Introdution

The problem – You have spent many hours perfecting your well designed and beautiful levels. But, everything looks to clean. To tidy and sterile. In the real world, many areas are unfortunately littered with trash and rubbish. In order for your levels to feel truly immersive, the streets of your post apocolyptic town need trash.

Mess Maker is the solution. It provides a system to generate piles of trash antwhere you chose, randomly creating them on the fly from a poole of selected objects and renders them to limited draw distance of your choice. The trash generated can even have physics, scripting or any components you chose. It is compatible with all render pipelines and any shaders of your choice can be used.

Mess Makers is supplied with 41 different trash prefabs in 2 levels of detail. You can easily replace these or add your own. And why limit it to just trash? A selection of 5 different kinds of rock is supplied in 3 levels of detail. You can use your own, though the rocks supplied are designed to blend well with Unity's standard free terrain textures.

Included within the trash is a selection of bottles and cans. To avoid any copyright infringements, innocuous and uncopyrighted words have been used in place of brand names with fairly simplified labels. Don't like our fictional brands? Make your own! Blank templates are provided with instructions to create any soda cans, beer cans, beer bottles, wine bottles or spirit bottles you want. You could even make these into pick up items unique to your game.

List of Assets

Cans:

- **Beer Can Crushed - with versions in blue, green and red colour schemes – textures in 1k resolution**
  Low poly: 1,246 vertices
  High poly: 4,616 vertices
- **Beer Can - with versions in blue, green and red colour schemes – textures in 1k resolution**
  Low poly: 1,210 vertices
  High poly: 4,612 vertices
- **Soda Can Crushed - with versions in orange, green and red colour schemes – textures in 1k resolution**
  Low poly: 1,207 vertices
  High poly: 4,627 vertices
- **Soda Can - with versions in orange, green and red colour schemes – textures in 1k resolution**
  Low poly: 1,206 vertices
  High poly: 4,627 vertices

Food:

- **Apple Core - with versions in green and red colour schemes – textures in 512p resolution**
  Low poly: 1,053 vertices
  High poly: 3,783 vertices
- **Chicken Bone - textures in 512p resolution**
  Low poly: 154 vertices
  High poly: 573 vertices
- **Fish Bone - textures in 512p resolution**
  Low poly: 4,150 vertices
  High poly: 15,083 vertices
- **Fish Bone 2 - textures in 512p resolution**
  Low poly: 2,518 vertices

High poly: 9,228 vertices

- **Half Eaten Cheesburger - textures in1k resolution**
  Low poly: 684 vertices
  High poly: 2,759 vertices

<u>Glass Bottles:</u>

- **Beer Bottle with Label - with versions in blue, green and red colour schemes – textures in 1k resolution**
  Low poly: 1,820 vertices
  High poly: 6,548 vertices
- **Beer Bottle without label - with versions in green and red glass – no textures required**
  Low poly: 1,423 vertices
  High poly: 5,181 vertices
- **Vodka Bottle with Label – textures in 1k resolution**
  Low poly: 1,415 vertices
  High poly: 4,971 vertices
- **Vodka Bottle without label – no textures required**
  Low poly: 1,391 vertices
  High poly: 4,947 vertices
- **Wine Bottle Broken - with versions in green and red glass – no textures required**
  Low poly: 519 vertices
  High poly: 1,829 vertices
- **Wine Bottle Broken 2 - with versions in green and red glass – no textures required**
  Low poly: 889 vertices
  High poly: 3,025 vertices
- **Wine Bottle without label - with versions in green and red glass – no textures required**
  Low poly: 963 vertices
  High poly: 3244 vertices
- **Wine Bottle with Label - with versions in green and red colour schemes – textures in 1k resolution**
  Low poly: 1,359 vertices

High poly: 4,682 vertices

Porcelin and Pottery:

- **Broken Dinner Plater – no textures required**
  Low poly: 270 vertices
  High poly: 1,082 vertices
- **Broken Pottery – textures in 1k resolution**
  Low poly: 360 vertices
  High poly: 1,360 vertices
- **Broken Pottery 2 – textures in 1k resolution**
  Low poly: 290 vertices
  High poly: 1,145 vertices
- **Coffee Mug White Broken – no textures required**
  Low poly: 431 vertices
  High poly: 1,590 vertices
- **Coffe Mug White – no textures required**
  Low poly: 511 vertices
  High poly: 1,880 vertices
- **Wood Mug – textures in 1k resolution**
  Low poly: 511 vertices
  High poly: 1,880 vertices
- **Pint Glass – no textures required**
  Low poly: 593 vertices
  High poly: 2,208 vertices
- **Wine Glass – no textures required**
  Low poly: 607 vertices
  High poly: 2,085 vertices

