

# Creating a Destructible Object in Unity3D with PlayMaker

by Noel Lopes

## Introduction

This white paper outlines one of many pipelines that can be used to create a destructible asset within the Unity game engine. This particular pipeline is aimed at game artists, and those who have little to no programming experience.

The pipeline involves the following steps:

1. Create asset in Maya,
2. Fracture asset using Maya plugin/script,
3. Save out FBX of "unbroken" and "broken" asset,
4. Import FBX into Unity, create Prefabs, setup Unity scene,
5. Create destruction particle effect using Shuriken in Unity, and fractured assets,
6. Create trigger geometry for destructible object,
7. Create FSM using PlayMaker for Unity.

The pipeline utilises the following programs and plugins for the creation of the destructible asset:

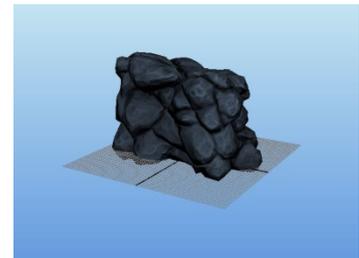
- Maya - asset creation,
- Ninja Fracture (Ninja Dojo) - fracturing of asset,
- Unity Game Engine - turn assets into a game objects for use in engine,
- Shuriken Particle System (Unity)- create destruction effect using fractured assets,
- PlayMaker (Unity) - create finite state machine (FSM) for destructible game object in engine.

NB: Some basic knowledge of the Maya, and Unity is assumed for this pipeline.

## Unity3D Destructible Object Pipeline

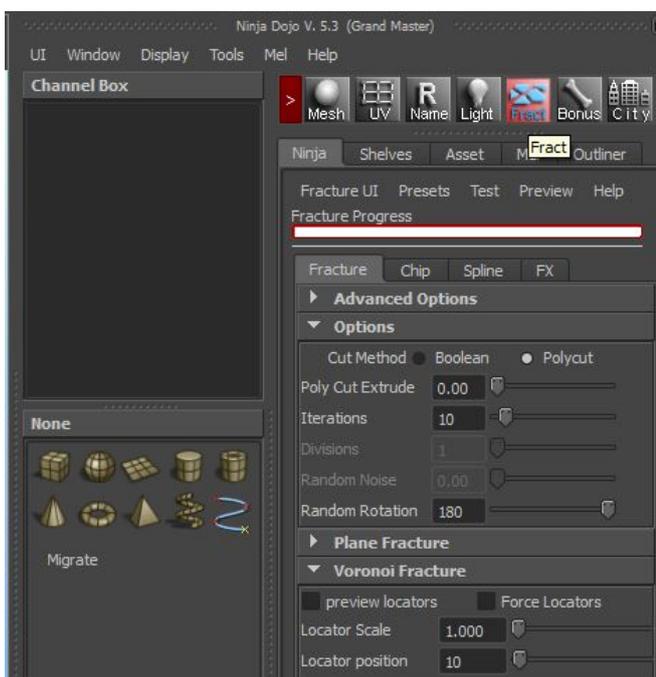
### 1. Asset Creation

It is assumed that the reader of this white paper has sufficient modelling knowledge in Maya to create an asset. I will be using a rock wall which I created in Maya, (and textured with Photoshop), as an example as I run through the pipeline. .



### 2. Fracture Asset

The asset must be pre-fractured, as this is currently still the optimal method used for destroying game objects. Dynamic fracturing in games can be very resource "expensive".



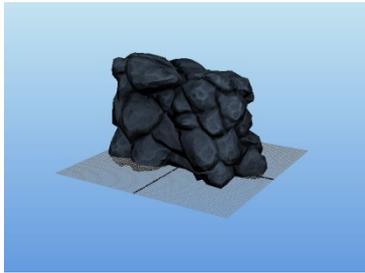
### 2a. Fracturing the asset using Ninja Fracture:

- Once the asset has been modelled, (and possibly textured too), it can be fractured.
- Run the Ninja Dojo script.
- Select Ninja Fracture menu set.
- In the Options box:
  - Set Iterations to ten (10), see diagram.
- In Voronoi Fracture box:
  - Set Locator position to ten (10), see diagram.
- Make sure asset is selected, and hit Voronoi Fracture!

