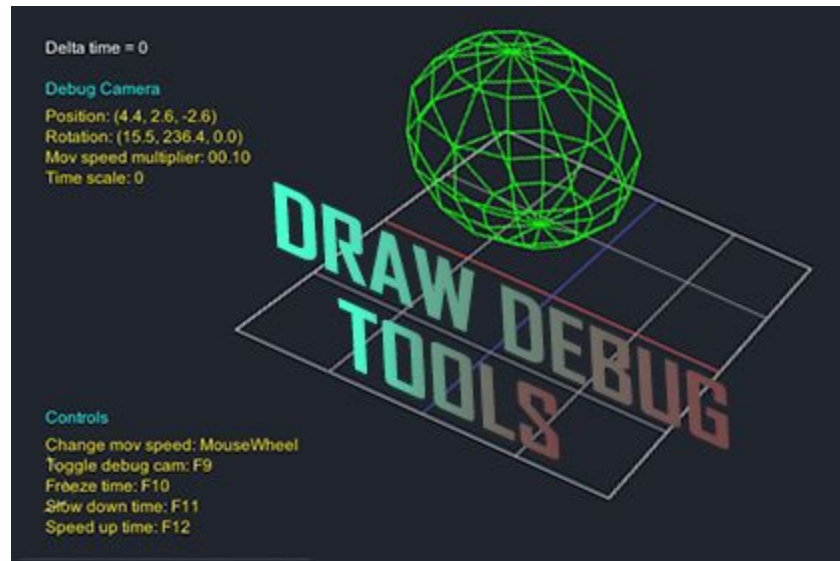


DRAW DEBUG TOOLS



OVERVIEW	2
FEATURES	2
FUNCTIONALITIES	2
COMING FUNCTIONALITIES	2
SETUP	3
EXAMPLE	4

Created by Mourad Bakhali

mourad.bakhali@gmail.com

GAMEDEVWORKS

www.gamedevworks.com

OVERVIEW

Developing gameplay mechanics and systems can be challenging but DrawDebugTools (DDT) make it easy for you by offering several draw functions to help you debug and know what your code is doing.

FEATURES

1. Very easy and intuitive to use
2. Easy to setup
3. Many examples that demonstrate how to use it
4. It has everything you need to visualize complex spatial debug data
5. It uses unity GL functions
6. Optimized to work without impacting your game performance
7. You can call draw functions from anywhere in your code
8. Works on Desktop, Web and Mobile (IOS is not supported)
9. Dedicated support

FUNCTIONALITIES

- **DebugCamera:** A detachable camera to help you move freely in your scene while running your game in editor or in standalone.
- **TimeControl:** While in debug camera mode you can stop time, slow it down or speed it up.
- **Object debug info:** While in debug camera mode you can see the name of the object you are looking at, you also see the materials applied to that object.
- **FloatGraph:** Visualize value changes of a given float variable.

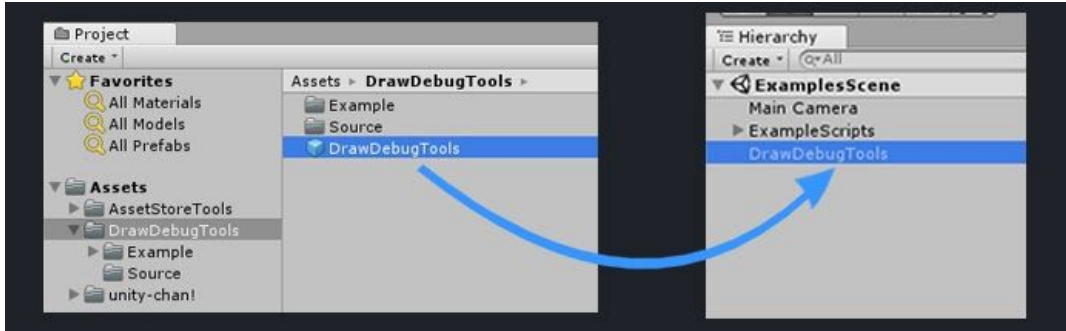
COMING FUNCTIONALITIES

- **FPS Graph:** Display an overlay graph that shows the changes of the frame rate per second over time.
- **Physic debugger:** Display collision shapes and other datas like contact points when a collision occurs, force vectors,

SETUP

To start working with DDT is very easy.