Rocks:

All Rocks use a shared material with a tiled texture in 512p resolution

- **Rock 1**
  Low poly: 127 vertices
  Medium poly: 437 vertices
  High poly: 1,645 vertices

- **Rock 2**
  Low poly: 123 vertices
  Medium poly: 441 vertices
  High poly: 1,691 vertices
- **Rock 3**
  Low poly: 121 vertices
  Medium poly: 441 vertices
  High poly: 1,678 vertices
- **Rock 4**
  Low poly: 125 vertices
  Medium poly: 440 vertices
  High poly: 1,668 vertices
- **Rock 5**
  Low poly: 125 vertices
  Medium poly: 455 vertices
  High poly: 1,829 vertices

Getting Started

Mess Maker scan be set up to work very quickly. All materials are supplied with SRP Standard Shaders. If you are using URP or HDRP, you will need to convert them using the standard render pipeline upgrade options. If you wish to use your own custom shaders, there is no issue with this.

Mess Maker comes with 2 demo scenes. The first simply shows all artwork assets for your perusal. The second is a small scene set up to deomonstrate how Mess Maker works. If you open the scene, you will find that a simplified city like environment has been constructed using primitive shapes.

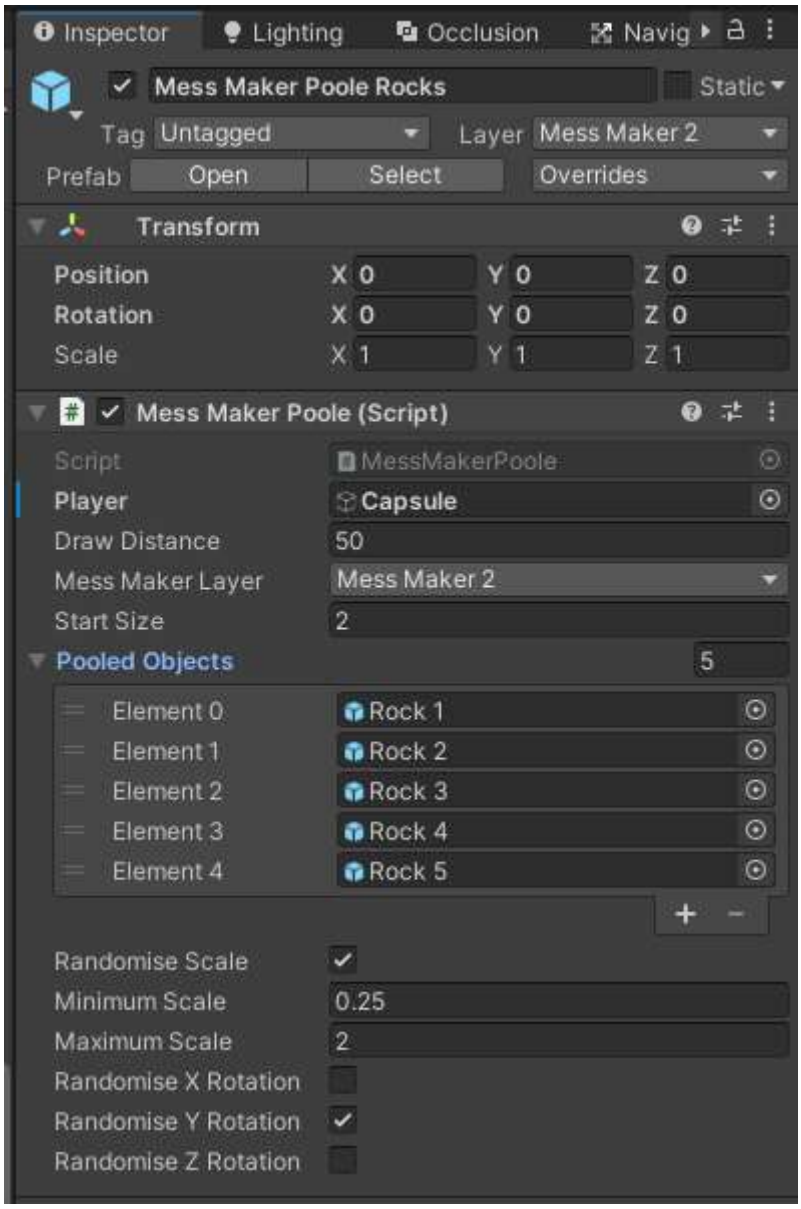
You will see 2 prefabs have been added to the scene. “Mess Maker Poole Low Poly” and “Mess Maker Poole Rocks”. In the prefabs folder, you will see that there are 5 non artwork related prefabs for you to use.

