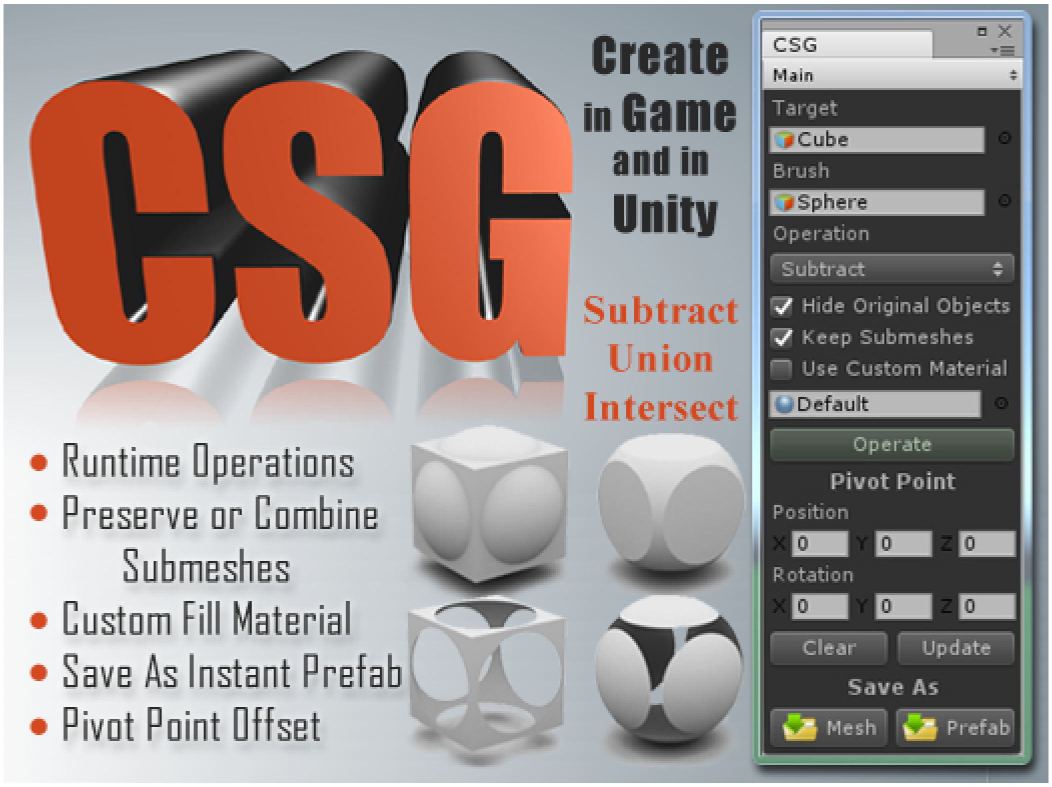


Target: Cube
Brush: Sphere
Operation: Subtract
☑ Hide Original Objects
☑ Keep Submeshes
☐ Use Custom Material
Default
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0

Constructive Solid Geometry in Game or in Unity

CSG allows you to create new meshes by performing Subtraction, Union and Intersection between objects. You can perform the operations in the Unity editor or at runtime. The resulting geometry preserves the texture UV coordinates of the original meshes. The program also handles rotated and scaled objects as expected and is suitable for objects with multiple materials and submeshes.

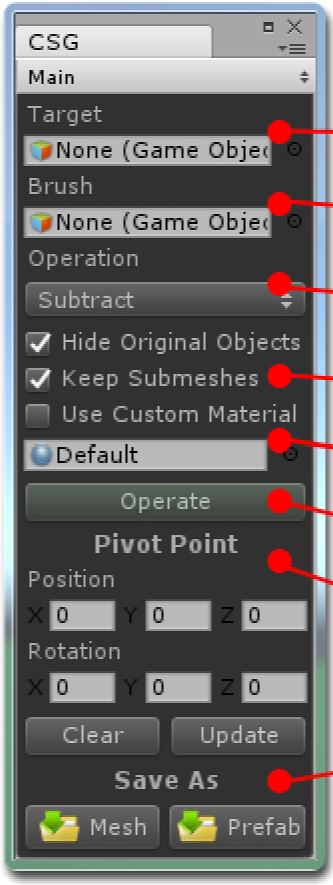
Starting The Program

To start the program first click on the Window menu item on the Unity menu bar, scroll down to the CSG menu item and then click on Open Window.

