Thank you for purchasing **PrimitivesPro**!

Version: 1.6.1

About

PrimitivesPro is a collection of 19 parametric primitives you can create inside Unity Editor or by using the script.

Parametric means that there are no geometric prefabs, everything is generated from the script.

The collection contains

- Triangle
- Plane
- Circle
- Ellipse
- Ring
- Box
- Sphere
- Cone
- Cylinder
- Ellipsoid
- Pyramid
- Geo-Sphere (also known as Ico-Sphere)
- Tube
- Capsule
- Rounded box
- Torus
- Torus knot
- Super-Ellipsoid
- Arc

19 Primitives
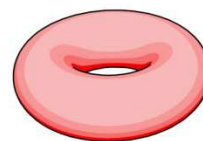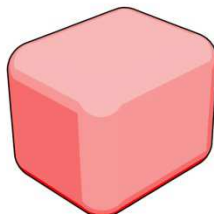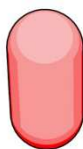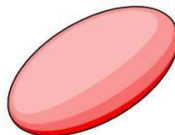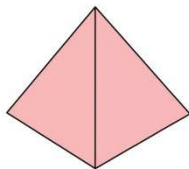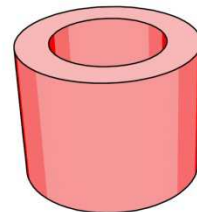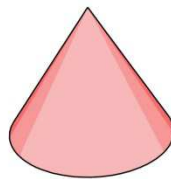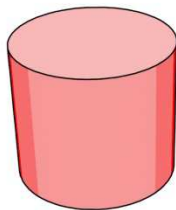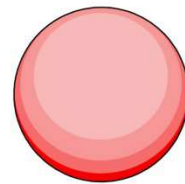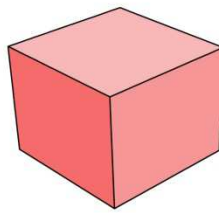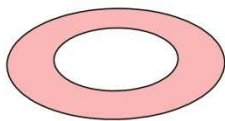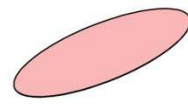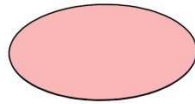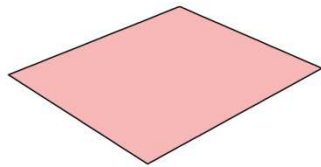
Creating Primitives inside Unity Editor

A primitive from the collection can be created inside Unity Editor from the main menu.

GameObject ->

    Create Other ->
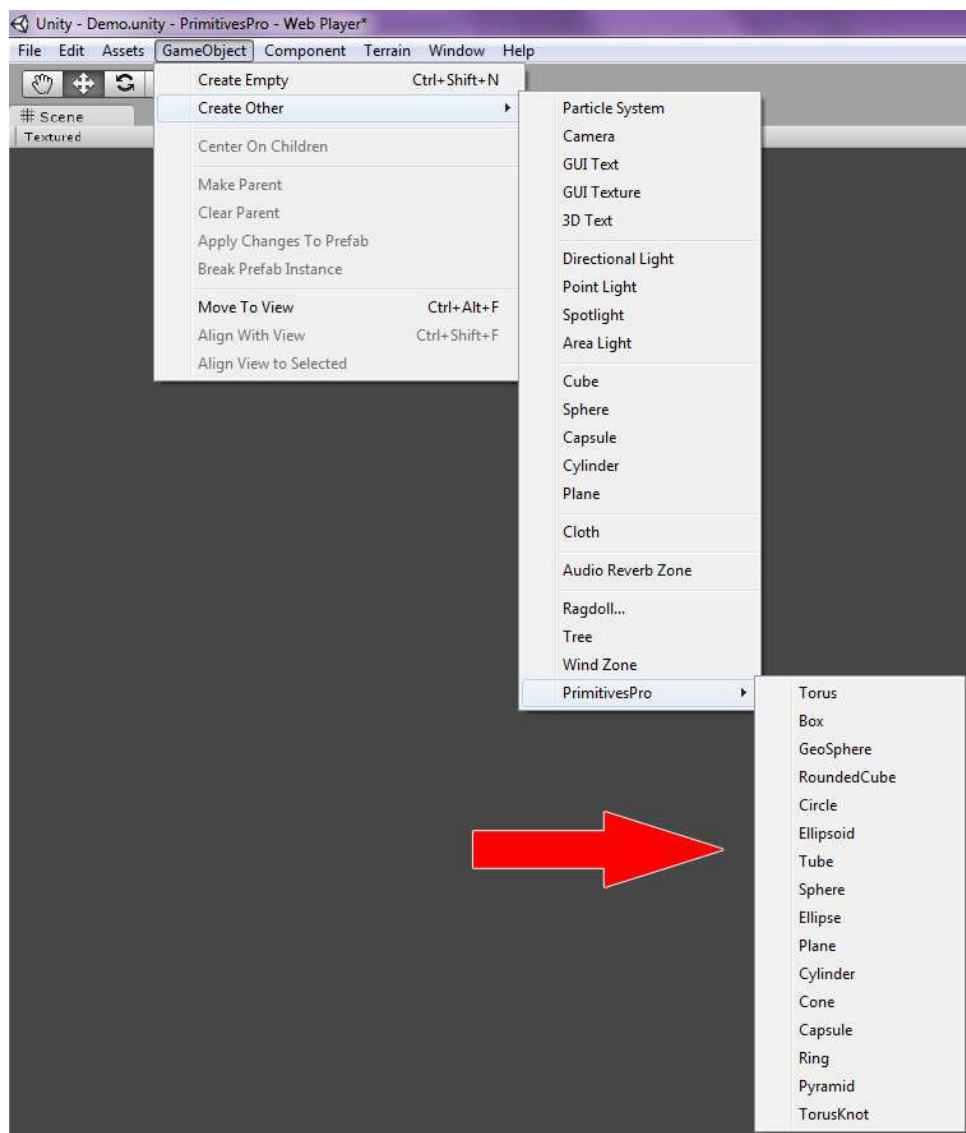
        PrimitivesPro -> …

