

**Edit1: Their seems to be confusion of who is the Author of this. It's me "Rapture" on the frictionalforums. Apparently the HPL2 wiki placed this under Elven's tutorials, including my outdoor\_level\_design. Sorry!**

**Edit2: In addition, I'm adding another section to this Elevator Tutorial. Scroll down till you see PART 2.**

This tutorial will teach you how to make a fully functional elevator that travels along the Y-Axis (Up & Down). There are probably a lot better ways of making an Elevator. But I find this to be a simple brute force technique that requires only basic scripting knowledge.

This elevator will be created with the default *Amensia: The Dark Descent* models. Before starting this tutorial, please download the map. Please refer to the map for objects and their dimensions, names, properties, position and etc.

[www.mediafire.com/](http://www.mediafire.com/)

To start off with, make a semi-large area. The dimensions are (Including *wall\_corner\_concave\_round* on the sides) 6 (*wall\_default*) wide, by 3 (*wall\_default\_15*) length, by 5 stacks tall. Fill the top with the (*ceiling\_concave*) and (*ceiling\_corner\_concave\_round*) and a ceiling plane or something.



The objective of this challenge, is to bring some boxes to a platform, far too high for you to reach or throw them.

We will place down some crates and barrels that signify our Damsel in Distress. A platform (Name = "**Entity\_Elevator**") as our makeshift elevator, a piston (name = "**Entity\_Piston**") to raise the elevator (I have tried different methods such as PropImpulse, but this can be very buggy.) and a Lever to activate the sequence.



First off, name your Lever to "**Entity\_ElevatorLever\_01**". In its "Entity" property tab, check "**Override Defaults**" and set the "**MiddleAngleAmount**" beneath it to "-0.8".

Then place a *Script Area* underneath it, and name it to "**Area\_ElevatorLever\_01**". Make sure it's beneath it and not touching it.



```
void OnStart()
{
    SetSanityDrainDisabled(true);

    GiveItemFromFile("lantern", "lantern.ent");

    for(int i=0;i<25;i ) GiveItemFromFile("tinderbox_" i, "tinderbox.ent");

    AddEntityCollideCallback("Entity_ElevatorLever_01",
    "Area_ElevatorLever_01", "Script_ElevatorLever_01", true, 1);
}
```

```
SetEntityInteractionDisabled("Entity_Elevator", true);

SetPropStaticPhysics("Entity_Elevator", true);
}

void Intro()
{
}

void OnEnter()
{
    PlayMusic("07_amb", true, 1.0f, 4.0f, 1, true);
}

void Script_ElevatorLever_01(string &in asParent, string &in asChild, int
alState)
{
    SetMoveObjectStateExt("Entity_Piston", 0.6, 0.6, 8, 0, false);

    AddDebugMessage("Elevator is about to start!", false);
}
```

When you pull down the lever, it will trigger a *AddEntityCollideCallback*. Great, we have a elevator that works. But theirs alot of problems...

1. I can shove the Elevator off, destroying it.
2. Your not on the elevator.
3. Oh, it was a great idea at first. But how do I get down? How do I get up if I fall down? I'm down here and its up their, how do I call it back down? ... .. /ragequit.
4. Do we want the piston to be seen? In this test case, it is alright because the elevator is not magical.

Yep, bunch of things we developers have to work out. We have to dummy-proof these simple contraptions to keep morons or trolls from ruining the immersion.

Addressing problem one. We have to first make sure that the elevator is stuck and in place. We will be using the "**SetEntityInteractionDisabled**" function for this purpose. Disabling player interaction will keep the player from grabbing and tossing the thing around. But theirs still the problem of the player pushing it with their body, or using a object to push it off. We will then use the "**SetPropStaticPhysics**" to keep it rigid and in place.

```
void OnStart()
{
    SetSanityDrainDisabled(true);

    GiveItemFromFile("lantern", "lantern.ent");

    for(int i=0;i<25;i ) GiveItemFromFile("tinderbox_" i, "tinderbox.ent");

    AddEntityCollideCallback("Entity_ElevatorLever_01",
```

```
"Area_ElevatorLever_01", "Script_ElevatorLever_01", false, 1);

SetEntityInteractionDisabled("Entity_Elevator", true);

SetPropStaticPhysics("Entity_Elevator", true);
}
```

Problem two. There's a couple of ways to tackle this, one way is to have a timer before the elevator starts to move up. That may work for your CS, maybe it does not.

Another way is to have a Area, already Not-Active. But activated when the lever is used. The player must then run to the Area (On your Elevator) to activate the elevator. This could be more practical if say your elevator was in a far away area. And saves the player from wasting time waiting for the elevator to go up and/or down.



We would name the Script Area, "**Area\_BottomElevator\_PlayerStart**" and uncheck its "**Active**" state.