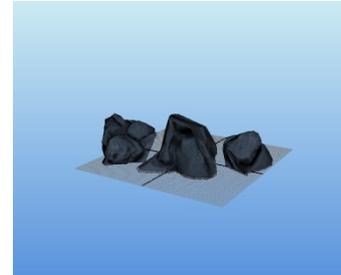


### 3. Save Assets as FBX

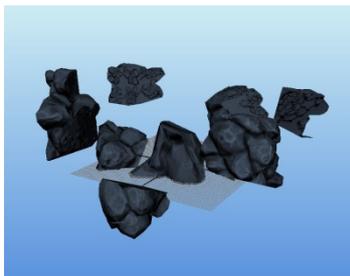
Once fractured, save out both unfractured and fractured assets in the FBX format, for use in Unity. When saving out the fractured assets, save out a fractured base, that represents the remains after destruction, as well as some of the individual fractured assets.



<Unbroken Rockwall



Broken Rockwall Base>



<Rockwall Fragments

### 4. Create Prefabs in Unity and setup Scene

Once FBX assets (and textures) have been imported into Unity, drag the FBX assets into a scene, and attach any materials required. Then drag these scene assets back into the project folder. If the scene item turns blue in the hierarchy, then a prefab has been created. Ensure that the unbroken and broken assets have the same origin.

Other elements that should be added to the scene for the purposes of this tutorial include:

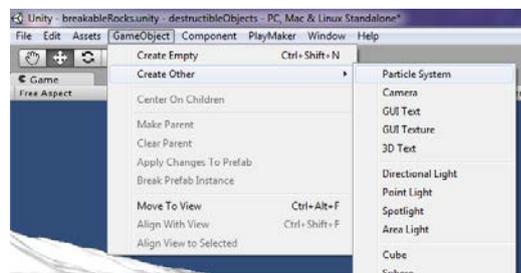
- Terrain/Ground Plane
- First Person Controller
- Light (Directional)

### 5. Create Destruction Particle Effect

To create a particle effect for the destruction of the asset, we will use the Unity Shuriken Particle System. The effect will actually comprise of two particle systems. One for the fractured pieces of the asset, the other for a puff of dust.

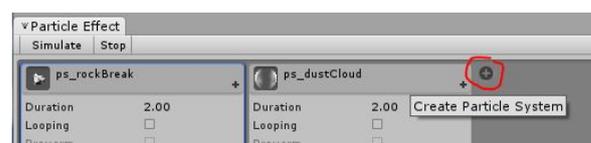
To create a Shuriken particle effect:

- i. Go to GameObject > Create Other > Particle System
- ii. Name the particle system "ps\_rockBreak"

