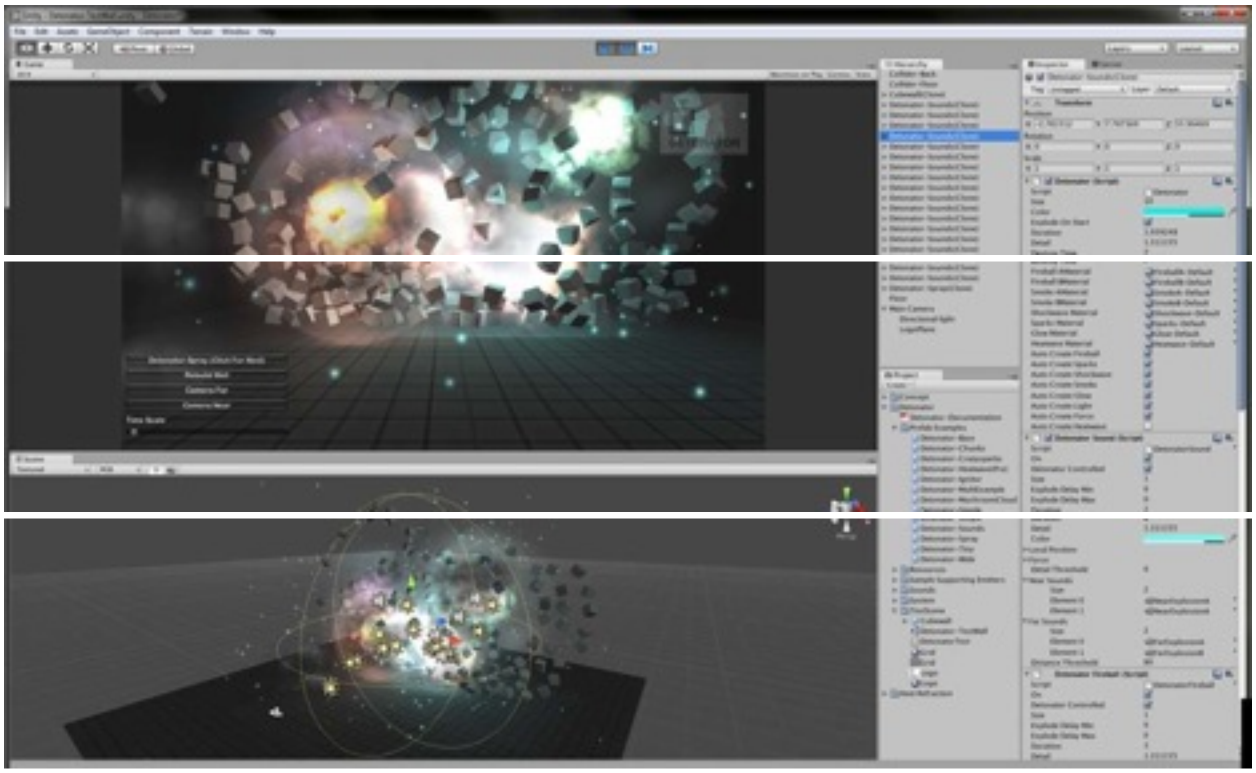
Detonator

Explosions for Unity

Detonator is a system that lets you make good looking explosions quickly and easily. It was originally created in 2009 and as of right (v1.1) now uses the legacy Unity particle system. It can make some very impressive effects but at the expense of many (100+) draw calls per effect. For that reason it is much more appropriate for desktop and console level graphics hardware.



Quick Start

1. Open the scene Detonator-TestWall and flip through the different examples until you find a prefab you like. Click on the wall to create an explosion.
2. Make a copy of the prefab you like and put it in your project somewhere outside of the Detonator folder.
3. In your code, add a hook to that prefab and then instantiate it when you want in your own code. For example, when an object reaches zero health, or when a projectile impacts a gameobject. For help with this, see the Unity documentation here: http://docs.unity3d.com/ScriptReference/Object.Instantiate.html
4. That's pretty much it. The explosion will clean itself up.

What’s Included

An outline of the directory structure underneath /Detonator Explosion Framework

/Prefab Examples

These examples are configured GameObjects with Detonator and one or more Detonator Components applied.

