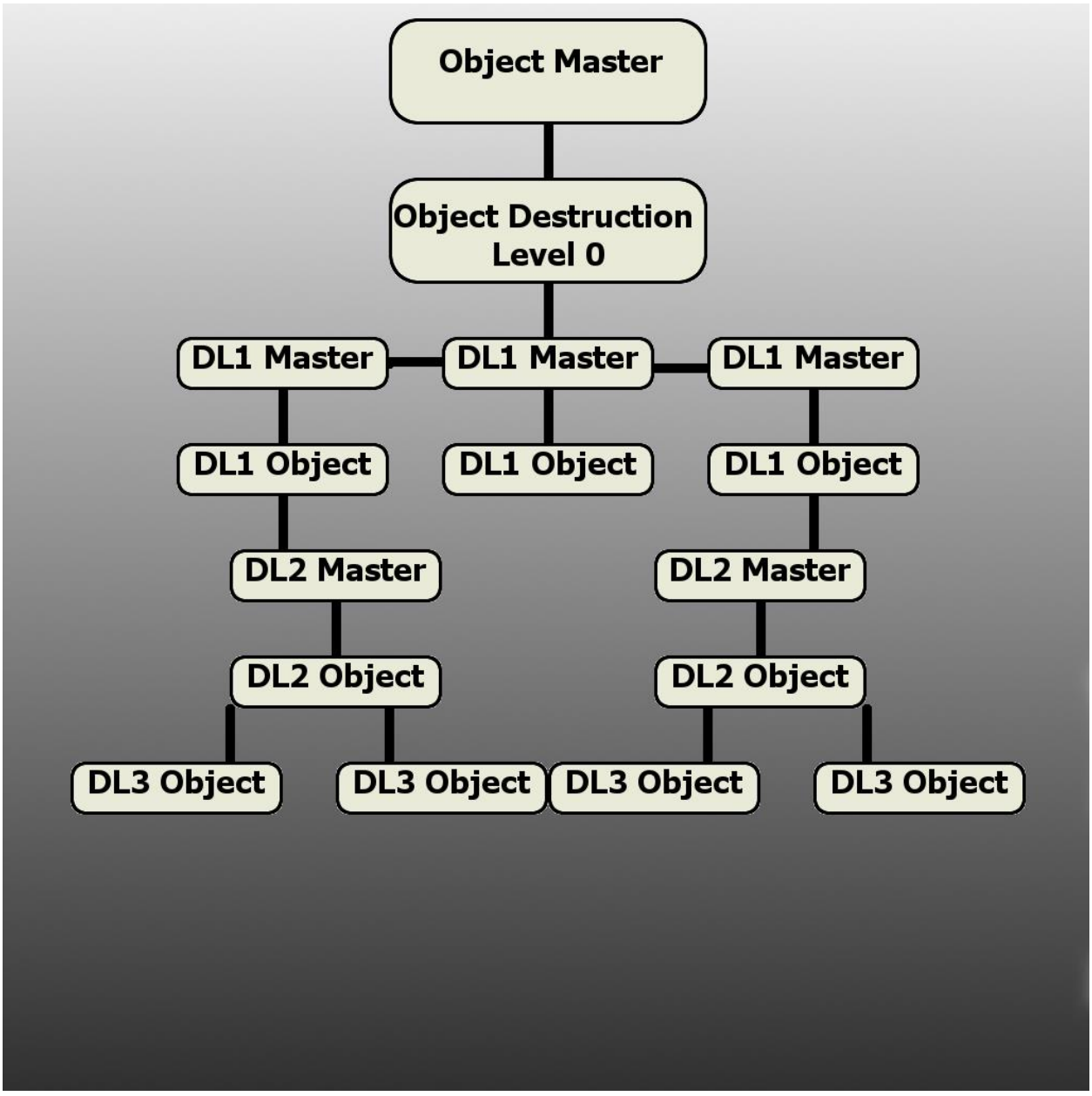
Game Developers Guild Breakable Objects Framework

Instructions

To add a breakable object to your scene you must first set up the fractured mesh in Blender. The exported mesh must follow a strict naming convention so that the editor script can get everything set up for you.



We'll start by setting up a pre-built mesh in Unity, then we'll go over basic instructions on fracturing the meshes in Blender.