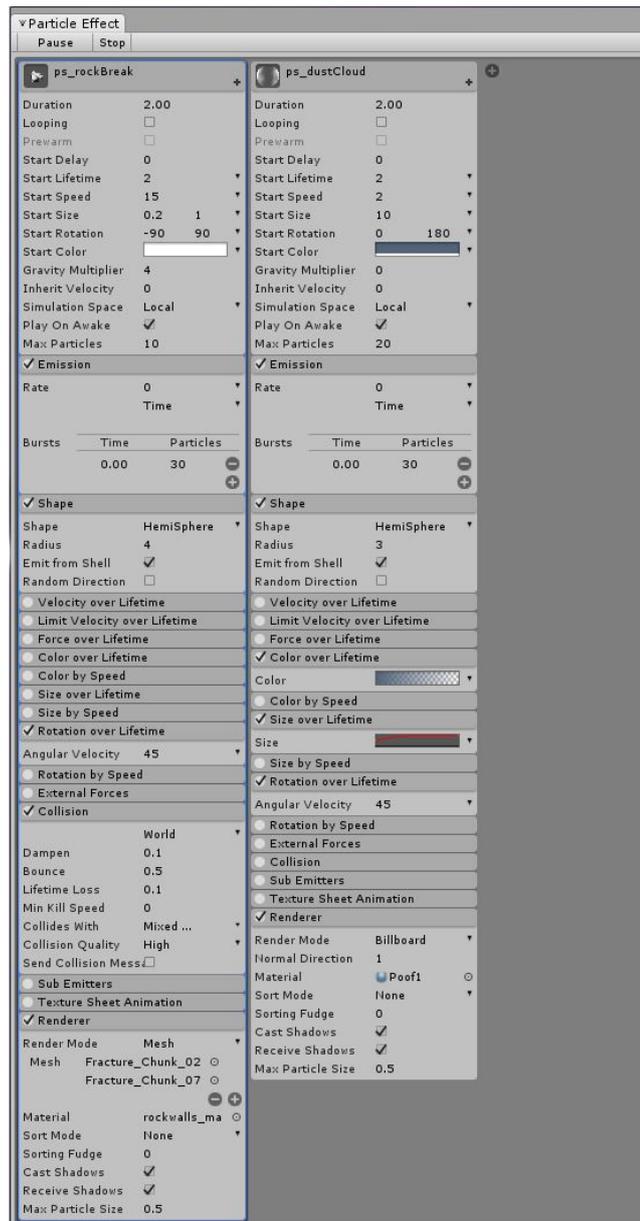


- iii. To add another system to the existing one:

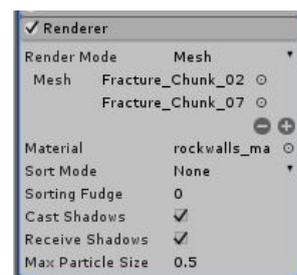
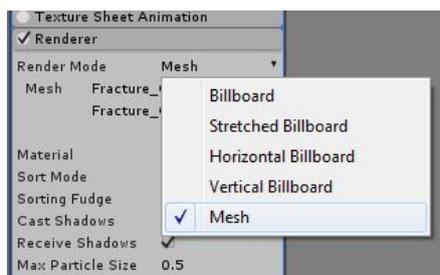
- Go to "Open Editor...", and
- Click "+" to create a new system



- iv. Fill in the fields for the particle systems, as shown in the diagram, to create the destruction effect.



- v. Shuriken allows billboard images to be replaced by meshes in Render > Render Mode:
- Change Render Mode to Mesh, and
  - Add the individual fractured pieces to the system,
  - Note: A maximum of four (4) meshes can be added to a system.



- vi. Once completed, add the effect to the project folder to create a prefab.  
vii. Delete the particle effect from the scene.

## 6. Create Trigger Geometry for Destructible Object

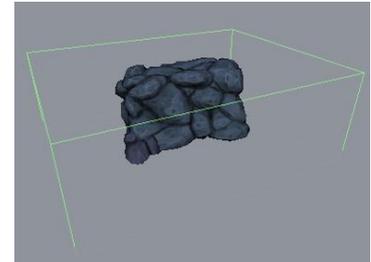
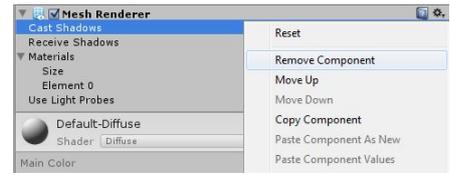
We will make the object destroy itself when the player approaches the object. This requires a trigger to be made:

To create the geometry for the trigger:

- Create a cube,
- Rename the cube "t\_rockwall\_Trigger",
- Remove the Mesh Renderer component,
- In Box Collider component, check "Is Trigger",