- **Detonator-Insanity** – A demo effect that puts everything into a single explosion.
- **Detonator-Upwards** - Similar to Insanity but with Detonator.UpwardsBias turned up to show how an explosion can easily be made to shoot upwards instead of radially.
- **Detonator-Base**–A straight out of the box Detonator. Same as you’d get by just attaching a Detonator component to any GameObject and letting it blow up.
- **Detonator-Chunks**–Shows the Detonator Spray component emitting some GameObjects that include smoke trails.
- **Detonator-Crazysparks**–Shows a configured DetonatorSparks object that shoots out a ton of sparks in a flattened pattern.
- **Detonator-Heatwave(Pro)** – Demonstrates the heat-distortion effect. Only works in Unity Pro, not Indie.
- **Detonator-Ignitor**–This effect shows a configured DetonatorForce which in addition to causing a RigidBody explosion also ignites any RigidBodies it hits.
- **Detonator-MultiExample**–Shows a single Detonator with three separate DetonatorFireballs, all different colored, positions, and sizes.
- **Detonator-MushroomCloud**–Shows an example of a complex effect - a nuclear blast.
- **Detonator-Simple**–Shows what happens when you replace all of the Detonator materials with a single glowing dot and turn off more intensive components.
- **Detonator-Sounds** – Demonstrates the DetonatorSound component, which lets you have lists of sounds that are randomly chosen to play, and also are different based on whether the explosion is near or far. If you use the Test scene described below, you can switch your camera distance between near and far easily to see the difference between the near and far sound.
- **Detonator-Spray** – Similar to Detonator-Chunks, but emits particle emitters which have RigidBody components.
- **Detonator-Tiny** – Demonstrates the effect Detonator’s size parameter.

- **Detonator-Wide** – 3 Fireballs arranged horizontally for a wider effect.

/Resources

This contains the default textures that Detonator uses when building its materials on the fly. You can replace these or just make new materials pointing to other textures. Since these are in /Resources they'll automatically get included in your build so watch out for that!

/Sounds

Some sample sounds are included. These sounds are in the public domain and were found in a sound pack at: http://www.freesound.org/packsViewSingle.php?id=4366

/Sample Supporting Emitters

These emitters are used by other effects, like DetonatorSpray and DetonatorForce to do various neat things.

/System

This is the actual Detonator code. You can drag components directly from here or use the Component menu.

FAQ

What do all of the different parameters do on a prefab?
They are pretty self explanatory, but really the best way to learn them is to tweak them and see what they do. Easiest way to iterate on an explosion is to put it into the test scene.

How do I put my own explosion prefab into the test scene and iterate on it?
1. Open Detonator-TestWall
2. Select the main camera object
3. In the "Detonator Test" component, drag your explosion prefab to the Current Detonator field.
4. Run the scene. That prefab should be assigned.
5. Click the wall to test it.
6. Tweak the ***prefab*** and re-explode to see the changes. The prefab changes WILL persist in play mode.

Can I use my own materials and textures?

Yes. While a Detonator will build its own default materials, you can easily use your own by just making the Detonator and dragging your materials into the corresponding slots. The 8 slots all cascade down to subcomponents, so you don't need to create them if all you want to do is change materials. The Detonator-Crazysparks prefab demonstrates these material overrides. However, animated textures are not currently supported. It wouldn't be terribly hard, but it's not in the first release.

How do I avoid using the resources folder?

The resources folder is not required as long as you create materials explicitly for your Detonator objects to use and manually assign them to the Detonator component. Resources is set up out of the box so that the code will always be able to populate each sub-piece of the explosion with a valid material.

Can I pool Detonator objects to avoid heavy instantiation?

This is a common question and the short answer is it won't help that much. Most things get instantiated on Explode() and not on the actual instantiation of the Detonator holding GameObject. This is an area worth investigating in future releases.

What's going on under the hood (the very short version)

Detonator is the main component that is the "master" of the explosion. It can generate, and/or control DetonatorComponents. DetonatorComponents all inherit common properties, like size, color, and duration. Detonator is responsible for altering these as needed, so the user can say the Detonator size to 5m and all pieces will follow accordingly.

That's it. Feedback is great! Please visit the Unity forum thread for Detonator to comment.

http://forum.unity3d.com/threads/detonator-explosion-system.259607/#post-1715950