RPG Extension – Ground Target

Setup and Instructions

What is It

Ground Target is an extension for the GameGrind Simple RPG tutorial that allows you to add ground targeting effects like in the scene above. The system works by projecting a horizontal particle system onto any item that is tagged as part of the ground layer. When the player clicks the desired spot, the system then spawns the provided gameobject.

Why this Approach

The design approach was for the ultimate flexibility. Using a particle system not only avoids the overhead of a projector, but also allows designers to change the way the ground target looks during gameplay. Any particle system property (color over time, size over time etc) works with the system.

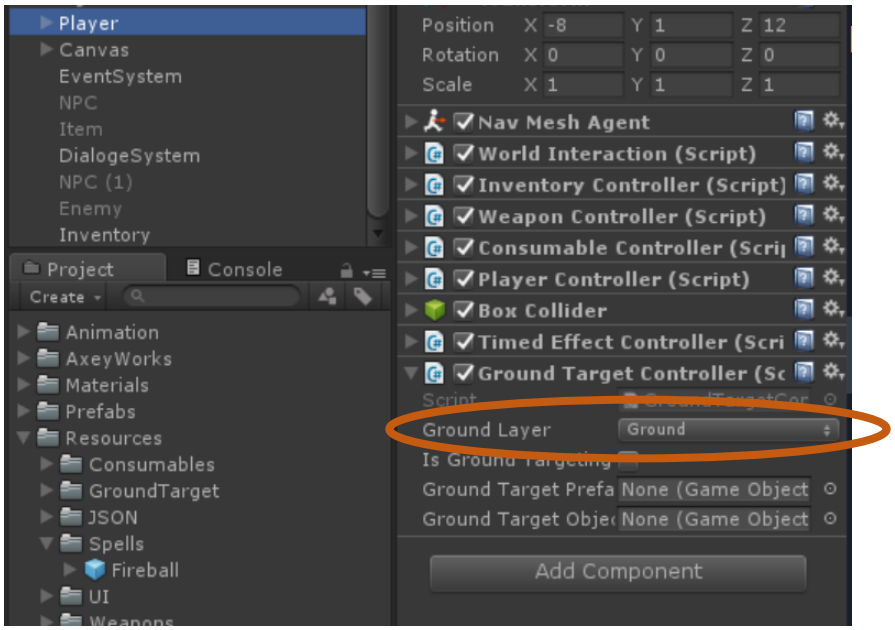
By spawning a GO on click, the system allows for any type of skill to be created. Totems, character spawns, teleportation and even the trusty meteor shower can all be created by simply writing the appropriate script, adding it to a GO and telling Ground Target to spawn that object on click.

How do I use It

1. Download the .cs file and the unitypackage. Please test import the unitypackage into a backup project so you don't lose anything.

2. Add GroundTargetController to your player. It is a singleton so you can only have one.

3. Just like the enemy requires the player to be on a specific layer, the GroundTargetController requires the ground to be on a specific layer. Add a new layer (I called mine Ground), set your ground plane so that it has this layer and then add the layer to the Ground Layer variable inside the GroundTargetController



4. In order to work, the system requires 2 items: a ground target graphic and an item to spawn when you click.

5. The ground target graphic should be located in Resources/GroundTarget/Graphics. There is an example in the unitypackage. To create yours from scratch, do the following:

    a. Create a new cylinder 3D Object. Its name should be the one you want to call from your code.
    b. Disable the mesh renderer in the inspector. The particle system will act as the visual queue for the player so we do not need to see the cylinder, however, the code uses the size of the cylinder to center the target on the mouse so you cannot delete it.
    c. Right click on the cylinder and create a new particle system (or attach one previously made. As mentioned you can use any particle property you want in your system.
    d. In order to get the particle to display correctly however, you must have specific settings under the renderer properties

Render Mode	Must be Horizontal Billboard
Material	Must be the Material that contains the graphic you want to display on the ground
Min and Max Particle Size	Have the largest effect on the size the graphic is on the ground. Change these if your graphic is too large or too small.

6. The spawned object should be placed in Resources/GroundTarget/Objects. Again, it should be named exactly what you want to call it from the code. There is an example in the unitypackage. The object can be anything that you like. In my case I spawned a meteor shower
7. 

8. GroundTargetController has 2 functions: ShowGroundTarget and ActivateGroundTarget.

9. ShowGroundTarget is the one that makes the ground circle appear. It should be called when you want to activate a ground targeting skill. It requires a string. This string should be the name of your ground target graphic from inside the Resources/GroundTarget/Graphics folder. As an example, if you want to toggle ground targeting on and off with the T key you can use the following code:

```
if(Input.GetKeyDown(KeyCode.T))
{
    GroundTargetController.Instance.isGroundTargeting = !GroundTargetController.Instance.isGroundTargeting;
    if (GroundTargetController.Instance.isGroundTargeting)
    {
        GroundTargetController.Instance.ShowGroundTarget("PracticeTarget");
    }
    else
    {
        if (GroundTargetController.Instance.GroundTargetPrefab != null)
            Destroy(GroundTargetController.Instance.GroundTargetPrefab);
    }
}
```

You would simply replace the name "PracticeTarget" with whatever your graphic is named in the resources folder. This code can be found in the update method of GroundTargetController if you wish to copy it.

10. ActivateGroundTarget spawns the item you want. It also requires a string that should be the name of the object located in Resources/GroundTarget/Objects. To call it you need to modify your WorldInteraction script. In the GetInteraction function, at the bottom modify the else clause so that it looks like this:

```
    //move
    else
    {
        if(GroundTargetController.Instance.isGroundTargeting)
        {
            GroundTargetController.Instance.ActivateGroundTarget("GroundObject");
        }
        else
        {
            navAgent.stoppingDistance = 0;
            navAgent.SetDestination(hit.point);
        }

    }
```