Game Object Boundaries

Offline Documentation
Last update : January 31st 2019

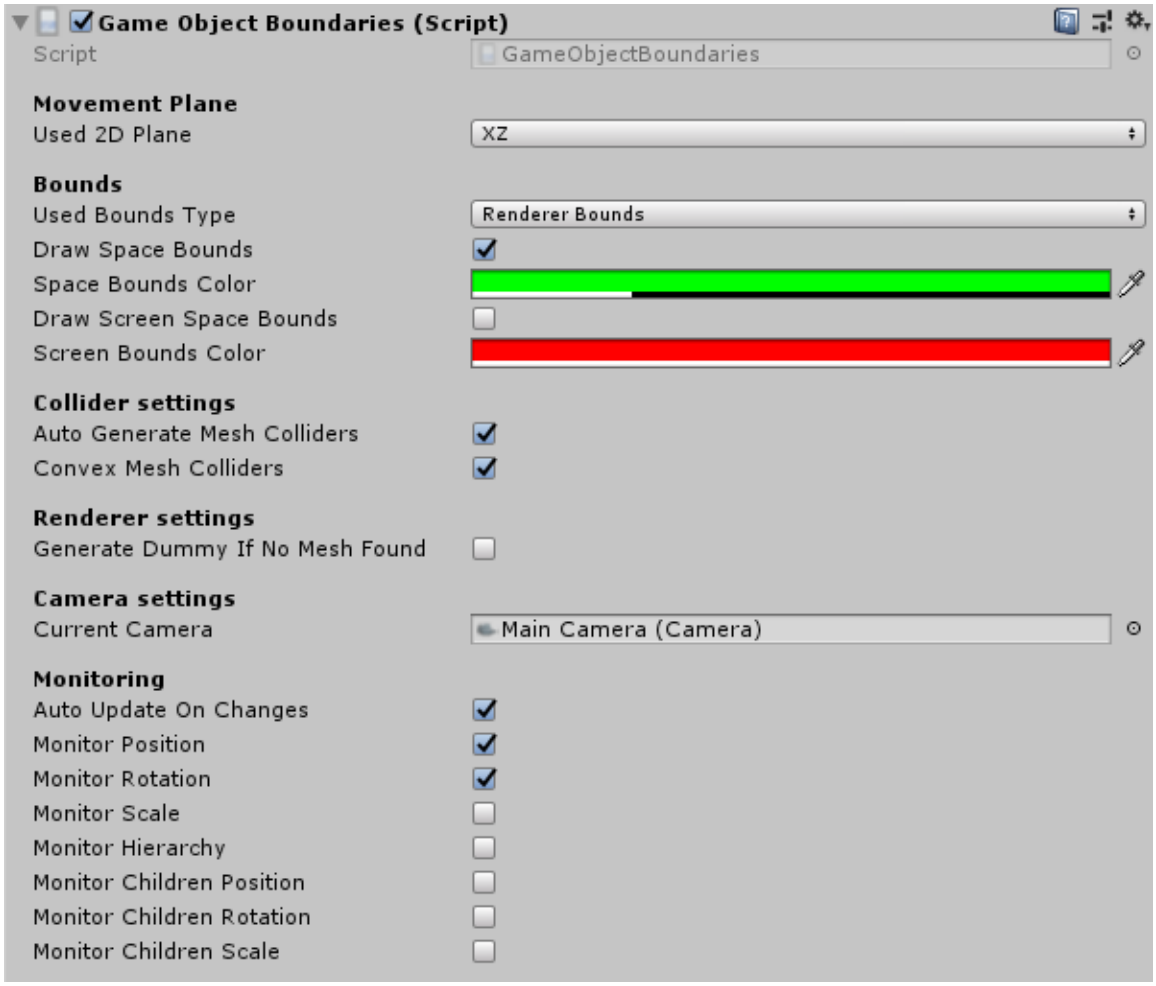
Introduction

Game Object Boundaries or GOB for short, Will allow you to get the bounds of a Gameobject with a simple function, taking in consideration the renderers, colliders and the hierarchy of the Gameobject, GOB also allows you to position the Gameobject below or above a certain point taking the bounds in consideration or scale the Gameobject to fit certain bounds, GOB can also get screen bounds from world bounds.