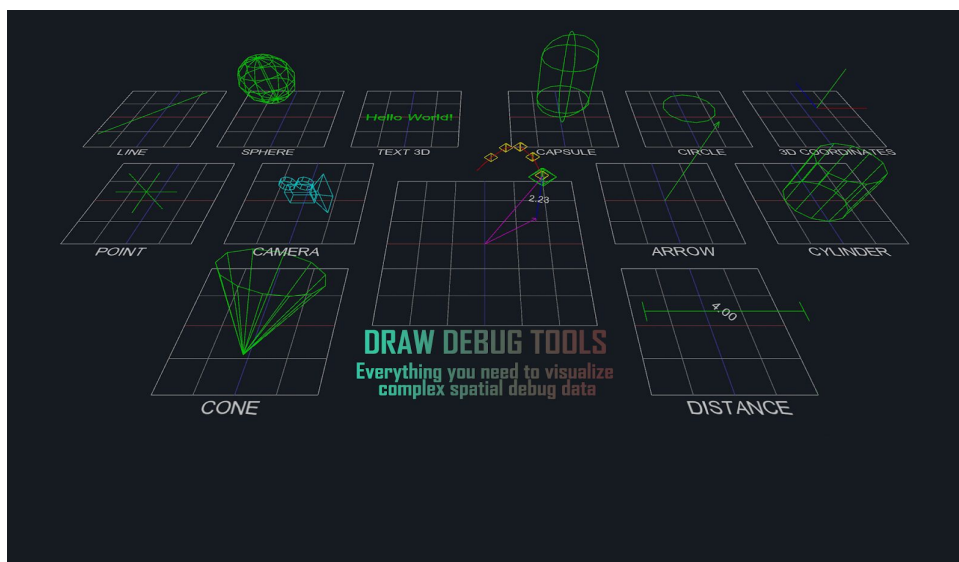
1. First, drag and drop DDT prefab in your scene:



2. Secondly, Select DDT prefab and set your active camera:



3. Finally, you are ready to call drawing functions !

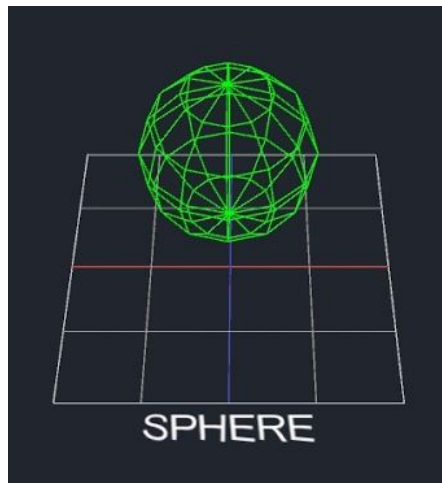


EXAMPLE

This is an example of drawing a sphere, grid, and a 3d text:

Code :

```
→ DrawDebugTools.DrawGrid(Position, GridSize, 1.0f, 1.0f);  
→ DrawDebugTools.DrawSphere(Position, 1.0f, 12, Color.Green);  
→ DrawDebugTools.DrawString3D(Position, Rotation, "SPHERE",  
    TextAnchor.LowerCenter, Color.White, 1.5f);
```



The signatures of the functions used are:

```
→ DrawDebugTools.DrawGrid(Vector3 Position, float GridSize, float CellSize,  
    float LifeTime = 0.0f);  
→ DrawDebugTools.DrawSphere(Vector3 Center, float Radius, int Segments,  
    Color Color, float LifeTime = 0.0f);  
→ DrawDebugTools.DrawString3D(Vector3 Position, Quaternion Rotation, string  
    Text, TextAnchor Anchor, Color TextColor, float TextSize = 1.0f, float  
    LifeTime = 0.0f);
```