# Part one: 3d reconstruction

## Set up the project

1. Create a new Unity project named Tango Workshop.
2. File > Build Settings > change **Platform** to Android.
   1. Player Settings > change **Bundle Identifier** to com.TangoWorkshop.Example.
3. Assets > Import Package > Custom Package… > Import Tango package (Gankino release).
4. Assets > Import Package > Custom Package… > Import workshop package.
5. Delete Main Camera.

## Add tango prefabs and reconstruction scripts

1. Project tab > Assets > TangoPrefabs > add Tango Manager to the scene.
   1. Check **Enable Depth**.
   2. Check **Enable Video Overlay**.
      1. Change **Method** to Texture and Raw Bytes.
   3. Check **Enable 3D Reconstruction (Experimental)**.
      1. Change **Resolution (meters)** to 0.05.
      2. Check **Generate Color**.
2. Project tab > Assets > TangoPrefabs > add Tango Camera to the scene.
3. Hierarchy tab > Create > Create Empty.
   1. Rename to Dynamic Mesh.
   2. Project tab > Assets > TangoSDK > Examples > Common > Scripts > add TangoDynamicMesh.
   3. Inspector tab > Add Component > Mesh > Mesh Renderer.
      1. Materials > change **Element 0** to unlit\_vertex\_color.
   4. Inspector tab > Add Component > Physics > Mesh Collider.
4. Hierarchy tab > Create > Create Empty.
   1. Rename to GUI.
   2. TangoSDK > Examples > ExperimentalMeshBuilderWithColor > Scripts > add MeshBuilderWithColorGUIController.

## add the workshop game manager prefab

1. Edit > Project Settings > Physics > Gravity > change **Y** component to -1.
2. Assets > TangoWorkshop > Prefabs > add Game Manager to the scene.
3. File > Save Scene > choose a filename for the scene.
4. File > Build & Run > (change to Android again if needed) choose a filename for the APK.

1.1 - A color mesh is generated and simple shapes can be placed or thrown.

# part two: augmented reality

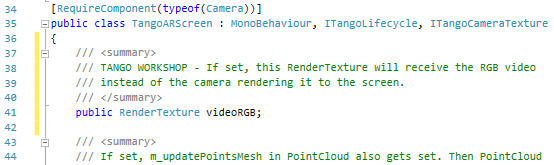
## add tango augmented reality camera prefab

1. Hierarchy tab > select Tango Camera.
   1. Inspector tab > Camera > Clipping Planes > increase **Far** to 10 (or larger).
   2. Inspector tab > Add Component > search for and add Tango AR Screen.
2. File > Build & Run.

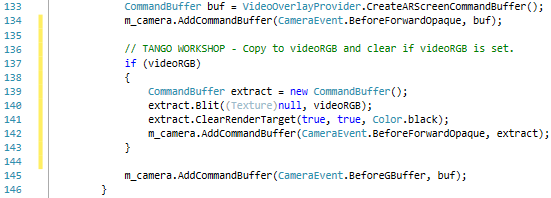
2.1 - The camera’s RGB video now appears behind the less accurate Dynamic Mesh.

## extract rgb video as a render texture

1. Tango Manager > uncheck **Generate Color**.
2. Project tab > Assets > Tango Workshop > Textures > Create > Render Texture.
   1. Rename to VideoRGBTexture.
   2. Change **Size** to match device:
      1. 1920 x 1200 for the Yellowstone development kit (landscape).
      2. 1440 x 2560 for the Lenovo Phab 2 Pro (portrait).
3. Tango AR Camera > Tango AR Screen (Script) > Edit Script.
   1. Line 37: declare a RenderTexture reference:



* 1. Line 135: insert a custom CommandBuffer when the RenderTexture is set:



1. Hierarchy tab > select Tango Camera.
   1. Inspector tab > Tango AR Screen (Script) > change **Video RGB** to VideoRGBTexture.
2. With VideoRGBTexture visible in the preview window, press the editor play button. (Ignore NullReferenceException from MeshBuilderWithColorGUIController).

2.2 - The Render Texture shows the emulated room and the camera display is black.

