

# ADVANCED PRIMITIVES

## USER MANUAL

This package allows you to create parametric 3D primitives straight from the *Unity Editor* and/or your own scripts. Each comes with its own parameters and (optional) collider. They all have their normals, uvs and (optional) tangents set. The following primitives are included:

- Capsule
- Cube
- Cylinder
- Disk
- Plane
- Pyramid
- Slope
- Sphere
- Torus

To create one of these primitives, you can either go to the **Game Object -> Create Other** menu or open the toolbox at **Window -> Advanced Primitives**.



Once you've created one, you can select it in the *Scene View* and look for the primitive component in the *Inspector*. The first settings are the same for every primitive:

- **Flip Normals:** Will "reverse" the mesh. It can be useful to create sky spheres etc.
- **Flip UVs Vertical:** The texture mapping will be flipped vertically.
- **Flip UVs Horizontal:** The texture mapping will be flipped horizontally. You may want to check this if you've flipped normals so the texture isn't mirrored.
- **Generate Tangents:** If you won't use a normal mapped material on this primitive, uncheck this to save some memory.
- **Optimize Mesh:** Calls `Mesh.Optimize()` on the primitive. You generally want to leave this on.
- **Generate Collider:** Will generate an adapted collider. Uncheck if you don't need to solve collisions for this primitive.
  - **Collider Details:** This parameter will only be available for the cylinder and the torus and takes a value in the range **[0.0 ; 1.0]**. It allows you to optimize the generated collider as much as possible. For example, if you create a detailed cylinder with 64 sides, the collider mesh will also have 64 sides by default, which would certainly be overkill. If you drop the **Collider Details** value down to 0.5, the collider will be created with only 32 sides (50%). Try to keep this value as low as possible for performance reasons.

You can also create or update primitives through code, which can be quite useful to optimize for low-end platforms.

Let say you need a cylinder with variable LOD in your scene (one for *iOS*, one for the *Wii* and one for everything else). If you were to use a modeling application, you would create three cylinders with 12, 24 and 32 sides, import them into Unity and use a script to switch to the correct one as needed.

With **Advanced Primitives**, you can now create your cylinder inside the *Unity Editor* and simply change its number of sides on the go. Here's what such a script would look like:

```
void Start()
{
    CylinderPrimitive cylinder = GetComponent<CylinderPrimitive>();

    #if UNITY_IPHONE
        cylinder.Sides = 12;
    #elif UNITY_WII
        cylinder.Sides = 24;
    #else
        cylinder.Sides = 32;
    #endif

    cylinder.UpdatePrimitive();
}
```

Don't forget to call `UpdatePrimitive()` once you've made your modifications to the primitive to rebuild the mesh. Jump to the next page for a complete API reference.

# API REFERENCE

## Commons

### UpdatePrimitive()

Rebuilds the mesh. Call it after you've changed some parameters. Beware that this method will rebuild the entire mesh for the given primitive, don't use it on every frame unless you really need to.

**FlipNormals** (bool, default: false)  
**FlipUVsVertical** (bool, default: false)  
**FlipUVsHorizontal** (bool, default: false)  
**GenerateCollider** (bool, default: true)  
**OptimizeMesh** (bool, default: true)  
**GenerateTangents** (bool, default: true)

## CapsulePrimitive

**Radius** (float, default: 0.5)  
**Height** (float, default: 2.0)  
**HeightSegments** (int, default: 1)  
**Rings** (int, default: 6)  
**Sides** (int, default: 12)

## CubePrimitive

**Width** (float, default: 1.0)  
**Length** (float, default: 1.0)  
**Height** (float, default: 1.0)  
**WidthSegments** (int, default: 1)  
**LengthSegments** (int, default: 1)  
**HeightSegments** (int, default: 1)

## CylinderPrimitive

**ColliderDetails** (float, default: 1.0)  
**Radius** (float, default: 0.5)  
**Height** (float, default: 1.0)  
**Sides** (int, default: 12)  
**HeightSegments** (int, default: 1)

## DiskPrimitive

**Radius** (float, default: 1.0)  
**Sides** (int, default: 12)

## PlanePrimitive

**Width** (float, default: 1.0)  
**Length** (float, default: 1.0)  
**WidthSegments** (int, default: 1)  
**LengthSegments** (int, default: 1)

## PyramidPrimitive

**Width** (float, default: 1.0)  
**Length** (float, default: 1.0)  
**Height** (float, default: 1.0)  
**Height Segments** (int, default: 1)

## SlopePrimitive

**Width** (float, default: 1.0)  
**Length** (float, default: 1.0)  
**Height** (float, default: 1.0)  
**Width Segments** (int, default: 1)  
**Length Segments** (int, default: 1)

## SpherePrimitive

**Radius** (float, default: 0.5)  
**LatitudeSegments** (int, default: 12)  
**LongitudeSegments** (int, default: 12)

## TorusPrimitive

**ColliderDetails** (float, default: 1.0)  
**OuterRadius** (float, default: 1.0)  
**InnerRadius** (float, default: 0.5)  
**Sides** (int, default: 12)  
**Segments** (int, default: 12)