

DimBoxes (6.0)

The package serves to draw the ingame dimension boxes around GameObjects using GL.Lines.

The dimension box is just a bounding box aligned to the GameObject space (AABB) with dimensions.

It is based on one single C# script:

- DimBox.cs script, which has to be attached to the GameObject

This is also the base for an extended version:

- DimBoxProgressive.cs script, with six progressive loading options added

The bounding box calculation is taken from the collection of the mesh.vertices from all the MeshFilters in the GameObject hierarchy. This is done in Editor just once on adding or resetting the script.

If you need a progressive loading animation, use the extended script version, otherwise use the base version.

Works both in editor and player mode.

Supports rotation, translation and scaling.

Scaling – on local coordinates only. (Reset the script before starting editor or player session if the object got scaled in the previous session).

The script is meant to showcase the dimensions of the real world objects, consisting of hierarchies of compound elements. (Note: avoid parenting to non-uniformly scaled transforms, because it will not produce good results.)

Sometimes you might prefer to showcase the measurements other than resulting from the meshes i.e. include some clearances or show in-axis distances in case of modular assemblies – so I have enabled an option of using the arbitrarily sized box collider as the measurement source. If that is a case - just attach and size a BoxCollider and check the “colliderbased” variable on the script in the inspector.

Colliders can also be needed to get an user interaction (on mouse events).

The dimension box offers various options for placing the dimensions and extension lines:

If the floating option selected – when the camera moves – the dimensions are supposed to change their positions in order to get the best visibility.

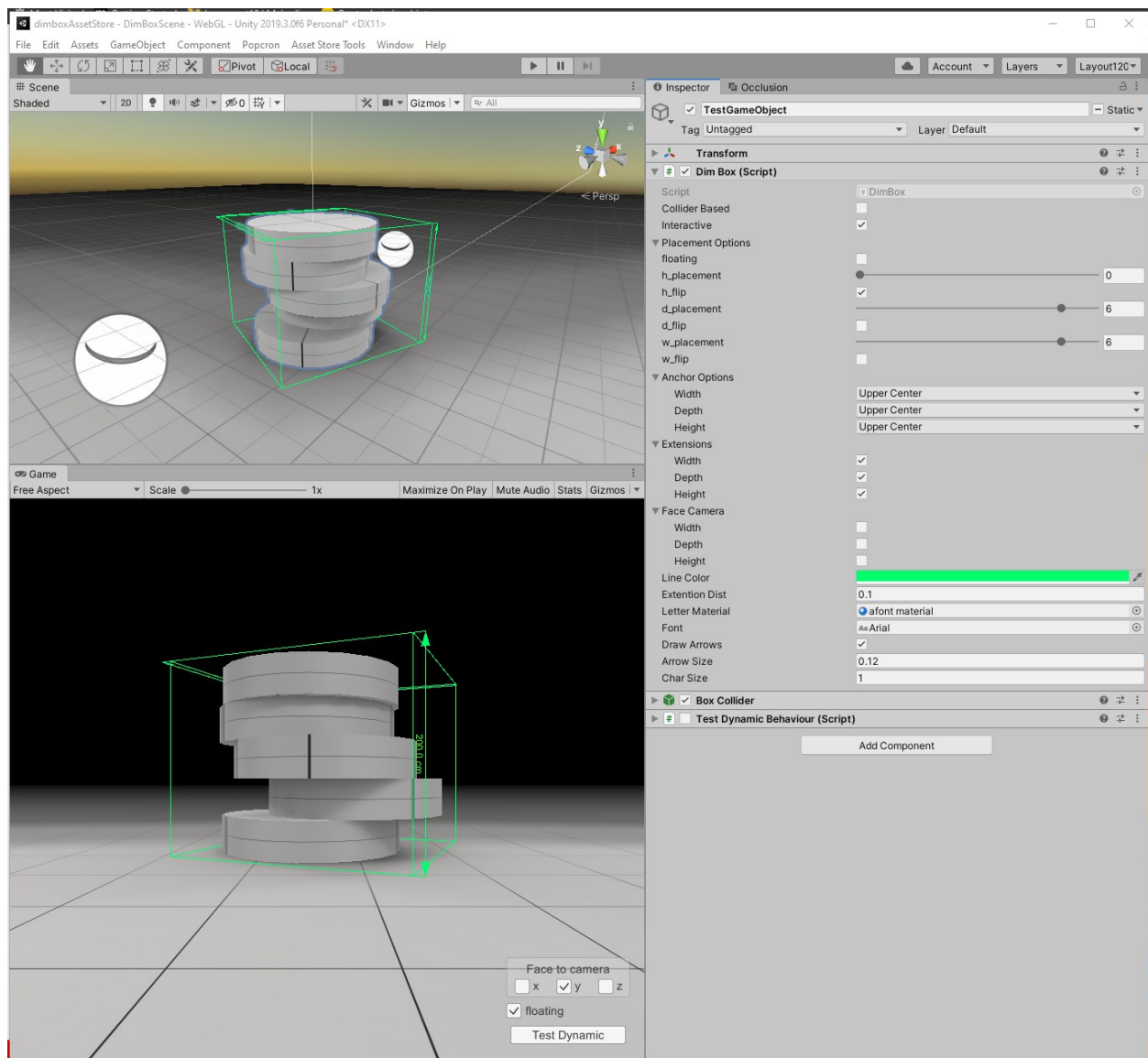
If the floating option deselected – the dimensions will stay at fixed positions, selected using the sliders.

The faceCamera switches turn the billboard on/off.

Warning: the faceCamera option can give unpredictable results when using non uniform scales.

The placing options relate to the pivot placement of the lines and the text mesh objects.

The anchor options enable the selection of two available text anchoring options to their pivots.



The asset works in Standard (Legacy) and Universal (URP) render pipelines. Works also in VR, tested on OculusGo.

I would greatly appreciate any suggestions for improvement of the package.

WebGL demo: <http://virtualplayground.d2.pl/dimboxes/>

Android demo: <http://virtualplayground.d2.pl/dimboxes/dimBox.apk>

Help and improvement suggestions: virtual@virtualplayground.d2.pl

Package version changes

.....

5.5

- update for Lightweight and Universal render pipelines (LWRP and URP) and for Unity 2019.3.

6.0

- Simplification - combining the base features in a single component. DrawLines.cs script is depreciated since now – the code has got moved to the main component.

- new extended component versions DimBoxProgressive.cs and BoundBoxProgressive.cs with six progressive animation options added

- using MathGeoLib library for bounding box calculations, and using threading in editor

- units/caption setup