Performing Boolean Operations

Position the two objects you want to operate on within the scene so that they overlap each other. Select the game object you want to operate on (with a Mesh Filter component attached) from the hierarchy window and drag it into the Target slot. Do the same for the game object you want to use as a brush (also with a Mesh Filter component attached) and drag it into the Brush slot. Choose the type of operation you want to perform from the drop down list, select the options you want from the checkboxes and then click on the button marked Operate.

The newly created object will get its pivot point position from the target object by default. You can also add an offset to this position in the editor window if you would like to adjust the final position of the pivot point.



Target: None (Game Objec

Brush: None (Game Objec

Operation: Subtract

☑ Hide Original Objects

☑ Keep Submeshes

☐ Use Custom Material

Default

Position: X 0 Y 0 Z 0

Rotation: X 0 Y 0 Z 0

Add the target object to this slot

Add the brush object to this slot

Select the operation to perform here

Choose which options to use here

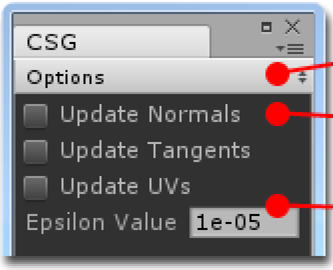
Select a custom fill material if needed

Click on the Operate button when ready

Optional pivot point offset in editor only

After the new object is created you can save it as a mesh or as an instant prefab

Options & Help Pages



☐ Update Normals

☐ Update Tangents

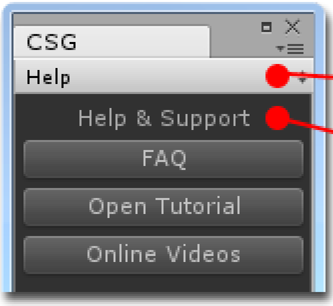
☐ Update UVs

Epsilon Value: 1e-05

Click on the toolbar to select the Options page

Check the additional options you want here

Change the epsilon value if problems occur



Click on the toolbar to select the Help page

Select the type of help required here

Basic Runtime Operations

In the CSG folder is a subfolder called Files and within this folder you will find the DLL for performing runtime operations via script. Simply add the DLL to your project, create a new script and copy the following code to it.

This is a simple script with the bare minimum required to perform CSG operations in game. Start by creating a new cube and sphere object in your scene. Position them so that they intersect each other and attach the script to any object in the scene before clicking on the play button.

Please note that you need to include the "using MeshMakerNamespace;" directive at the top of the script.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using MeshMakerNamespace;

public class RuntimeCSG : MonoBehaviour {

        // Use this for initialization
        void Start () {
                CSG csg = new CSG();
                csg.Brush = GameObject.Find("Sphere");
                csg.Target = GameObject.Find("Cube");
                csg.OperationType = CSG.Operation.Subtract;
                GameObject newObject = csg.PerformCSG();
        }