In our script, we would do these modifications...

```
void OnStart()
{
    SetSanityDrainDisabled(true);

    GiveItemFromFile("lantern", "lantern.ent");

    for(int i=0;i<25;i ) GiveItemFromFile("tinderbox_" i, "tinderbox.ent");

    AddEntityCollideCallback("Entity_ElevatorLever_01",
"Area_ElevatorLever_01", "Script_ElevatorLever_01", false, 1);
    AddEntityCollideCallback("Player",
"Area_BottomElevator_PlayerStart", "Script_StartElevator", false, 1);

    SetPropStaticPhysics("Entity_Elevator", true);
}
void Intro()
{

}

void OnEnter()
{
    PlayMusic("07_amb", true, 1.0f, 4.0f, 1, true);
}
//*****
*****
void Script_ElevatorLever_01(string &in asParent, string &in asChild, int
alState)
{
```

```
SetEntityActive("Area_BottomElevator_PlayerStart", true);

AddDebugMessage("Entity_ElevatorLever_01 has been pulled.", false);
}
//*****
void Script_StartElevator(string &in asParent, string &in asChild, int
aIState)
{
SetPropStaticPhysics("Entity_Elevator", false);

SetMoveObjectStateExt("Entity_Piston", 0.6, 0.6, 8, 0, false);

AddDebugMessage("Elevator is about to start upwards!", false);
}
//*****
*****
```

Now if you were the use the Lever, you would have all the time in the world before deciding to ride it.

To adress the 3rd problem, just copy what you did to make it go up. Reverse everything (Including the renaming, areas, etc... These are the most likely places were you will screw up on!) so it goes down. Wala, you have a working elevator. Now we will quickly address other problems and fine tuning you may do.

*Btw, check up in the "ElevatorTutorial\_01.hps" for the full code as it may adress the problems below more and less.*

**Just so you know, the Elevator in the tutorial map can be used repeatedly up & down.**

Misc. Problems

1. When the elevator is going up and stops. You will see your character and any objects on the elevator make a jump. I know of some other variables to why it happens, but the easiest fix would just to limit the elevator's maximum speed to reduce this effect.
2. At the end, the elevator will also rise up for a bit before touching back down onto the piston. To fix this, you would make a Script Area that would trigger that would turn the Elevator Static again. The code & map will have it all done for you, so no need to worry about that. But for your map, you will have to reposition the Script Areas.
3. In the full script, I also disabled the levers after they were used. It gives feedback back to the player that it has been used properly and might of done something important.
4. Ok so your on top, and you fell down. Well I put a ladder as a way to get back up, since I haven't included in the Tutorial a Area Script that would kill you if you jumped from that height. You can figure out how to make a lever on either the top or bottom control the piston to go up or down. Not just one direction at a time.
5. To achieve a magical elevator. (Hint: Use Pink Block boxes & SetEntityActive)
6. One Con to this, is that you can only use the Piston in the default Amensia model kit. Their might be a few more you could use, but I'm not aware of them. You would either have to make a custom model, and make it work the same way as a piston. Or use a invisible block box, that can be moved up and down like a piston.

Someone was wondering to how you could get the **Static Object > Machine > Elevator** to work. First of all, it's a Static Object. Static Objects cannot be moved or changed. We have to go into the Model Editor and change it so it will work the way we want it to. And before I get into that, someone else asked to why I would make this Tutorial if the Elevator in Amnesia could serve the same purpose...

No, the Elevator in Amnesia does not travel in the Y-Axis like you are led to believe. The only part moving is another special created entity that moves, and they add some sparks & sounds to make you believe that its moving. Very clever trick, but it's usage is low and does not serve as a living, breathing, moving Elevator. If you don't believe me, check it in the Level Editor yourself. And next time your playing the game, activate the Yellow Microphones (Developers talk about the game) and interact with the one next to the Elevator. He will explain everything that you need to know about it.

---

## PART 2

Now to start off, go into your **Amnesia root folder, then > static\_objects > machine. In their are 6 files starting out with "elevator", select and copy them.**



Go to your **Amnesia the Dark Descent > entities** folder now. Since this is a edited prop, we might as well put it under a folder we can easily recognize. I prefer to put it under a folder under my name so its easier to manage. And withen "**Rapture**" I make a new folder called "**Elevator**".



Now, go back to your **Amnesia the Dark Descent** root folder. Inside it, near your *Level Editor*. Their should be a **Model Editor**, open it up. Do a **File > Import Mesh**. Go to your **Amnesia the Dark Descent > entities > "Your Folder Name" > Elevator > elevator.dae**

This will generate a elevator.ent and you should see it properly display in front of you.

On the left side, their should be a button called **Shapes** (*Hotkey = 6*)



Plop one down, and lets start off with the floor. Scale it on its X, Y, Z axis's until it roughly covers the ground floor like so.



Repeat this step for all 5 sides.



Once that is done, select all 5 purple boxes. *After your done selecting them, as a test. Move them around to make sure you haven't select the Elevator mesh by accident.*

On the right side, their should be a option "**Create Body**", click it. Go to the "**Body**" tab on its

properties (Right side again). And as for now, change it to this... (You can always fidget with the properties later, so no worries)



Then go to the “**Attachments**” tab. Click the “**Su**” button, and click “**Attach**”. Now click in the general area of your Elevator (It will not select your boxes) and it should be highlighted. Click “Done” to finish. If done properly, you should see a green line attached to your Elevator and the Combined green body IF YOU MOVE THEcombined green body around. You WILL SEE A RED LINE if you move the Elevator body around.



Since we will need lighting for the Elevator, we'll just take a shortcut and add a...

MORE TO COME SOON...

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

Permanent link:  
[https://wiki.frictionalgames.com/hpl2/tutorials/scripting/elevator\\_tutorial](https://wiki.frictionalgames.com/hpl2/tutorials/scripting/elevator_tutorial)

Last update: **2012/05/18 20:26**

