Workshop 2 Lab1

1. Create a new Unity 3D project.
2. Create an empty GameObject, rename it as MeshGenerator
3. Add in Mesh Filter and Mesh Renderer component to MeshGenerator object.

Graphical user interface, text, application, email

Description automatically generated

1. New a C# script, MeshGenerator

Graphical user interface, text, application

Description automatically generated

Note: to support culling in Unity, polygon must be specified in a clockwise direction to be visible (i.e., the front side).

The “front” side is the clockwise face. The “back” side is the anti-clockwise face.

|  |  |
| --- | --- |
| Diagram  Description automatically generated | Chart, surface chart  Description automatically generated |

Chart, radar chart

Description automatically generated

1. Attached the C# script MeshGenerator to MeshGenerator object.
2. Run the scene.

A picture containing background pattern

Description automatically generated

1. Add in material to the Mesh Renderer

Graphical user interface, application

Description automatically generated

1. Adjust the light to see the reflection.

A picture containing sport

Description automatically generated

1. Add in one more polygonA picture containing calendar

   Description automatically generated
2. Run the scene. Rotate the object around and observe that the new polygon “disappeared” at certain angle (it has been culled).

Adjust the triangle array sequence (0,4,1) and observe the effect as well.

|  |  |
| --- | --- |
| Radar chart  Description automatically generated | A picture containing kite  Description automatically generated |