Setting Up a Monster

To set up a monster in your custom story, you need to place it in the level editor. Monsters can be found when you click the “Entities” tab and then go to the subsection “enemy”. For this tutorial, we'll choose the monster that is called “servant_grunt”. Select it and then place it wherever you want. I'll assume you have a least part of your level constructed and know how to make a custom story entirely. Once you placed it, go to the select tool and select the monster you just placed. Under it's name, there will be a checkbox called “Active”. Click the checkbox to set the monster inactive. You will notice that the monster is partially “faded” in a sense. We do this so that the monster doesn't exist from when you start the level.

Now go to the tab called “Areas”. The area type should say “Script”. Click in the level editor on where you want the player to touch for when you want the monster to spawn. Go to the select tool and click on the area where it says “AREA”. You will see that it's name is “ScriptArea_1”. It is possible that it could be a different number at the end. Now click on where it says it's name and change it to “PlayerCollide” without quotes. Press enter. Then near the top right corner of the level editor you'll see that there are 3 buttons. From top to bottom they say “Translate”, “Rotate”, and “Scale”. Click on the bottom one that says “Scale”. Now scale the script area box to fit whatever hallway or path the player will touch to set the monster active.

Adding Path Nodes

We will now add path nodes to the map so the monster walks from one place to another. When testing/playing your custom story, the monster will find you if you go too close to it. If the path nodes aren't near you, then the monster can't look for you over where you are, but for now we will have the monster walk from one place to another.

Go click on the tab where it says “Areas” and then go to the subsection called “PathNode”. Now make a path you want the monster to go to. Try not to put too many of them; Perhaps keep them a meter apart. Make sure the monster doesn't have to cut corners which could get him stuck. If you make him go up or down some stairs, make sure you put a path node on every step up or down. The monster will automatically disappear if it runs out of path nodes and the player isn't looking at it.

Keep track of the last path node you put. If you click on the last one, that is how many path nodes you put. It's name should be “PathNodeArea_ ” with the number of that path node is at the end.

Scripting It All Together

Now you got the monster and path nodes set up. Now it's time to script. If the player collides with an area called “PlayerCollide”, and, for this example, have ten path nodes with a monster called “servant_grunt_1”, then we can create the script for our monster.

```
void OnStart()
{
```

```
    AddEntityCollideCallback("Player", "PlayerCollide", "MonsterFunction",
true, 1);
}
void MonsterFunction(string &in asParent, string &in asChild, int alState)
{
    SetEntityActive("servant_grunt_1", true);
    AddEnemyPatrolNode("servant_grunt_1", "PathNodeArea_1", 2, "");
    AddEnemyPatrolNode("servant_grunt_1", "PathNodeArea_2", , "");
    AddEnemyPatrolNode("servant_grunt_1", "PathNodeArea_3", , "");
    AddEnemyPatrolNode("servant_grunt_1", "PathNodeArea_4", , "");
    AddEnemyPatrolNode("servant_grunt_1", "PathNodeArea_5", , "");
    AddEnemyPatrolNode("servant_grunt_1", "PathNodeArea_6", , "");
    AddEnemyPatrolNode("servant_grunt_1", "PathNodeArea_7", , "");
    AddEnemyPatrolNode("servant_grunt_1", "PathNodeArea_8", , "");
    AddEnemyPatrolNode("servant_grunt_1", "PathNodeArea_9", , "");
    AddEnemyPatrolNode("servant_grunt_1", "PathNodeArea_10", 4, "");
}
```

Keep adding more “AddEnemyPatrolNode”s if there are more path nodes that you placed and then take away some “AddEnemyPatrolNode”s if there are less path nodes that you placed. In the Script Functions Page, this is what it says about the “AddEnemyPatrolNode” command:

```
void AddEnemyPatrolNode(string& asName, string& asNodeName, float
afWaitTime, string& asAnimation);
```

Adds a patrol node to the enemy's path.

asName - internal name of the enemy
asNodeName - path node
afWaitTime - time in seconds that the enemy waits at the path node before continuing
asAnimation - the animation the enemy uses when reaching the path node

Loops

It is possible to make the monster to follow a set of patrol nodes and never stop repeating them until the monster is set disabled manually by “SetEntityActive(“MonsterName”, false);” With “MonsterName” being the name of the monster. Lets say, from the example above, that the monster follows those 10 path nodes, then reaches another 5 path nodes that he repeats until disabled. When the monster collides with the area “PNScriptArea”, he will do the 5 path nodes in a loop The example below will include the “for” loop in which I discuss in another tutorial. Feel free to check it out.

```
void OnStart()


{
```

```
    AddEntityCollideCallback("Player", "PlayerCollide", "MonsterFunction",
true, 1);
    AddEntityCollideCallback("Player", "MonsterEnd", "MonsterEnd", true, 1);
    AddEntityCollideCallback("servant_grunt_1", "PNScriptArea",
"MonsterFunction2", false, 1);
}
void MonsterFunction2()
{
    for (int i = 1; i <6; i++)
    {


AddEnemyPatrolNode("servant_grunt_1", "PathNodeArea_"+x, , "");
    }
}
void MonsterFunction(string &in asParent, string &in asChild, int alState)
{


SetEntityActive("servant_grunt_1",
    for (int i = 1; i <11; i++)
    {
        int x = i;


if (i == 1)
        {
            x = 2;
        }
        if (i = 10)
        {
            x = 4;
        }
        if (i != 1 || i != 10)


{
            x = ;
        }
        AddEnemyPatrolNode("servant_grunt_1", "PathNodeArea_"+i, x, "");


}
}
void MosterEnd(string &in asParent, string &in asChild, int alState)
{
    SetEntityActive("servant_grunt_1", false);
}


Here is some tips and tricks that can help when trying to avoid disaster
```

```
when it comes to monsters and path nodes.
```

Tips

1. A Monster can't follow the player too far away from their path that is designated to them or else they'll ignore you and continue on their path. Even the unscripted path nodes work.
2. Be careful when placing path nodes for a monster on stairs because if there is one that is missing or not placed right, it'll mess up completely.
3. The Script Function page is your friend! Under “Enemies”, there are many things that can be used to “spice it up”.
4. Any monster can bash through doors if their path nodes go through it and the box is unchecked for “DisableBreakable” for it.

This wiki entry has been made by Kyle S. If you have any comments or need help with this, send me a private message on the Frictional Games Forum. (My name on there is Kyle)