

Scares

Ever wanted to throw the player through a doorway, have him collapse, or just want to scare the crap out of your players in general? You've come to the right place! Today I'll show you, in as much detail as possible, how to scare your players without dragging out the brutes/grunts/water monsters.

Scare 1 - Throwing The Player

I once played a custom story where, at totally random times, I(Daniel) would be thrown around the room. (Apologies, I can't remember the name.) This is a great scare!

First make sure you have a little area where you'd like this to happen. For mine, I chose the bottom of a staircase which leads into another open room. It's important to make sure the area the player gets pushed into is a rather open area, unless you'd like the player to run into something and say, knock it over. For my scare, we're going to push the player into the other room, then have the door slam and lock behind him.

1. Place the script area. Once again, mine is right at the bottom of a staircase. Re-name the area to what ever you'd like. I named mine **push**. (See picture 2 at the bottom.)
2. Now you need to open the door. In the Level Editor, click on the door. Under the Entity tab, there's an option called OpenAmount. Keep clicking the + button or simply type in the number **1**. This will make the door all the way open. You can also re-name your door. Mine is called **cellar_1**.
3. Now you need to make the area the player will run into that causes the door to slam. Where you put it depends on how far you want the player to be thrown. For mine, the distance between the script area and the door is about the width of the cabinet_nice or cabinet_simple. Make this area rather large to be certain the player will collide with it. I re-named that script area **door_slam**. (See picture 1 at the bottom.)
4. Now it's time for scripting! Open the map's .hps file. We have to make **2** collides, one for **push** and one for **door_slam**.

```
void OnStart()  
{  
    AddEntityCollideCallback("Player", "push", "Push", true, 1);  
    AddEntityCollideCallback("Player", "door_slam", "Slam", true, 1);  
}
```

The Callback for **push** I named **Push**, and the Callback for **door_slam** I named **Slam**.

Now you have to make the Functions.

```
void OnStart()  
{  
    AddEntityCollideCallback("Player", "push", "Push", true, 1);  
    AddEntityCollideCallback("Player", "door_slam", "Slam", true, 1);  
}
```

```
void Push(string &in asParent, string &in asChild, int alState)
{
    PlaySoundAtEntity("", "react_pant.snt", "push", , false);
    AddPlayerBodyForce(30000, , , false);
}

void Slam(string &in asParent, string &in asChild, int alState)
{
    SetSwingDoorClosed("cellar_1", true, true);
    SetSwingDoorLocked("cellar_1", true, true);
    PlaySoundAtEntity("", "00_laugh.snt", "door_scare", , false);
}
```

Under the **Push** function, **PlaySoundAtEntity** is not necessary, but if you want that extra scare, you can keep it and Daniel will react with a gasp. The value I picked for the **AddPlayerBodyForce** is 30000. This will push the player as far as about the width of the cabinet_nice or cabinet_simple as mentioned above. I only wanted the player to be sent forward, but you might have to adjust your values based on where the door is (you might have to put it on the z one) Remember the values for **AddPlayerBodyForce** go **X, Y, Z**. Therefore the player will be pushed 30000 along the x axis. Y will send the player up or down. For **Slam**, I set the door to be closed, then locked. I also added a laugh to make it extra creepy.

Put all of that together and the scare should be a successful one! Good luck!

Pictures: [1](#) [2](#)

Scare 2 - Exploding Pots

This scare is an unexpected one. I warn anyone who wants to use this scare to NOT over-use it. It's great to use when there is peaceful music and everything seems so perfect.. and then a pot explodes, dramatically changing the music to that of a nervous feeling. I won't be showing you guys how to get the music, only how to make the pot explode. The rest is up to

you.%%</nowiki>*</nowiki>WARNING: THIS SCARE AS ONLY BEEN TESTED WITH VASE_01 AND VASE_02 FOUND IN ENTITIES>CONTAINER*

First things first: grab yourself a pot. I'll be using vase_01 found under entities>container. You can experiment with different ceramics, but only things Daniel can pick up, throw, and break will work. Place your chosen ceramic on a surface if you'd like. I'll just put mine on a small table. Re-name your 'pot grenade' to whatever you'd like. I'm naming mine **pot_explode**.

Next, make a script area. It shouldn't be too far away from the exploding pot. Make sure it's big/tall enough so the player will collide with it! Re-name it as well. I named mine **explode_scare**. Now it's already time to script!

```
void OnStart()
{
    AddEntityCollideCallback("Player", "explode_scare", "Explode", true, 1);
}
```

```
}

void Explode(string &in asParent, string &in asChild, int alState)
{
    SetPropHealth("pot_explode", );
}

void OnEnter()
{
}

void OnLeave()
{
}
```

The script is very self-explanatory. You collide with the script area, and the health of the pot goes to 0, causing it to explode. Feel free to add some smoke particles or other

sounds!