## write a shader to project the rgb video on to the dynamic mesh

1. Project tab > Assets > TangoWorkshop > Shaders > Create > Shader > Standard Surface Shader.
   1. Rename to ARProjectionShader.
   2. Emulate the shader seen below:



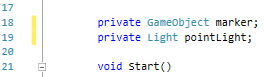
1. Assets > TangoWorkshop > Materials > Create > Material.
   1. Rename to ARProjectionMaterial.
   2. Change **Shader** to Tango Workshop > AR Projection (matches line 3 above).
   3. Change **Video RGB Texture** (matches line 5 above) to VideoRGBTexture.
2. Dynamic Mesh > Mesh Renderer > Materials > change **Element 0** to ARProjectionMaterial.
3. File > Build & Run.

2.3 - The render texture is projected on to the dynamic mesh.

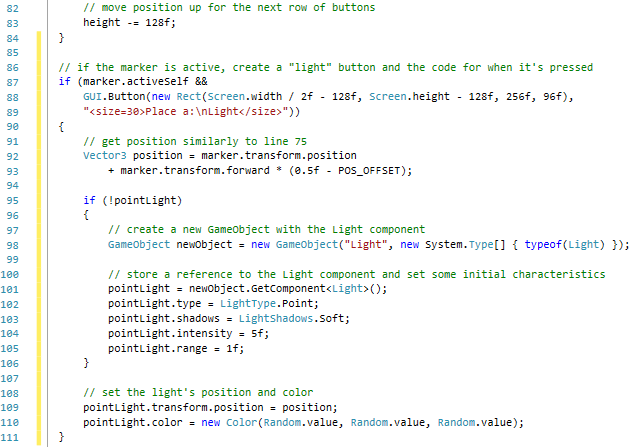
# part three: lighting and point cloud

## Extend the workshop Game manager to create and move a point light

1. Game Manager > Game Manager (Script) > Edit Script.
   1. Line 19: declare a Light reference:



* 1. Line 85: write GUI code for creating and moving a point light:

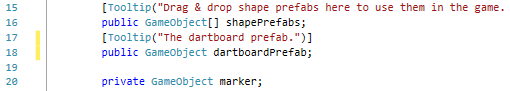


1. File > Build & Run.

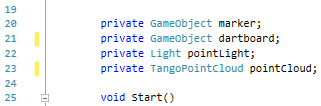
3.1 - A point Light can now be created and relocated in augmented reality

## create a dartboard game that detects walls using tango point cloud

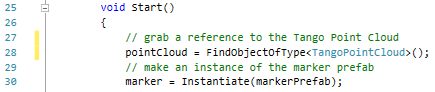
1. Assets > TangoPrefabs > add Tango Point Cloud to the scene.
2. Game Manager > Game Manager (Script) > Edit Script.
   1. Line 17: declare a GameObject reference for the dartboard prefab:



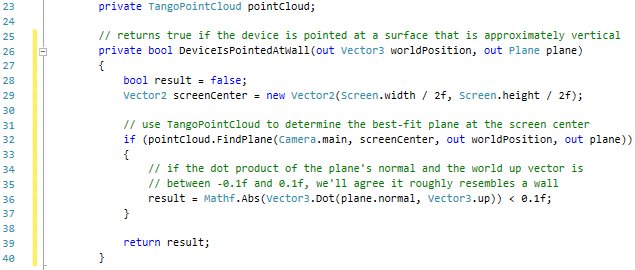
* 1. Lines 21 and 23: declare a GameObject reference for instantiating a dartboard and a TangoPointCloud reference:



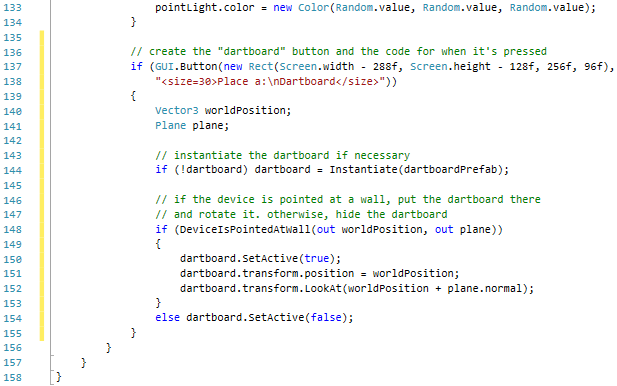
* 1. Line 27: grab a reference to the TangoPointCloud attached to Tango Point Cloud:



* 1. Line 25 (before void Start()): write a method for detecting a wall surface:



* 1. Line 135: write GUI code for creating, moving and rotating the dartboard:



1. Game Manager > Game Manager (Script) > change **Dartboard** to Dartboard.
2. File > Build & Run.

3.2 – A dartboard can now be created when a wall is detected at the center of the screen