There are 4 prefabricated versions of the Mess Maker Poole supplied, although you may duplicate them or edit their settings to suit your needs.

Mesh Maker works by building a poole of objects of your choice, and creates piles of rubbish as defined by you where as the player moves around the scene. The Mesh Maker Poole is first added to the scene. You can add multiples of these with different pooles with no issue.

The Mess Maker Prefabs

Let's take a look at the Mess Maker Poole Rocks prefab and it's settings:



The script attached takes 11 public variables to set the prefab poole.

1) Player – This is the object that you define as your player. Generally, it will have a camera attached to it. In the case of the demo scene, it is a capsule with a simplified FPS controller attached to allow you to navigate the scene. You will to set this in your own scenes, and the oject will need to have a Rigid Body or Character Controller component attached.

2) Draw Distance – Once spawned, object further away from the camera than this specified distance will be culled.

3) Mess Maker Layer – It is best if you create a new layer for Mess Maker. But you may use another layer as if you please. The script will set all objects on this layer to be culled after the draw distance, and all objects spawned from this poole will be placed on this layer. If you wish to use different draw distances for different pooles, you will need to use different layers. If this layer is not set once you have imported the package, you must ensure it is set in both prefabs before pressing play.

4) Start Size – As a general rule, this can simply be left at 2. This is number of each of the types of objects which be created and stored within the poole. If more than this are required, more will be instantiated. If you will be creating areas which are particularly dense, you may want to increase this to avoid any instantiation during the running of the level. But it is not generally necessary.

5) Pooled Objects – This is a list of all objects that will be used by any Mess Makers that use this poole. You may add any game object you desire to this list without limitations.

6) Randomise Scale – Some object types, like the bottles and cans, should be kept at a consistent scale. However, it will increase the feeling of randomisation to randomise the scale of some object type, like rocks. Rocks are never not a uniform size.

7) Minimum Scale – if you not selected to randomise scale, ignore this field. If you have, it is the smallest scale that the objects taken from this poole will be spawned at.

8) Maximum Scale – if you not selected to randomise scale, ignore this field. If you have, it is the largest scale that the objects taken from this poole will be spawned at.

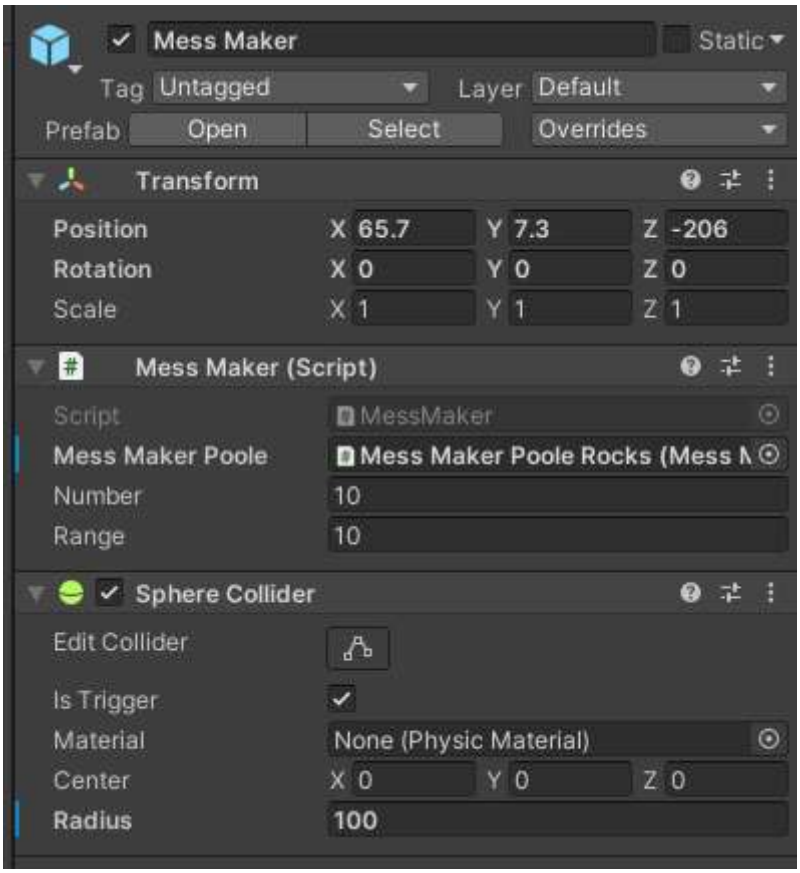
9) Randomise X Rotation - The X Rotation of any objects taken from this poole will be randomised.