Creating Primitives inside Unity Editor

Once a primitive is created you should see the model inside scene view. By default the position is set to [0, 0, 0].
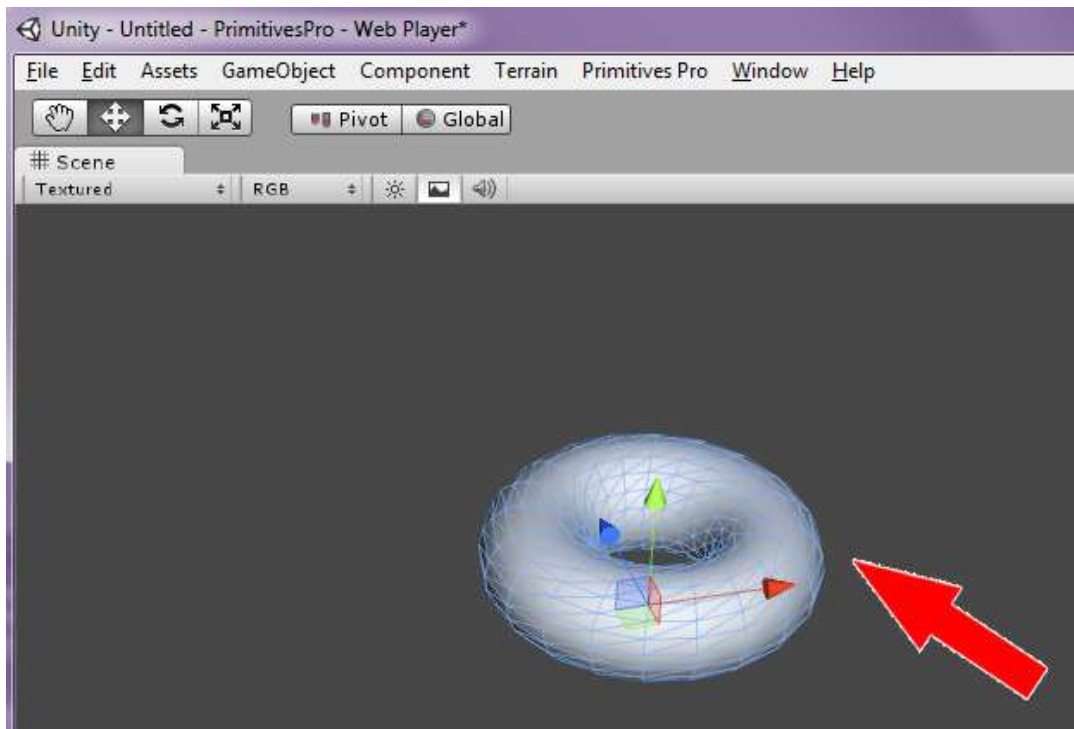
Diffuse material is assigned as a default material.
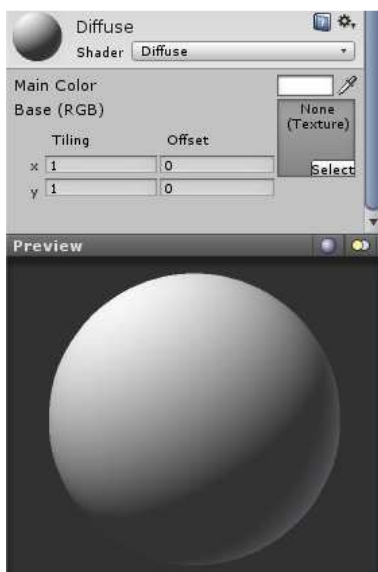
Creating Primitives inside Unity Editor