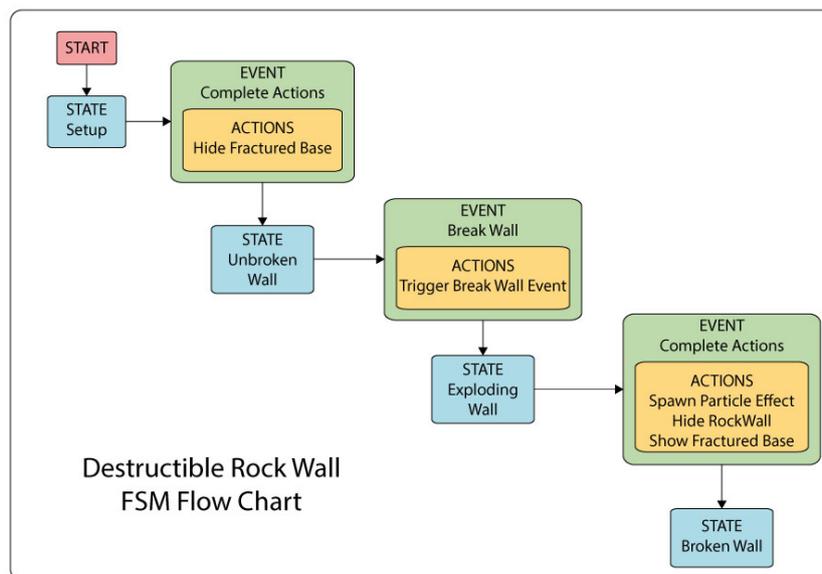
- Scale/move the trigger box to encapsulate the rockwall. (The trigger can be quite large).



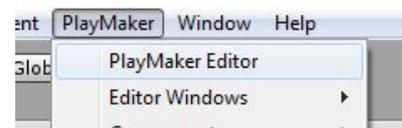
An FSM can now be added to the trigger geo to create the destructible rockwall.

## 7. Create the PlayMaker FSM

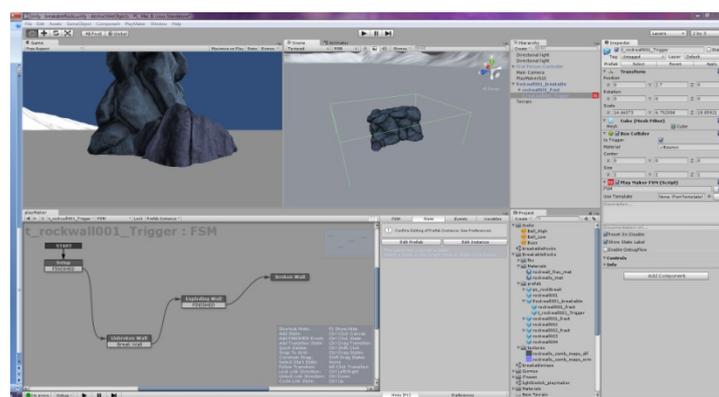
With all the assets loaded as prefabs into Unity, and the trigger geo created, we can now create the FSM that will allow us to make the asset destructible. The following diagram is a simple visual representation of the FSM we will be making:



The FSM will be attached to the trigger geo we created earlier. Ensure the PlayMaker package has been imported into the project, and open the PlayMaker Editor.

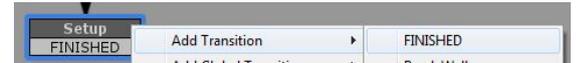
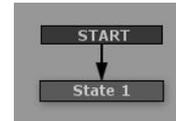


An example of how you might want to layout your Unity windows is shown below.

