

Entities

Work in progress page.

****Please add more information about missing subjects****

This page talks about the different types of entities found within the HPL2 engine. The type determines how the entity will be set up, what kind it is, what it does under different circumstances etc. Using the Model Editor you can identify and specify which type an entity has. In the Level Editor, the different types have different settings applicable for them. For instance a MoveObject type will have the OffsetArea box available in the Level Editor, and an Item (Note) type will have a NoteText box available.

A list of all the types and subtypes can be found in the table of contents to the right.

StaticProp

A StaticProp entity resembles a static object. They have no available type properties, but they can still use a collision box and have mass/gravity. Note that they cannot be interacted with in the game, and cannot be disabled. If you need this feature, use Object (Static) instead.

CommentaryIcon

A CommentaryIcon entity is simply an object that can execute commentary sound files, as it was done in the main game. These only appear if the player has "Commentary" enabled in their settings.

Object

Static

Not to be confused with an actual static object, the Object type is a general object with common properties. The static subtype simply disables any additional properties from the other subtypes. An Object (Static) is similar in concept to a StaticProp type entity, however the Object (Static) can be disabled in the game and interacted with by the player.

Grab

A Grab subtype is a normal Object entity with the ability to be grabbed by the player in the game. In order for this to work, the object must have a mass above 0.

Push

Similar to the Grab subtype, the Push subtype allows the player to push the object by taking hold of it and walking or right clicking. Example of this is the prison bed.

Slide

A Slide Object is similar to the Grab subtype, except it can only be slid in a single direction for a certain amount. An existing entity that uses this is the secret floor carpet you can slide to the side.

SwingDoor

A SwingDoor entity is just that; a swing door. Any door you can grab and swing open is considered a SwingDoor. The other option would be LevelDoor, which goes under its own category. To use a SwingDoor, you need to attach joints to where you want the door to swing around. These are usually placed by the hinges of the door. On the joint you can set the min and max values to limit how much the door can be opened. These are usually approximately 90 degrees apart in either positive or negative directions.

Lever

A Lever entity is an entity that can be swung between states to trigger callbacks. A lever requires a joint to swing around.

Wheel

A Wheel entity is a wheel that can be turned. They use states similar to the levers. Some wheels can be turned indefinitely and some (mostly valves) can be turned for a fixed amount in order to trigger an event. Wheels require a joint.

MultiSlider

Add info.

Lamp

A Lamp entity is a light source. If they are unlit, they can be lit by the player, consuming a tinderbox. Examples are torches and candles.

Button

A Button entity acts like a lever, except it does not use a joint. It simply calls a callback when interacted with.

Photocell

Add info.

MoveObject

A MoveObject entity is an entity that can be specifically moved on a single axis through the use of scripting. It can be set to either Angular or Linear. Angular rotates the object around its own center point, unless otherwise specified by the use of an OffsetArea. Linear moves the object in a direction for a specified amount. Note that a linear move object can only move on one axis. You can pick X, Y or Z in the entity and this cannot be changed mid-game. If you want the same object to move on more than 1 axis (for example left then forward) you can switch out the entity with an identical one that uses another axis. The script to move a MoveObject is SetMoveObjectState and SetMoveObjectStateExt, found on the engine scripts page.

Item

Puzzle

A Puzzle subtype item is a general item with no extra properties. Use this if you only need an inventory item to be used to trigger a scripted action. Other subtypes have special properties, but Puzzle only has the ability to be collected and used from the inventory. When used, it's loaded onto the crosshair so the player can use it on objects.

Coins

Add info.

Note

A Note item is an item that adds text to the player's journal. It displays itself when picked up. The note can be narrated, in which case it will automatically close once it's finished. If it isn't voiced, the player must close it. Use this if you want to make a Note item using a custom model / custom properties that differ from the main notes.

Diary

A Diary item is nearly identical to the Note item. Instead, it adds itself to the Diary section of the player's journal, whereas the Note adds itself to the Notes section. Diaries are also always voice enabled. Notes have the option for it.

Lantern

A Lantern item can be picked up like a normal item, but is added as a lantern to the inventory. While having this item, the player can press F to enable/disable it, or click it in the inventory. Note: This is NOT the lantern model, only the item that gives it to the player. If you want a custom lantern model, you can follow [this guide](#).

Health

A Health item acts like a health potion (Laudanum). It acts like a normal item, except when used it is consumed while increasing the player's health by a specified amount.

Sanity

A Sanity item acts like a sanity potion. While used, they increase the player's sanity by a fixed amount, and then consumed from the inventory.

LampOil

A LampOil item acts like an oil potion. When used, it increases the amount of oil for the lantern, and then consumed.

Tinderbox

A Tinderbox item adds itself to the player's tinderbox counter in the inventory. Tinderboxes can be used once to light a light source.

HandObject

Add info.

Chest

Add info.

OilBarrel

An OilBarrel entity simply adds some oil to the player's lantern storage upon interaction. The oil amount will be random unless otherwise specified.

EmotionStone

An EmotionStone entity works like the glowing jar-like objects from the game. Upon interacting with an EmotionStone entity, the screen fades to white and displays a message for the player.

NPC

Add info.

LevelDoor

A LevelDoor entity is in concept similar to a SwingDoor, but they function very differently. A LevelDoor instantly transports the player to the specified level when they click it. The sounds, text and destination can be entered in the Level Editor itself.

CritterBug

Add info.

Enemy_Grunt

An Enemy_Grunt entity is a standard enemy entity. It uses properties like the grunt and brute in the game. Enemy entities are advanced to set up, and should include animations added through Settings > Animations. Enemy properties can be modified in the entity type window, like certain sounds and music tracks playing, the enemy's awareness levels and damage, speed and other similar settings. An enemy does not use a collision box, but instead a skeleton and the before mentioned properties to determine how to collide/hit something.

Enemy_Waterlurker

Add info.

PlayerHands

Add info.

HandObject

Add info.

From:

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

Permanent link:

<https://wiki.frictionalgames.com/hpl2/amnesia/entities>

Last update: **2014/12/16 13:40**