10) Randomise Y Rotation - The Y Rotation of any objects taken from this poole will be randomised.

11) Randomise Z Rotation - The Z Rotation of any objects taken from this poole will be randomised.

After this has been added to your scene, you will want to put piles of trash all over the place. Now you can go wild. Drag a Mess Maker prefab into your scene. For organisation purposes, the Mess Makers have been placed in the scene heirarchy as children of the Poole they are taking from. This is not required, and is done purely for simplicity of organisation.

Let's take look at the scripts and settings of the prefab:



The Mess Maker script takes 3 public variables.

1) Mess Maker Poole – This is the Poole we just looked through. You can drag and drop the Poole prefab into this slot, and the script will be selected. This will define which poole this Mess Maker will spawn.

2) Number – The number of objects that will be spawned in this pile.

3) Range – The distance from the centre of this object that the objects will be randomly spread across. The objects are spread across the 2D plain only, so if you have placed the object on an incline, be careful as some objects may spawn beneath it. In the demo scene, the Mess Makers have been placed slightly above the ground to allow the rubbish to fall naturally.

The prefab also has a sphere collider trigger attached to it. When the player object enters the trigger, it will activate the mess pile. When they exit the trigger, the pile will deactivate and that pile will no longer exist. Every time the player enters, a new pile is generated randomly.

Now that you understand how the scene was created, you press play and navigate around the world to see how it all looks in action.

Templates

As mentioned in the introduction, the bottles and cans provided have been designed with generic branding in order to avoid any copyright infringements.

You can use the templates provided to create your own designs.

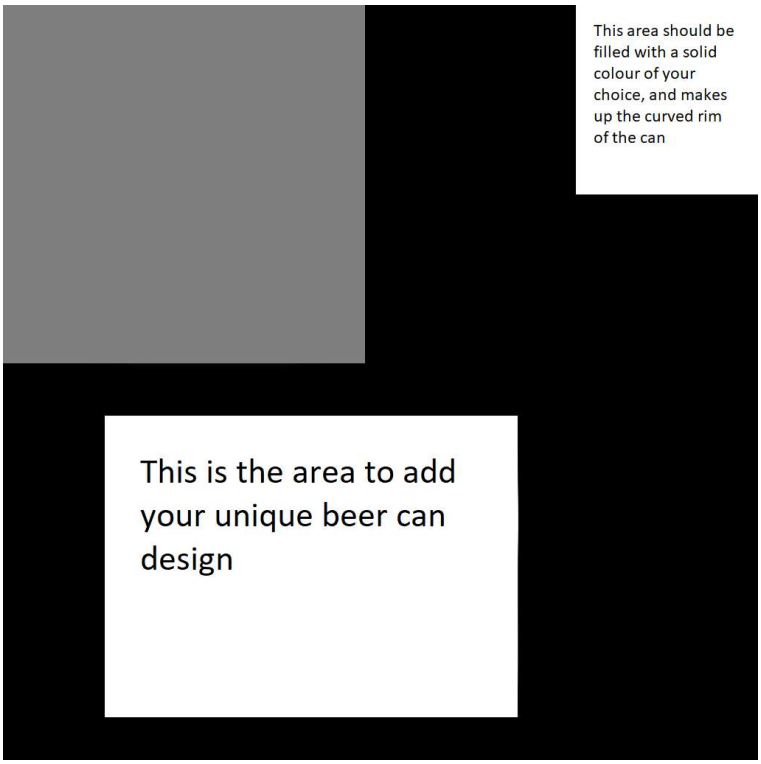
Blank versions of the following have been provided in the folder called “Templates” in both low and high poly:

1. Beer Bottle Green
2. Beer Bottle Red
3. Beer Can Crushed
4. Beer Can
5. Soda Can
6. Soda Can Crushed
7. Vodka Bottle
8. Wine Bottle Green
9. Wine Bottle Red