Setup in Unity

- First set up the tags "Breakable" and Explosive.
- Open the instructions scene and try it out. (L click shoots balls)
- Open the folder Instuctions in the GDG_Assets directory. In there you will find a cube model.
- Make sure the cube scale factor is 1 in the model tab and that the animation type is set to none under the Rig tab.
- Open the GDG breakable object setup tool in the window tab
- Drag the cube mesh into the imported mesh slot in the setup tool window
- Ensure that write classes, setup DL2, DL3 objects is checked, and can contain cargo is unchecked.
- Set the DL0, DL2, and DL3 colliders to cube and the DL1 collider to mesh. The mesh collider gives a more realistic collision geometry, but gets very expensive once there are a lot of objects in the scene.
- Click the Generate button and the editor will create the required folders, scripts, and prefabs. It will also create one instance of the cube controller and a cube child object in the scene.

Usage in Unity

- Once your cube controller is setup in the scene you can move and scale it at will. DO NOT modify the position or scale of the controller object.
- Check the DL1, DL2, and DL3 enabled boxes in the cube controller and press play to try out the breaking.
- DLX break strength adjusts the amount of force required to break an object.
- You'll notice that the cube tends to fly off the screen. This is due to the mass being 0. You can either set the mass of all the cubes individually, or you can override the mass in the cube controller.
- Duplicate the cube a few times (make sure that they are still children of the cube controller), and change their scale a bit.

- Drag the included dust particle system into the break particle system slot on the controller, enable playDL0particlesystem and change the particle system lifetime to 3. Now all of the cubes will make a little dust cloud when broken.
- Change the nonbreaking tag size to 1 and add Projectile to the first slot. Notice that you cannot break the cube with the projectile anymore. You can however break one cube by hitting it with another. If you want to prevent this, add
  Breakable to the nonbreaking tag list.
- Change the nonbreaking tag list size back to 0
- You'll notice that the chunks tend to blow apart when they are instantiated. This is caused by the cube colliders that we set up earlier intersecting with each other. Enable shrink DL2 and DL3 colliders to instantiate the parts with smaller colliders. They will then grow according to the scale time attribute. Mesh colliders generally don't have this issue, however they are much less efficient than cube colliders.
- If you have not closed the setup tool, you can set up another cube prefab with all mesh colliders to compare performance. Make sure that write scripts is off.
- You can also add several audio clips by changing the break sounds size and adding audio clips. These clips will be played at random, and only on the breaking levels that are checked.

External Breaking

- Now let's break the blocks using the red switches in the scene.
- Select one of the red triggers and drag a cube into the break target slot in the break trigger script.
- There is a dropdown menu that offers 4 different breakage options. The break rotation effect is a bit weak unless you change the max angular velocity setting in the physics panel.
- In the cube controller set the breakthrough level to 3 if you want the switch to break all the way down to the lowest chunks.

Asset Creation

We're going to go an example of breakable object creation and along the way, we'll try to address any problems that you may encounter. For additional help, there are instructional videos on the GDG YouTube page : http://www.youtube.com/user/gamedevelopersguild/videos

Or you can contact us at GameDevelopersGuild@gmail.com

- The fbx file must be named after the object and be one word. So, if you're breaking a cube object, you must name the file cube.fbx.
- The main object DL0 (Destruction Level 0) should be kept on layer 1 and named cube.
- If you run the object through the cell fracture tool, the DL1 objects will be on layer 2 and named cube.cell.00X.
- DL2 objects will be on layer 3 and named cube.cell.00X.cell.00X
- Dl3 objects will be on layer 4 and named cube.cell.00X.cell.00X.cell.00X

Rotation fix and export.

Blender fbx files are imported into Unity with a 90 degree rotation. This is not a problem for most assets, but causes issues when using the breakable framework.

First, select all objects to be exported. Rotate everything on the X axis -90 degrees and apply rotation. Then rotate everything back on the X axis 90 degrees and export via FBX with the default settings.

Into Unity

Drag the fbx into the project as normal, set the scale to 1 and remove animations. Setup is the same as for the cube.

Objects that contain cargo

Import and setup process is the same, except that this time you'll turn on Object can contain cargo when setting up the folders. After everything is built, you need to add at least one cargo object to the egg controller. If you don't, it will throw an error. If you don't want the object to drop cargo, use the included null cargo object. I recommend trying one of the prefabs from the detonator explosion framework (free on the asset store) as cargo.

Advanced Breakable objects

Hollow objects need to have at least the DL1 levels modelled manually. We've included the egg and barrel blend files as examples.

If you have any questions or comments, send us an email at GameDevelopersGuild@Gmail.com.