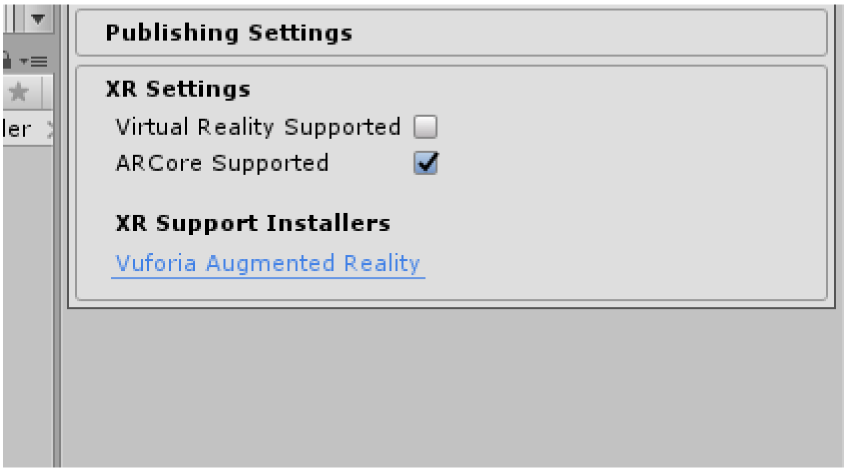
Index

1.Dependencies

This package needs one main package that is called ARcore:

https://github.com/google-ar/arcore-unity-sdk/releases

Remember to enable ARcore in the XR settings.



Publishing Settings

XR Settings

Virtual Reality Supported □

ARCore Supported ☑

XR Support Installers

Vuforia Augmented Reality

2.Description of the package.

This package allows to measure different distances between points in space using ARCore. The markers, lines, text and others are prefabs that can be changed easily. Follow the steps to configure and export the package:

1. Import ARCore sdk for unity.

2. Set the XR support to ARCore in player settings

3. Export to Android.

4. Search for planes

5. Touch the screen to instantiate a point.

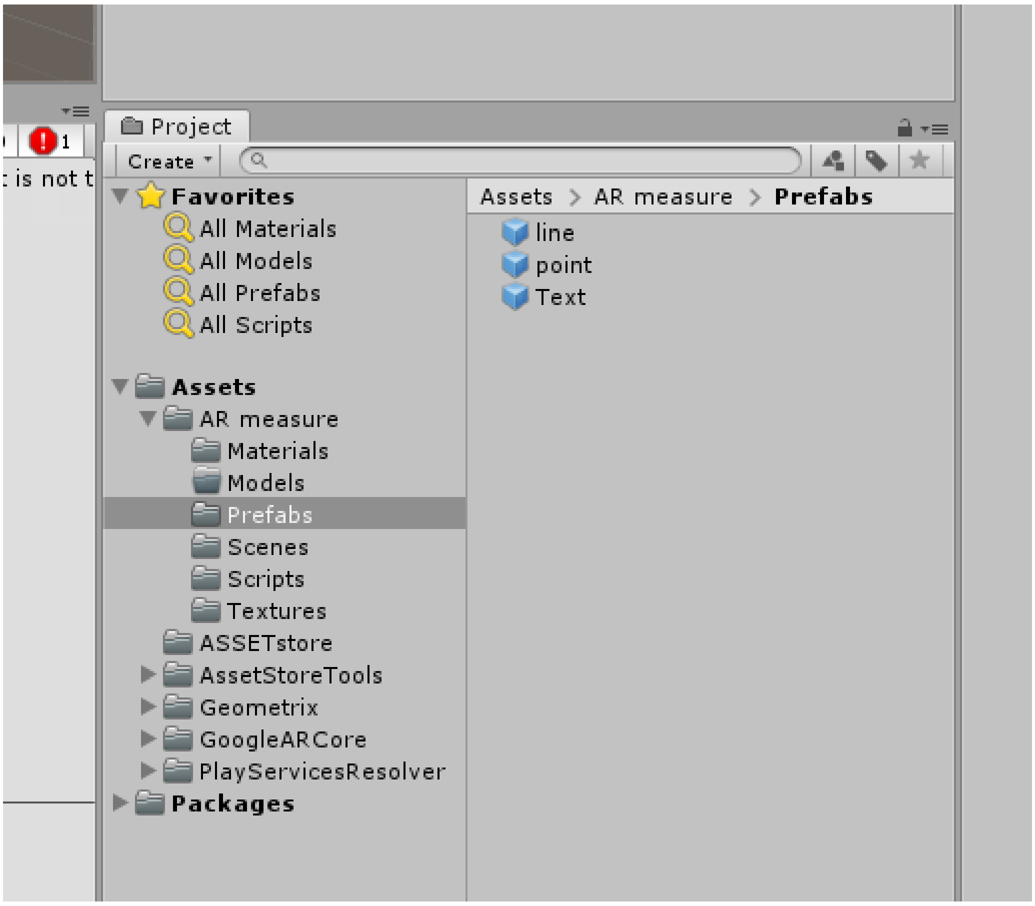
6. Measure distances between two consecutive points.