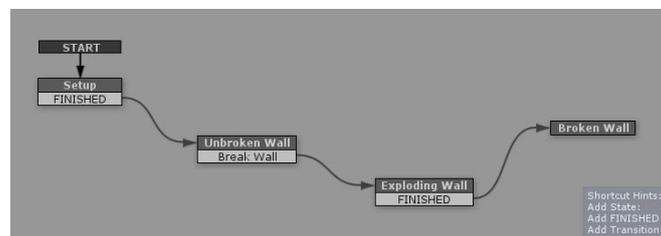


To create the FSM using PlayMaker:

- Select "t\_rockwall\_Trigger",
- In PlayMaker Editor window; right-click to add FSM,
- In State tab, rename "State 1" to "Setup",
- Right-click Setup, and go to Add Transition > FINISHED, to add "FINISHED" event to state,
- Right-click anywhere in editor window and Add State,
- Call new state "Unbroken Wall",
- Go to Events tab, and add "Break Wall" event,
- Right-click "Unbroken Wall" state, and add "Break Wall" event,
- Right-click anywhere in editor window and Add State,
- Call new state "Exploding Wall",
- Right-click "Exploding Wall" state, and add "FINISHED" event,
- Right-click anywhere in editor window and Add State,
- Call new state "Broken Wall",
- Connect "Setup" > "Unbroken Wall" > "Exploding Wall" > "Broken Wall"



The FSM should now look like this:



Now that the states (and events) have been added to the FSM, we can add the actions to them:

"Setup" Actions:

- Select "Setup",
- In the State tab, go to the Action Browser and add "Set Visibility",
- Follow settings in diagram to set visibility of fractured object,
- 



Note: "Set Visibility" action will be required for each piece of fractured geo in broken base.

"Unbroken Wall" Actions

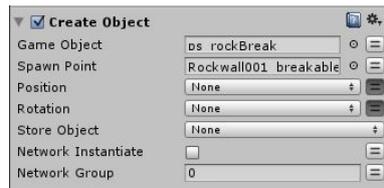
- Select "Unbroken Wall",
- In the State tab, go to the Action Browser and add "Trigger Event",
- Set trigger to "On Trigger Enter",
- Send Event "Break Wall", (created earlier),



"Exploding Wall: Actions

- Select "Exploding Wall",
- In the State tab, go to the Action Browser and add "Create Object",
- Drag the "ps\_rockBreak" prefab, (from the project folder), into Game Object,

- Drag the object that is to be the spawn point of the particle effect into Spawn Point, see diagram, (in this case Rockwall001\_breakable)



- Go to the Action Browser and add "Set Visibility",
- Hide unbroken rockwall, see diagram



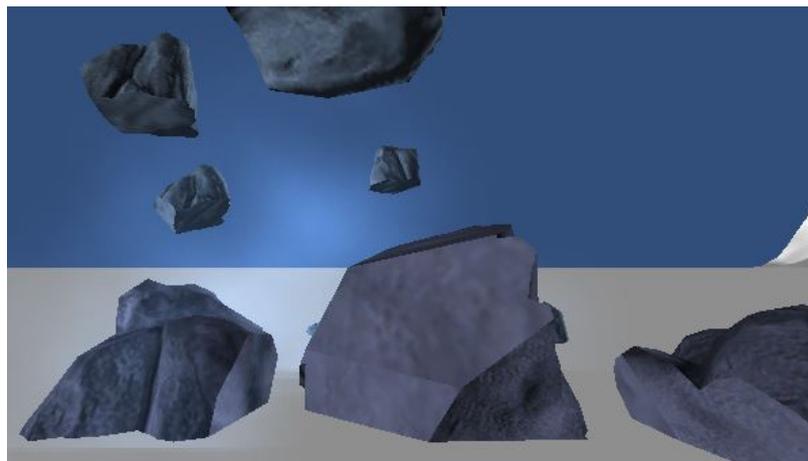
- Go to the Action Browser and add "Set Visibility",
- Show fractured rockwall pieces, see diagram



Note: Create "Set Visibility" actions for all fractured pieces, as required.

NB: The "Broken Wall" state does not require any actions, as nothing happens in this state. The rockwall has already been destroyed!

To test the FSM, play the Unity project in the Game window. Move towards the rockwall, and experience the destruction! =P



**Noel Lopes**

Email me at [noel.lopes@gmail.com](mailto:noel.lopes@gmail.com)

Visit my website, and blog, at [noellopes.com](http://noellopes.com)!