        // Update is called once per frame
        void Update () {

        }
}
```

Advanced Runtime Operations

The following example shows how to use the optional features available when using CSG at runtime.

The CSG class also includes a static epsilon value that you can adjust to improve the operations. This is used to calculate the splitting plane and smaller values can result in more accurate calculations.

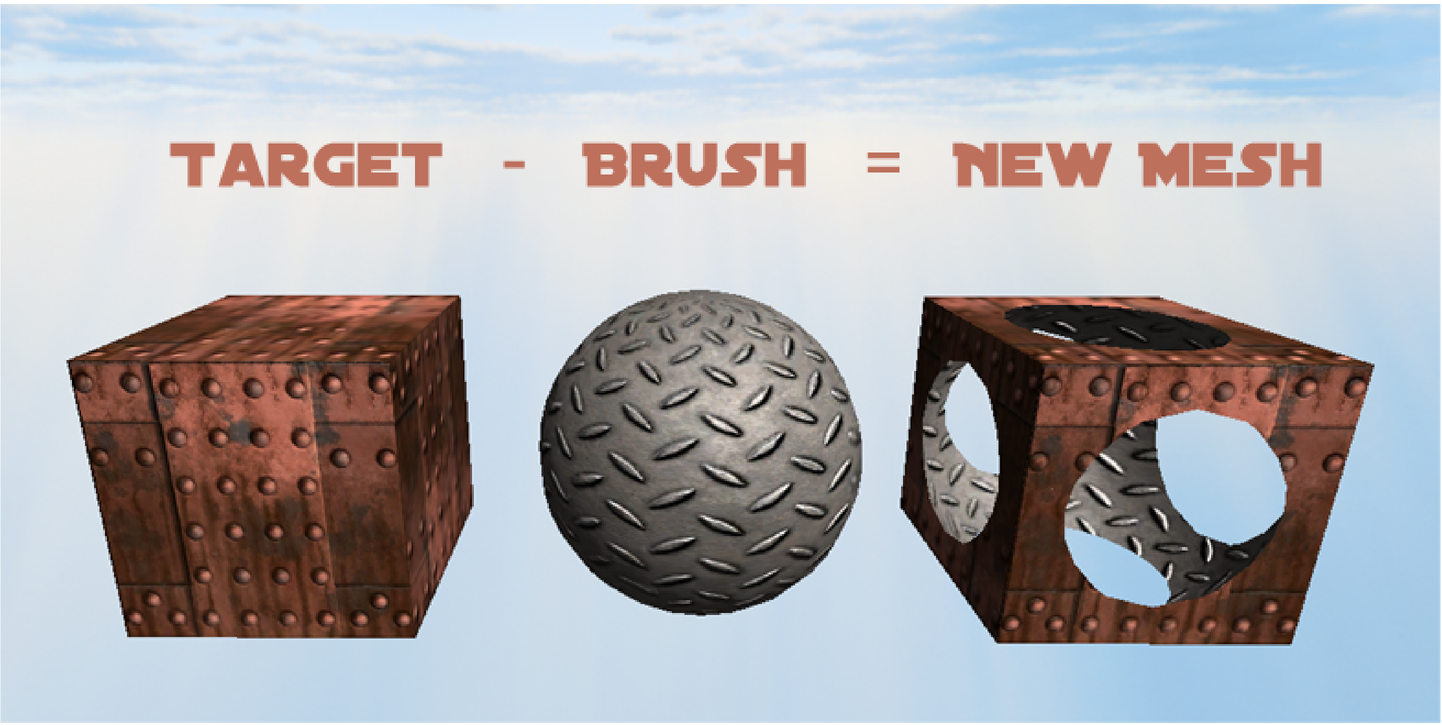
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using MeshMakerNamespace;

public class RuntimeCSG : MonoBehaviour {

	// Use this for initialization
	void Start () {
		CSG.EPSILON = 1e-5f;   // Adjustable epsilon value
		CSG csg = new CSG();
		csg.Brush = GameObject.Find("Sphere");
		csg.Target = GameObject.Find("Cube");
		csg.OperationType = CSG.Operation.Subtract;
		csg.customMaterial = new Material(Shader.Find("Standard"));   // Custom material
		csg.useCustomMaterial = false;   // Use the above material to fill cuts
		csg.hideGameObjects = true;   // Hide target and brush objects after operation
		csg.keepSubmeshes = true;   // Keep original submeshes and materials
		GameObject newObject = csg.PerformCSG();
	}

	// Update is called once per frame
	void Update () {

	}
}
```



Notes

The CSG operations cannot be performed on skinned meshes.

The newly created object may contain a lot of triangles and these are not optimized to reduce the triangle count.

It is not recommended to do more that a few operations on the same target object or the geometry may become corrupted.

While the program works well on simple objects it may fail on very complicated objects or objects with underlying problems in their geometry.

It is highly recommended that you save your work and scene before using this software.

To learn more about CSG and to view the tutorial videos you can visit
MeshMaker.com