(I felt like this one didn't need any pictures, so..)

Good luck

Scare 4 - Falling Objects

This one in particular is a great unexpected scare as well. Just a simple “box slides off of the top shelf and hits you in the head” scare. This is also not a complicated scare, though if you wish to make it complicated feel free to add health damage when the box collides with Daniel's head. I'd also like to mention I used a specific box for this one: `storage_box_wood01` found under Entities>Storage. You can experiment with different objects or boxes, but be sure they aren't static and Daniel can easily pick them up and toss them.

First get a box. As mentioned, I'm using `storage_box_wood01` which I re-named to **box_1**. Now we need a shelf. Any shelf will do really, but I'm using `shelf02` found under Entities>Storage. The shelf does not need to be re-named as it's only purpose is to hold the box.

Place your box ontop of the shelf. The very very top is the preferred place, but if you use smaller objects like pots, it can go on any shelf. Just be sure it fits right. Because the shelf I chose has a back, the box had to be placed slightly hanging off the edge of the shelf. This is perfectly fine to do, just make sure the box doesn't slide off before we add a prop force. Last thing we need is a script area. Because I want the Player to notice the box, I've placed my script area about a foot away from the actual shelf. If you place the script area right below the box, the box may not fall in time and scare the player, but confuse them. However you can place the script area wherever you'd like. I've re-named mine **box_fall**. Time for scripting!

```
void OnStart()
{
AddEntityCollideCallback("Player", "box_fall", "Falling", true, 1);
}

void Falling(string &in asParent, string &in asChild, int alState)
{
AddPropForce("box_1", , , 1500, "world");
}

void OnEnter()
{
}

void OnLeave()
{
}
```

Quite a few of you were confused on the prop force last time, by just copying what my script says. You NEED to check which way the object needs to go! For some it's Z and for others it will be X. It all depends on where you started on the map and the direction the shelf faces. To see how to find which variable to use, please refer back to my Flying Jesus scare.

The script here is rather self-explanatory. The Player collides with the script area, it triggers the prop force, the box falls. Super easy! I also didn't add pictures for this one as I figured it wouldn't be too difficult to figure out. Though if you need a picture reference I will add some.

Scare 5 - Kill The Lights

This scare was requested by [BrooksyFC](#). Check him out on Frictional Games!

BrooksyFC was inspired by the ending of Penumbra: Overture and asked me to re-create this scare. All of the lights in a long corridor go out, one by one. Today I'll be showing you using **6 wall candles**, but feel free to use more or less; even a different lamp will do.

As usual: we set up the script areas/lamps first. Place as many lamps as you'd like in a long corridor. (Please note: the more lamps you use, the longer the script will be!) Re-name the lamps something simple, like **lamp_1**, **lamp_2**, **lamp_3**, **lamp_4**, etc. [TIP: NAME THE LAMPS IN THE NUMBER ORDER YOU WANT THEM TO GO OUT. I FIND IT MUCH EASIER!] Now, throw down a Script Area. I've named mine **KillTheLights**.

Script time!

```
void OnStart()
{
```

```
AddEntityCollideCallback("Player", "KillTheLights", "LightsOut", true, 1);
}

void LightsOut(string &in asParent, string &in asChild, int alState)
{
SetLampLit("lamp_1", false, true);
AddTimer("", 2, "Out2");
}

void Out2(string &in asTimer)
{
SetLampLit("lamp_2", false, true);
AddTimer("", 2, "Out3");
}

void Out3(string &in asTimer)
{
SetLampLit("lamp_3", false, true);
AddTimer("", 2, "Out4");
}

void Out4(string &in asTimer)
{
SetLampLit("lamp_4", false, true);
AddTimer("", 2, "Out5");
}

void Out5(string &in asTimer)
{
SetLampLit("lamp_5", false, true);
AddTimer("", 2, "Out6");
}

void Out6(string &in asTimer)
{
SetLampLit("lamp_6", false, true);
}

void OnEnter()
{
}

void OnLeave()
```

{

This script is only good for **6 lamps**. If you have more, the easiest way is to copy/paste a void Out# section and just adjust it. When you reach your last desired lamp, delete the last Timer. It's your last one, so you don't need another timer.

Thanks for tuning in, and happy scaring!

Created by JenniferOrange

IF YOU HAVE ANY OTHER REQUESTS FOR SCARES, PLEASE PM ME AT

<http://www.frictionalgames.com/forum/user-15434.html> (YOU DO NEED TO BE EITHER REGISTERED/LOGGED IN TO FRICTIONALGAMES TO VIEW THIS PAGE!!)

From:
<https://wiki.frictionalgames.com/> - **Frictional Game Wiki**

Permanent link:
<https://wiki.frictionalgames.com/hpl2/tutorials/scripting/scaresbyjenniferorange?rev=1334798684>

Last update: **2012/04/19 02:24**

