

Shatter Toolkit

Copyright 2015 Gustav Olsson

<http://gustavolsson.squarespace.com/>

How it works

Attach one ShatterTool script, located in Core/, to a game object that you want to shatter or split. If the mesh of the game object has vertex uv-coordinates or tangents, also attach a UvMapper script (WorldUvMapper or TargetUvMapper). If the mesh has vertex colors, attach a ColorMapper script (SolidColorMapper). When shattering or splitting a game object the pieces will be instantiated as clones of the original game object and the original game object will be destroyed.

The ShatterTool script requires that the game object has a MeshFilter attached, as it needs some geometry to work with. Any other component you attach will be carried over without modification to the pieces when the game object is shattered or split. However, the ShatterTool script takes special care to itself, MeshCollider and Rigidbody components. If attached, any of these components will be updated to act realistically when the game object is shattered or split, by for example modifying a Rigidbody's mass and velocity.

> IMPORTANT: If you attach a MeshCollider and a Rigidbody, always keep the MeshCollider Convex in order to avoid errors while updating the mass properties of the Rigidbody.

> IMPORTANT: If ShatterTool.FillCut is enabled (default) every edge of the mesh needs to belong to exactly two triangles, ie. the mesh needs to be closed.

Here are a number of ways you can shatter or split a game object:

No scripting required:

- Attach the ShatterOnCollision script (located in Helpers/Game Objects) and specify the required velocity needed to shatter the game object (requires an attached rigidbody and convex collider)
- Attach any of the mouse scripts (located in Helpers/Mouse) to an empty game object in the same scene

Scripting:

- In a script, send a "Shatter" message to the game object and specify a world-space point (Vector3) where the game object should be shattered; for example

```
SendMessage("Shatter", Vector3.zero);
```

- In a script, send a "Split" message to the game object and specify an array of world-space planes (Plane[]) with unit-length normals where the game object should be split; for example

```
SendMessage("Split", new Plane[] { new Plane(Vector3.right, Vector3.zero) });
```

Example scenes:

Check out the example scenes to see how the shatter toolkit is used in practice.

Good to know

- * The ShatterTool properties have tooltips in the editor, mouse over to read them.
- * You can make the ShatterTool instance send pre- and post-split messages by toggling the "Pre Split msg" and the "Post Split msg" properties in the editor. These may be useful if you need to do something before and/or after a split occurs.
- * When shattering or splitting parented game objects, make sure that you always handle the children/parent carefully to avoid duplicating too many game objects. This can be done with the bundled HierarchyHandler helper script, see the Hierarchy example scene for more information.