In the Inspector view you can adjust object parameters.
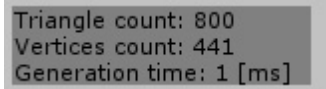
In this example you can adjust parameters to torus object such as

- torus radius

- cone radius

- number of torus segments

- number of cone segments

- type of normal vectors

- position of model pivot

- flipping the normal vectors

Creating Primitives inside Unity Editor

Note the small statistic window which can help you determine number of desired triangles or vertices.

```
Triangle count: 800
Vertices count: 441
Generation time: 1 [ms]
```

Since every object has different parameters we recommend experimenting by changing its values and see the results in scene view.

Generating normal vectors

Every 3D shape has an option to generate *Vertex* or *Face* normal vectors. By default all object are generated with *vertex* normals. It is a standard option for smooth shading and you will use it in most cases. However by selecting *Face* normals you can achieve a different look of the model.
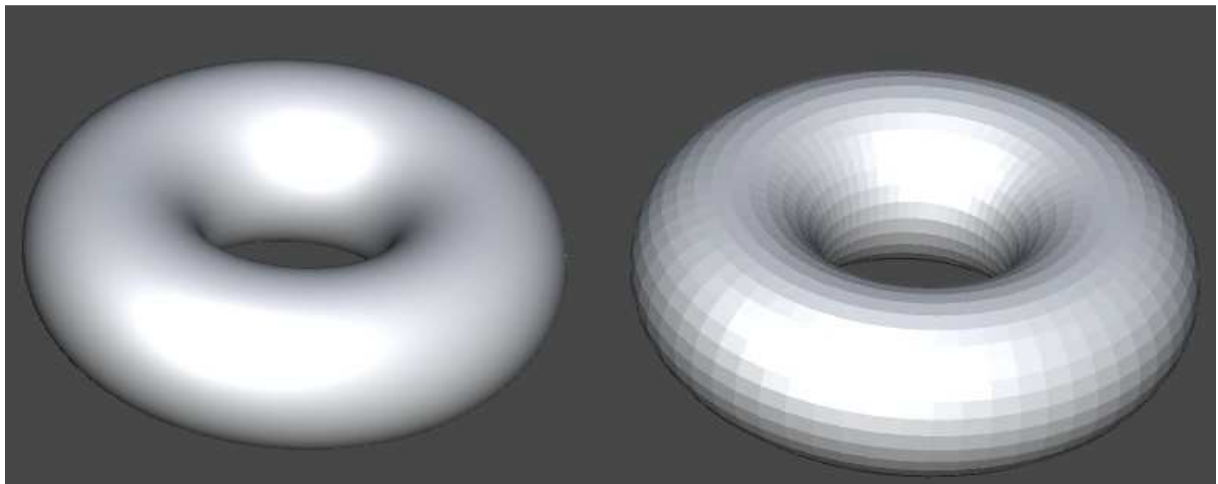
Vertex normals on the left, Face normals on the right.

IMPORTANT NOTE:

Because Unity supports only per-vertex normals, generating Face normals will result in higher number of vertices, up to 4x more than in Vertex case!

To minimize vertex count consider shader based solution for flat-shading effect.