3.Tags and other considerations

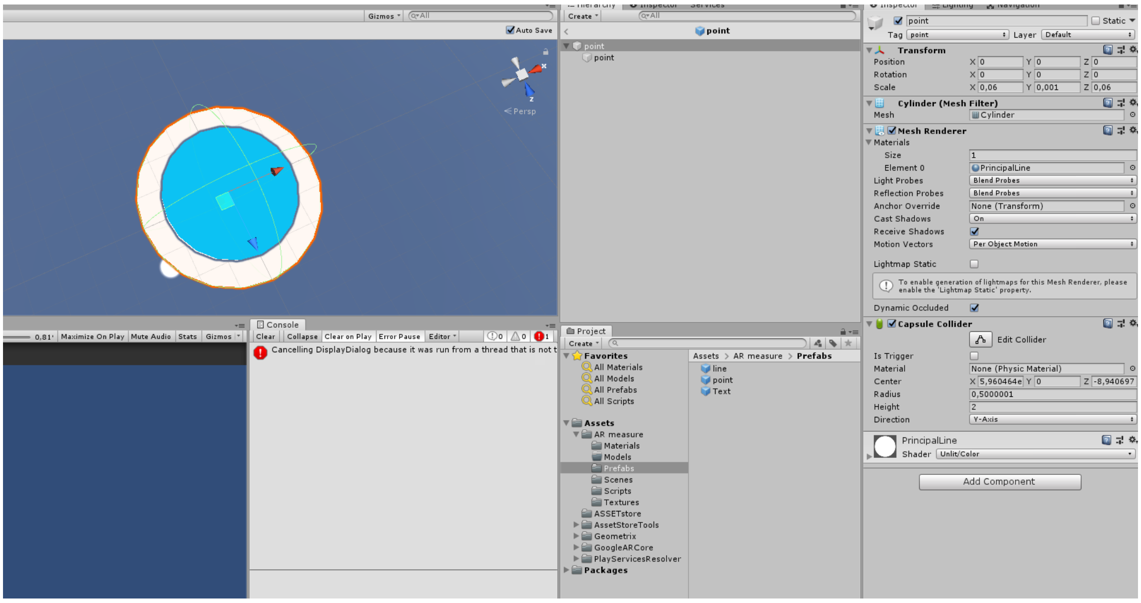
All gameobjects are untagged at the moment.

3.Prefabs

The prefabs consist on the following gameobjects:



Point: Used as the elemental part of the game to measure distances.