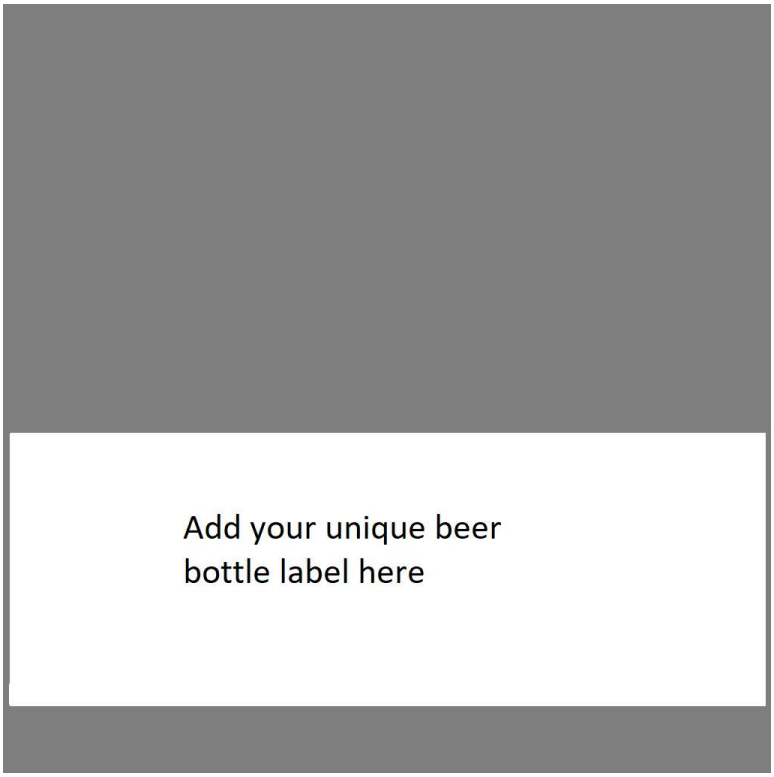
Each is set with a blank version of the relevant material. To edit the materials, take a look at the textures these assets use. In the case of the glass bottles, the labels are separate ojects within the children of the prefabs.

Each material has a texture composed of rectangular shapes, it’s texture atlas, with areas you can add you own design. You can duplicate these and create as many unique designs as you desire. Be sure to only edit the white areas of the texture, as they have been UV mapped onto the meshes provided.

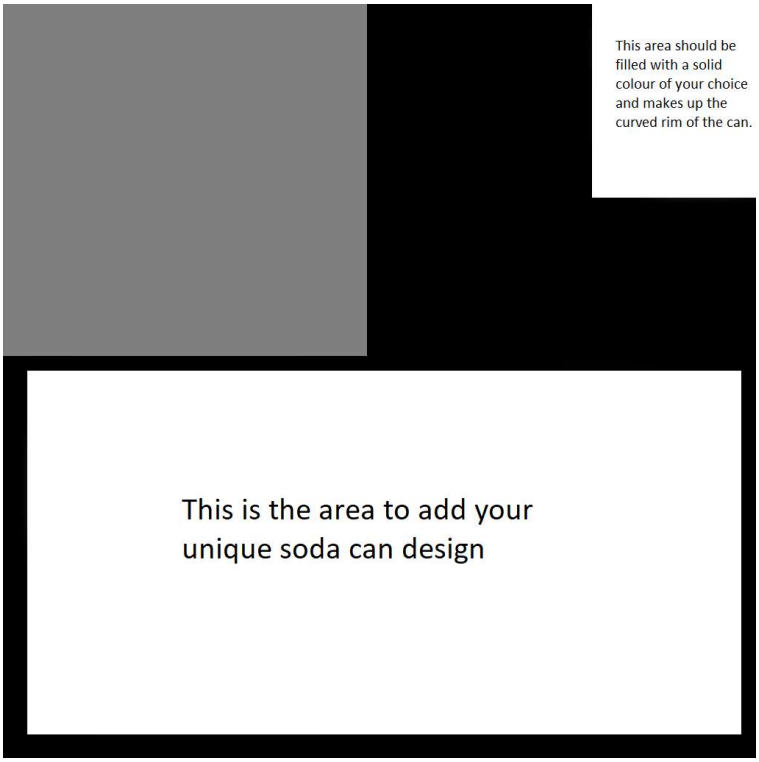
Beer Can Blank:



Beer Bottle Label:



Soda Can Blank:



Vodka Bottle Blank:

Wine Bottle Blank:

Add you unique wine bottle label here. Or any other drink you can think of that fits these bottles.

Well, that's it for the documentation. I think that's everything covered. But if I've missed anything, or you have any other questions, you can get in touch with me via e-mail at quantumnexusgames@gmail.com.

I hope you find this package useful.