|  |  |
| --- | --- |
| **LootMaker**  **Random Loot Generator**  Created by [Ripcord Development](http://www.ripcorddev.com/)  [info@ripcorddev.com](mailto:info@ripcorddev.com) | **icon.png** |

|  |
| --- |
| **This project has been thoroughly tested for bugs before being sent to the Unity Asset store. In the event that you do find an issue with this package, please contact us before posting negative feedback on the Unity Asset Store. We are more than willing to help solve any issues you may encounter.** |

LootMaker is a system for generating random loot objects in your game. The bulk of the work is contained in one function so it's easy to integrate into any project. The loot prefabs have their own scripts attached which will handle the rest of the work.

**PREFABS**

The LootMaker package includes a large number of models and textures for objects representing common types of loot found in games. These prefabs are sorted into two different categories, 2D and 3D. The 2D icons are sprites (a single polygon that always faces the camera and a texture). The 3D prefabs are full 3D models that can be viewed from any direction.

* **3D** - 3D models representing many common categories of loot objects found in games today.
* **2D** - 2D sprite versions of all the 3D models.
* **Particles** - Particle effects designed for each type of loot. These are generated when the loot object is collected.
* **Text Meshes** - Text meshes that display the value of the collected loot object.

**SCRIPTS**

LootMaker consists of a couple main scripts and a few support scripts that help enhance the look of the loot objects in the world.

**LootManager**

This script contains the variables that store the total amount of each type of loot collected. If you want to add more types of loot, or change the types that can be collected simply add or remove options from this script. There should only be one instance of this script in your scene.

**LootGenerator**

This script holds all of the loot prefabs you want generated. When the GenerateLoot function is called it will select one random prefab from the array and instance it into the scene. This script is currently set up to trigger on a mouse click, but it can easily be triggered under any circumstance you wish. It also has an option to add a force to the generated object when it spawns into the scene. This would be useful for having your loot object burst out of a box or enemy, etc.

**LootObject**

This script must be placed on any collectable loot object. It generates the numerical value of the object as well as generates any effects that may be attached to it.

* **LootType** – Select the type of loot you want the object to modify. The LootTypes are stored in the LootManager.
* **LootValueMin** – This is the minimum value the loot object can have.
* **LootValueMax** – This is the maximum value the loot object can have.

**Tip***: The value of the loot object will be randomly selected from anywhere between the min and max values. If you want the loot object to have a specific value just make the MAX and MIN values the same.*

* **LootHighlight**
  + **ShowHighlight** – If true, generate an effect to make the loot object more visible in the world.
  + **HighlightEffect** – An effect prefab that makes the loot object more visible in the scene.

**Tip:** *You could skip this feature entirely if you want to build an effect into the loot prefab itself.*

* **LootAudio**
  + **PlayAudio** – If true, audio clips will be played in the below instances. If value does not have an audio clip, it will be ignored.
  + **SpawnAudio** – Play an audio clip as soon as the object enters the scene.
  + **IdleAudio** – If true, play an audio clip while the loot object is in the scene. This clip will stop playing once the loot object is collected.
  + **CollectAudio** – An audio file placed here will be triggered when the loot object is collected.
* **LootEffects**
  + **Show Effects** – If true, an effect will be generated when the loot object is collected.
  + **LootCollectEffect** – Any prefab (ideally a particle effect) placed here will be instanced into the scene when the loot object is collected.
  + **DisplayValue** – If true, generate a TextMesh object that displays the value of the loot object collected.
  + **ValueDisplay** – The TextMesh object that will display the value of the loot object collected.
  + **ValuePrefix** – Any string entered will appear in front of the loot valueDisplay.
  + **ValueSuffix** – Any string entered will appear behind the loot valueDisplay.

**LookAtCamera**

This script is attached to all the sprite prefabs. Any object it is attached to will face the camera when it is spawned.

* **AlwaysLookAt** - If you want the object to always face the camera wherever it moves, make sure this option is enabled.
  + **Tip***: This feature is handy for 2D sprites in a 3D world.*

**SpinAndBounce**

This simple script modifies the movement of the loot object.

* **Spin Rate -** The speed at which the object spins. Negative numbers will make the object spin in the reverse direction.
* **Bounce Amount** - Based on a sine wave, the bounce amount controls how far the object will hover above or below its start point.
* **Bounce Speed** - The speed at which the object travels from the max and min points on the sine wave.

**TextBehaviour**

This script is attached to all the text mesh prefabs. It controls the colour and movement of the text mesh.

* **Text Colour -** The colour of the text mesh when it is generated.
* **Target Transform** - When the text mesh is generated its position and rotation will update until it matches the position and rotation of the target.
  + **\*Tip***: The target object in each of the text mesh prefabs is originally parented to the text mesh but when activated the parent/child link is broken. This allows the target to spawn relative to the text mesh but not move with it once the motion starts.*
* **Move Speed** - The speed at which the text mesh will move from its start point to the position of the target object
* **Rotate Speed** - The speed at which the text mesh will rotate from its start point to the rotation of the target object
* **Fade Out** - If active the text will fade over time until it is invisible.
* **Time Visible -** How long the text mesh is visible before it disappears.
* **Fade Speed** - The speed at which the text mesh fades out. The object will be removed from the scene once it fades out completely.

**DEMO\_ButtonBehaviour**

Attach this script to any object you want to act as a button. It controls the on, off, and click states of the button.

* **OverState** - When the cursor is hovering over the object, the object will change to the specified colour.
* **ClickState** - When the cursor clicks on the object, the object will change to the specified colour.
* **Fade Time\*** - The amount of time it takes for the colour to fade from one state to the next.
  + **\*Tip***: For instant colour change simply set the fade time to zero.*

**DEMO\_LootGenerator**

This script is an expanded version of the LootGenerator script. It was created to easily demonstrate the functionality and flexibility of lootMaker. The purpose of this script is to allow the user to view various combinations of loot objects and generate them over and over again without having to reload the scene. This script is thoroughly commented

Extra features include:

* Ability to dynamically add contents of multiple arrays into one main array that the generator will pull from.
* Instance a random loot object at any scene object named "mount".
* Remove all existing loot objects from the scene before generating new ones.

**DEMO\_OptionButton**

This script is mainly for demo purposes only. It controls the behaviour of the option buttons as well as handles the actual toggle of the option itself.

* **OverState, Click State, FadeTime** - Please see buttonBehaviour script above.
* **Active At Start Up** - If selected this will set the option button to "On" at start up, rather than the default "Off".
* **Target** - This is the object that contains the DEMO\_lootGenerator script. This tells the script where the options are that need to be toggled.
* **ObjectList** - These options correspond to the ones listed in the DEMO\_lootGenerator. The selected option is the one the button will toggle.