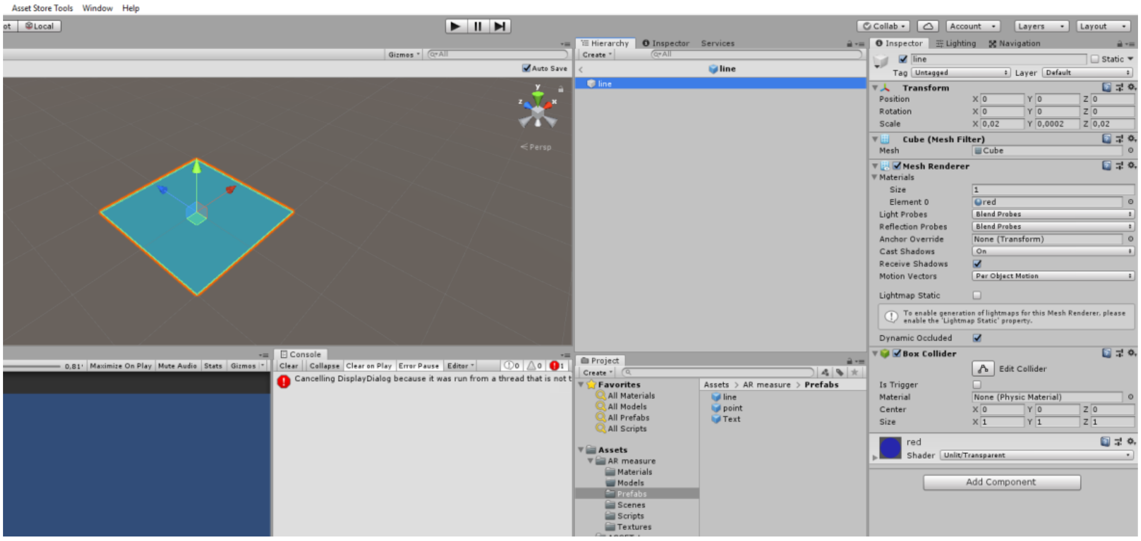


Cancelling DisplayDialog because it was run from a thread that is not t

```
☑ point                                   □ Static
Tag  point          Layer  Default
Transform
Position      X 0           Y 0        Z 0
Rotation      X 0           Y 0        Z 0
Scale         X 0,06        Y 0,001    Z 0,06
Cylinder (Mesh Filter)
Mesh                 Cylinder
☑ Mesh Renderer
Materials
  Size               1
  Element 0          PrincipalLine
Light Probes         Blend Probes
Reflection Probes    Blend Probes
Anchor Override      None (Transform)
Cast Shadows         On
Receive Shadows      ☑
Motion Vectors       Per Object Motion

Lightmap Static      □
To enable generation of lightmaps for this Mesh Renderer, please enable the 'Lightmap Static' property.
Dynamic Occluded     ☑
☑ Capsule Collider
Is Trigger           □
Material             None (Physic Material)
Center        X 5,960464e  Y 0     Z -8,940697
Radius               0,5000001
Height               2
Direction            Y-Axis
PrincipalLine
Shader               Unlit/Color
```

Line: it is created from one point to another.



Cancelling DisplayDialog because it was run from a thread that is not t

☑ line □ Static
Tag Untagged Layer Default

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 0,02 Y 0,0002 Z 0,02

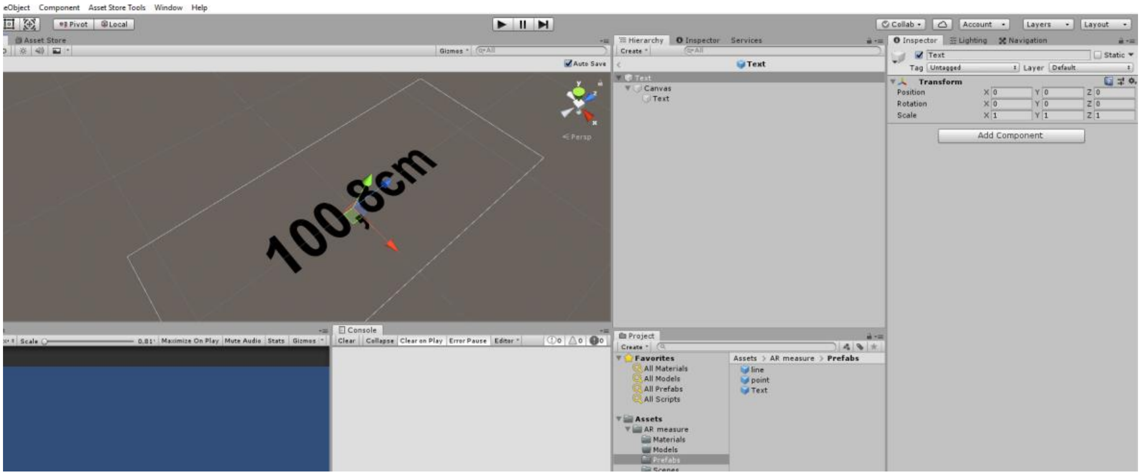
Cube (Mesh Filter)
Mesh Cube

☑ Mesh Renderer
Materials
Size 1
Element 0 red
Light Probes Blend Probes
Reflection Probes Blend Probes
Anchor Override None (Transform)
Cast Shadows On
Receive Shadows ☑
Motion Vectors Per Object Motion
Lightmap Static □
To enable generation of lightmaps for this Mesh Renderer, please enable the 'Lightmap Static' property.
Dynamic Occluded ☑

☑ Box Collider
Edit Collider
Is Trigger □
Material None (Physic Material)
Center X 0 Y 0 Z 0
Size X 1 Y 1 Z 1

red
Shader Unlit/Transparent

Text: displays the distance between two points.