GOB supports some other calculations on a certain 2D plane, like the 2D diameter of the Gameobject or scaling it so that it fills a certain area of the screen.

GOB takes in consideration the necessary time to calculate the bounds of complicated Gameobjects, so it stores the results internally so that you update these only if changes has been done, speaking of changes GOB can monitor changes of the Gameobject and recalculate the bounds automatically but the monitoring process comes at a cost (CPU) which must be took in consideration.

Inspector Settings



Used 2D Plane : Select the corresponding plane for 2D calculations like 2D diameter.

Used Bounds Type : Select the bounds used for bounds calculations.

Draw Space Bounds : Check to display the 3D bounds of the Gameobject.

Draw Screen Space Bounds : Check to display the screen space bounds of the
Gameobject.

Auto Generate Mesh Colliders : Create colliders for renderers without colliders.

Convex Mesh Colliders : The automatically created colliders will be convex colliders (https://docs.unity3d.com/Manual/class-MeshCollider.html)

Generate Dummy If No Mesh Found : Create a dummy sphere Gameobject if no
Renderer was found.

Current Camera : Select the current Camera that will be used for screen space
calculations like the screen space bounds.

Auto Update On Changes : Automatically recalculate bounds if changes to the
Gameobject has been detected.

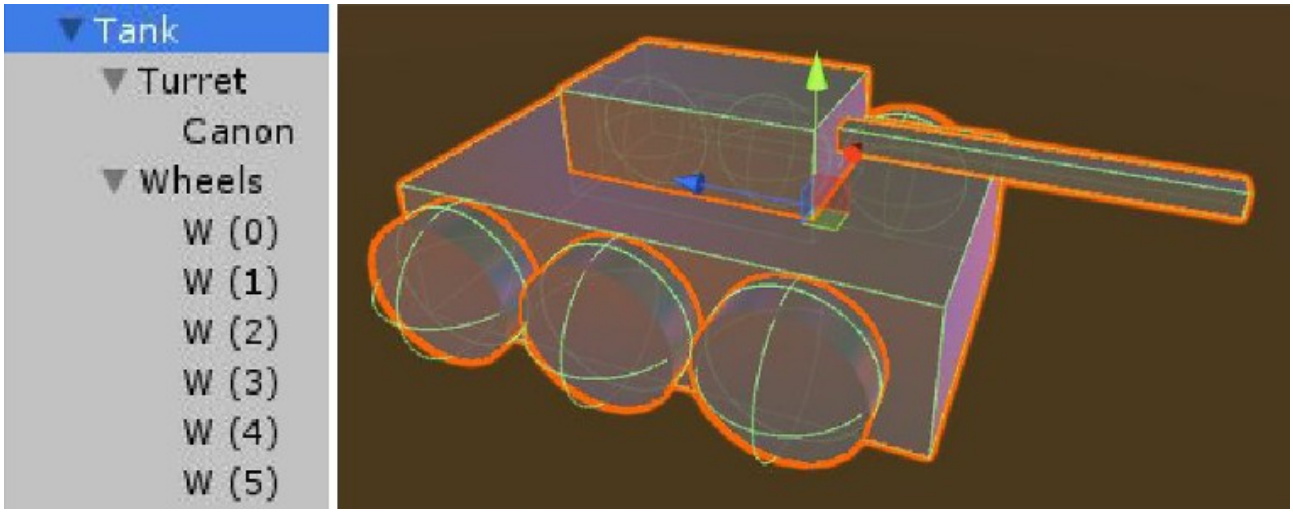
Monitor Position / Rotation / Scale : Monitor position, rotation and scaling changes to the parent Gameobject.

Monitor Hierarchy : Monitor children addition, removal or reordering.