Changing pivot position

Every 3D shape has an option to set position of the model Pivot to the bottom, center or the top of the model.
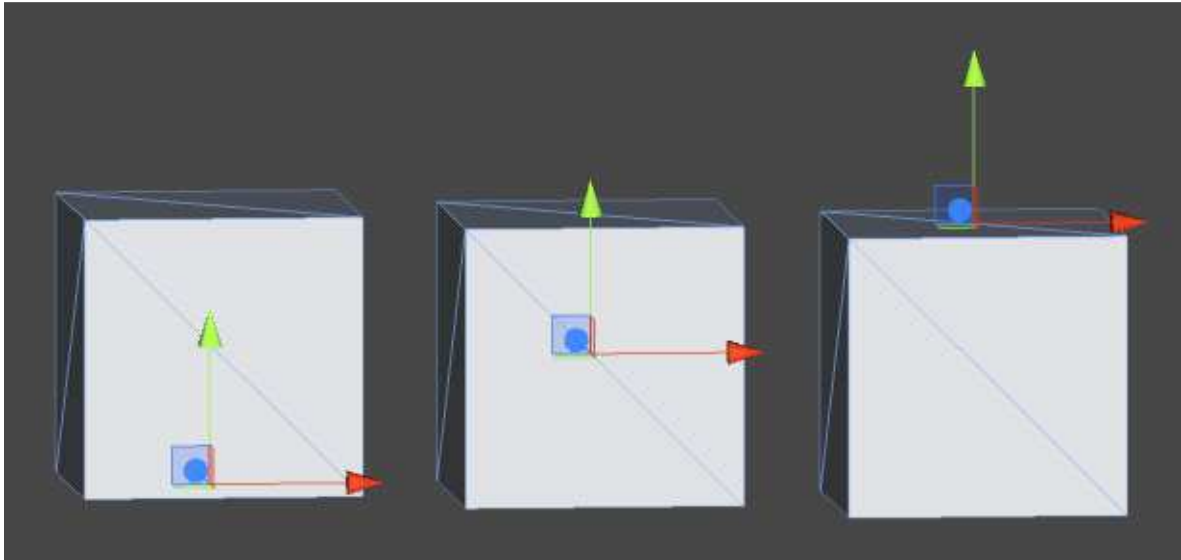
Left to right: Pivot in the bottom, in the center and on the top of the model.

Flipping normal vectors

Normal vectors can be flipped to opposite direction. This can be handy if you want to create a skybox by mapping a texture inside a cube. Or in general you want to view inside of the geometry.

Mesh cutter

From version 1.3 PrimitivesPro contains a "Mesh cutting" feature. It can be used inside editor or from the script. Mesh cutter can cut any mesh or any GameObject that contains a mesh.

First select cutting plane:

GameObject ->

    Create Other ->

        PrimitivesPro -> …

Manipulate cutting plane in scene view:

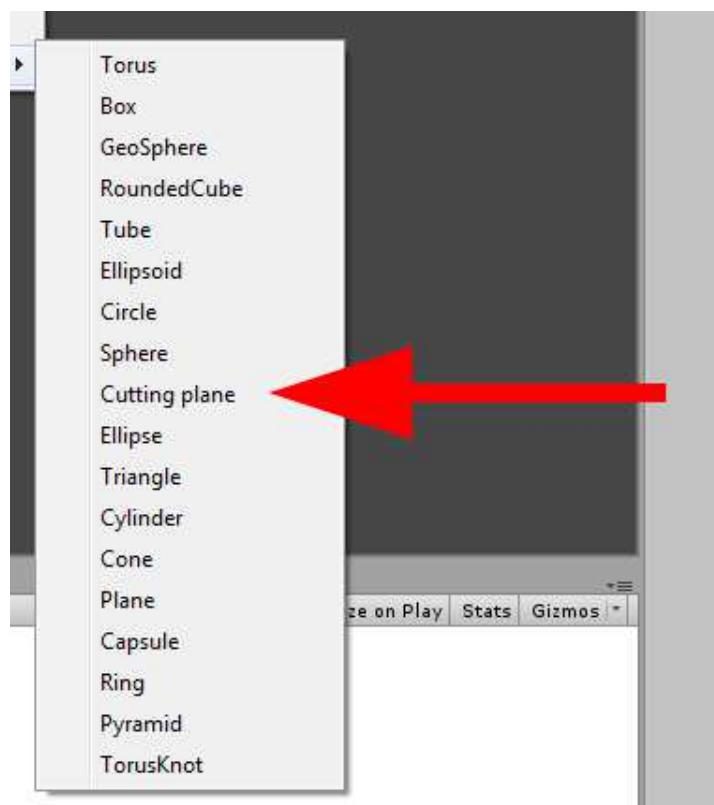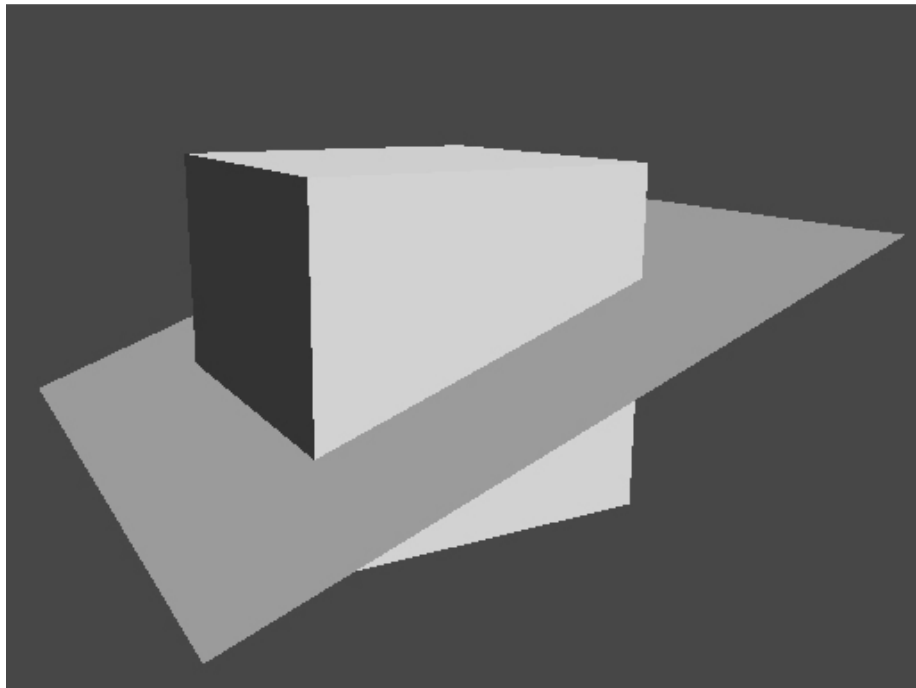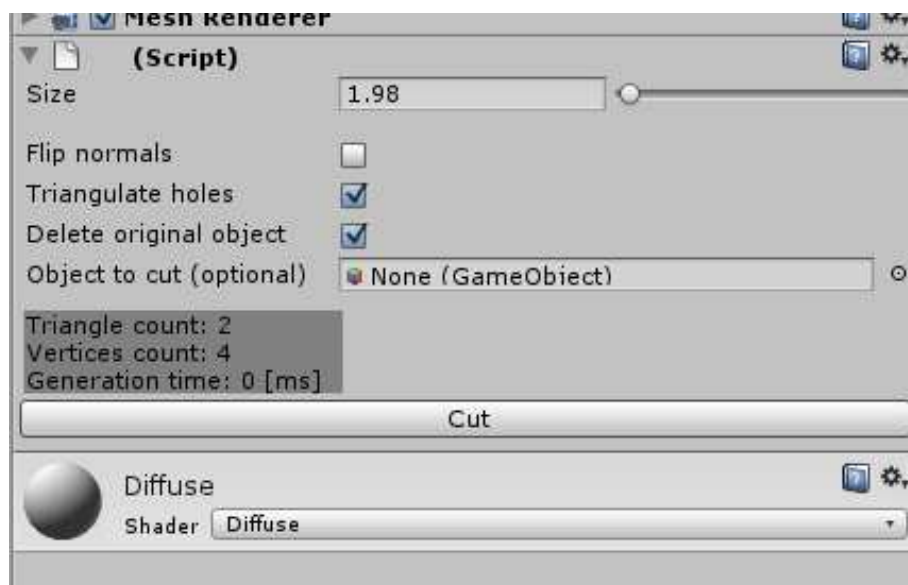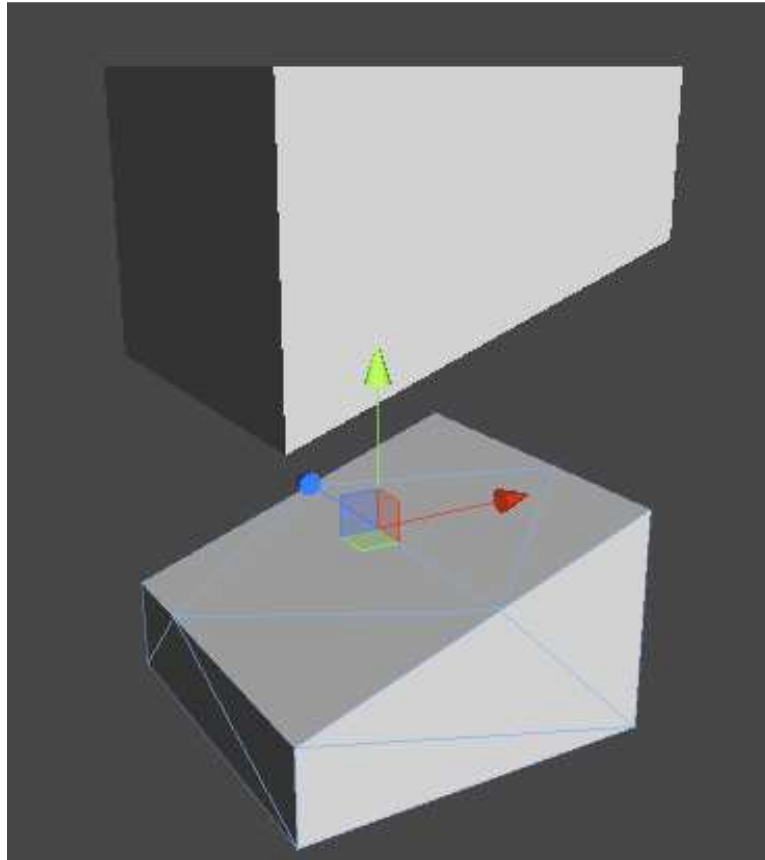Select "cutting plane" and press "Cut" in inspector view:

See the results:

Creating Primitives from the script

Creating a primitive from the script is very easy. All the important classes are encapsulated inside a namespace:

```
namespace PrimitivesPro
```

Generation algorithms are located inside namespace:

```
namespace PrimitivesPro.Primitives
```

Every Primitive object is a Unity Game object and has its own class derived from MonoBehaviour inside a namespace:

```
namespace PrimitivesPro.GameObjects
```

Example classes:

```
public class Box : BaseObject
public class Capsule : BaseObject
public class Circle : BaseObject
...
```

To create primitive objects just call a static method *Create* with parameters:

```
using PrimitivesPro.Primitives;
using PrimitivesPro.GameObjects;


// create a box game object
var box = Box.Create(1, 2, 3, 1, 1, 1, PivotPosition.Botttom);

// create a sphere object
var sphere = Sphere.Create(0.5f, 20, 0.0f, NormalsType.Vertex, PivotPosition.Center);

// create a torus object
var torus = Torus.Create(0.5f, 0.9f, 20, 20, NormalsType.Vertex, PivotPosition.Top);

// create a circle
var plane = Circle.Create(2, 20);


...
```

If you wish to change parameters of the object call method *GenerateGeometry:*

```
using PrimitivesPro.Primitives;
using PrimitivesPro.GameObjects;


// create a box game object
var box = Box.Create(1, 2, 3, 1, 1, 1, PivotPosition.Botttom);

// change parameters of the box
box.GenerateGeometry(3, 2, 1, 5, 5, 5, PivotPosition.Botttom);

// changing just one parameter
box.width *= 2;
box.GenerateGeometry();
```


To flip normals call method *FlipNormals:*

```
using PrimitivesPro.Primitives;
using PrimitivesPro.GameObjects;


// create a box game object
var box = Box.Create(1, 2, 3, 1, 1, 1, PivotPosition.Botttom);

// flip normals of the box
box.FlipNormals();
```


To add a Unity collider call method *AddCollider:*

```
using PrimitivesPro.Primitives;
using PrimitivesPro.GameObjects;


// create a box game object
var box = Box.Create(1, 2, 3, 1, 1, 1, PivotPosition.Botttom);

// create collider for the box
box.AddCollider();
```


To access Unity game object…

```
using PrimitivesPro.Primitives;
using PrimitivesPro.GameObjects;


// create a box game object
var box = Box.Create(1, 2, 3, 1, 1, 1, PivotPosition.Botttom);

// get unity game object and add a component
box.gameObject.AddComponent<Component>();
```

Support

If you have any questions or need a help with setting up the primitive please use Unity forum:

http://forum.unity3d.com/threads/173575-PrimitivesPro-RELEASED

You can also write me an e-mail:

gamesreindeer@gmail.com