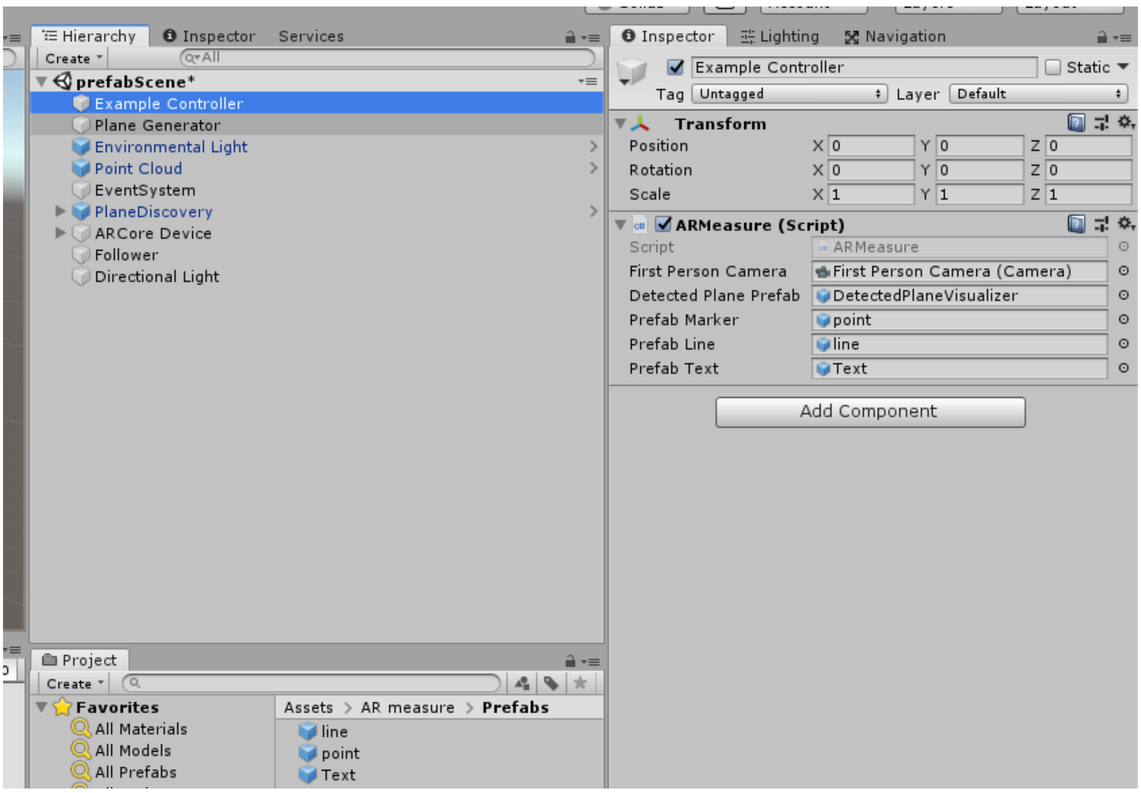


☑ Text □ Static
Tag Untagged Layer Default

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

4.Scripting

The main script is the one used to measure distances:



☑ Example Controller □ Static

Tag Untagged Layer Default

Transform

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1	1	1

☑ ARMeasure (Script)

Script	ARMeasure
First Person Camera	First Person Camera (Camera)
Detected Plane Prefab	DetectedPlaneVisualizer
Prefab Marker	point
Prefab Line	line
Prefab Text	Text

```
// Compensate for the hitPose rotation facing away from the raycast (i.e.
    // camera).
    points[p_index].transform.Rotate(0, k_ModelRotation, 0, Space.Self);

    // Create an anchor to allow ARCore to track the hitpoint as understanding 
of
    // the physical world evolves.

    // Make Andy model a child of the anchor.
    points[p_index].transform.parent = anchor.transform;



    //isntantiate line if number of points is greater than 2
    if (p_index>0)
    {
        /////////////////////////////////
        // MEASURE DISTANCES
        ///////////////////////////////////
        //direction and center of the line
        Vector3 dir = points[p_index].transform.position-points[p_index-
1].transform.position;
        Vector3 center = points[p_index-1].transform.position+dir/2;

        lines[l_index]=Instantiate(prefabLine, center, hit.Pose.rotation);
        lines[l_index].transform.rotation =Quaternion.LookRotation(dir, 
points[p_index].transform.up);
        lines[l_index].transform.localScale =new 
Vector3(lines[l_index].transform.localScale[0], 
lines[l_index].transform.localScale[1], dir.magnitude);

        // Make line model a child of the anchor.
        lines[l_index].transform.parent = anchor.transform;
        //instantiate text and set value too
        texts[t_index] = Instantiate(prefabText, center, hit.Pose.rotation);
        if (dir.x>0)
        {
            texts[t_index].transform.rotation = Quaternion.LookRotation(dir,
points[p_index].transform.up);
        }
        else
        {
            texts[t_index].transform.rotation = Quaternion.LookRotation(-dir,
points[p_index].transform.up);
        }
        UnityEngine.UI.Text txScp =
texts[t_index].transform.GetChild(0).transform.GetChild(0).GetComponent<UnityEn
gine.UI.Text>();
        txScp.text =Mathf.Round(dir.magnitude*1000)/10+"cm";

        // Make texts model a child of the anchor.
        texts[t_index].transform.parent = anchor.transform;

        t_index += 1;
        l_index +=1;
    }

    spawn = true;
    p_index += 1;

}
```

5.VIDEO TUTORIAL

We intend to give our customer the best service. To this aim, we upload tutorial videos of the asset always:

https://youtu.be/WlNp1T7Dl9I