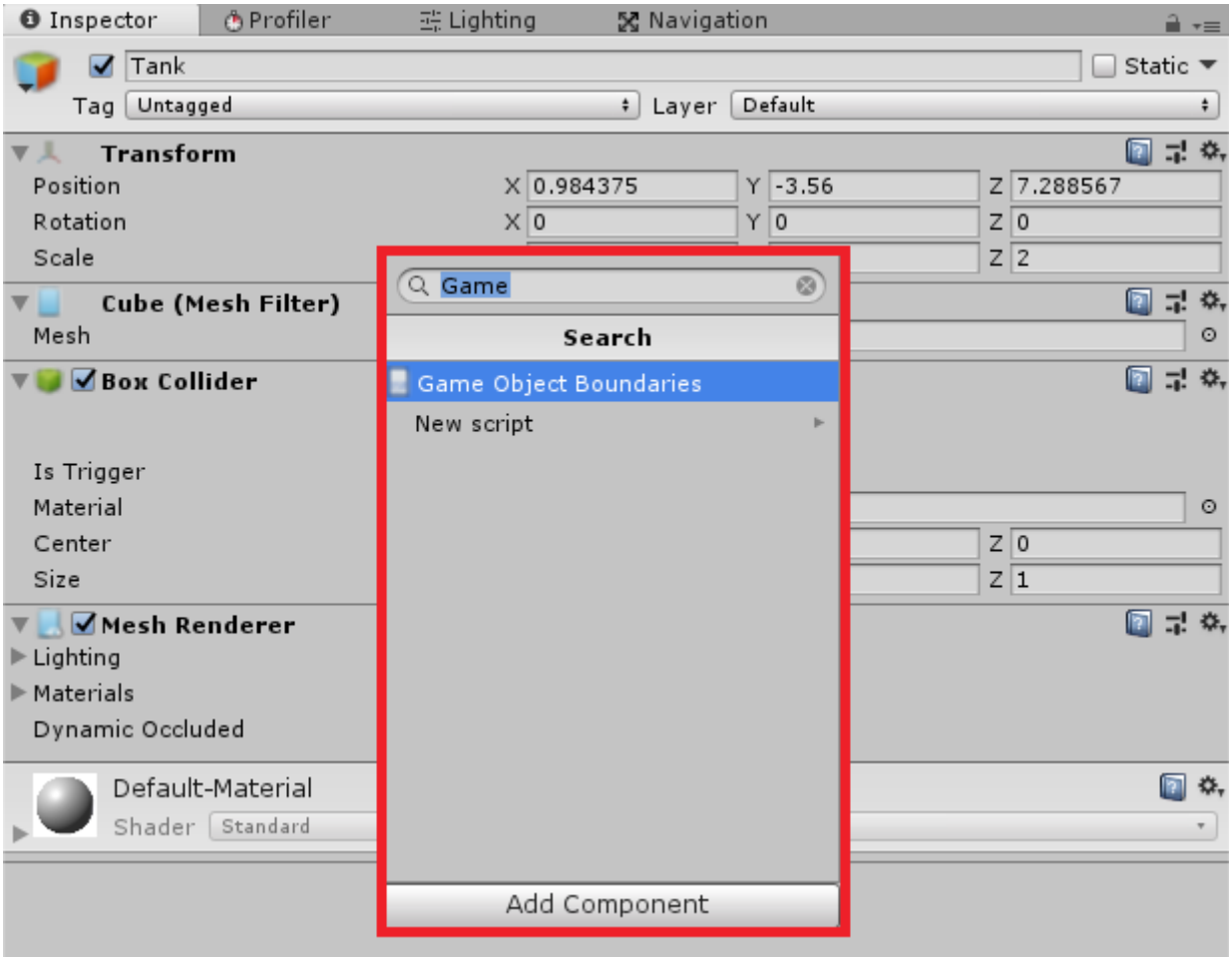
Monitor Children Position / Rotation / Scale : Monitor position, rotation and
scaling changes to any children of the parent Gameobject.

Usage Example

Step 1 : Create a Gameobject with children and sub-children.



Step 2 : Add a GameObjectBoundaries component.



Step 3 : Access the MetaComplexGameObject component from code and calculate the bounds using the GetBounds() Method.

```
GameObject go = GameObject.Find ("Tank");

GameObjectBoundaries component = go.GetComponent<GameObjectBoundaries> ();

Bounds bounds = component.GetBounds (true);

print (bounds);
```