

5. HPL Helper

5.1 General

HPL-Helper is a front-end for various programs and an editing tool for various engine purposes. Latest version of the .Net Framework must be installed to get it working. See [Getting Started](#) for details and download location.

5.2 Settings

Under this tab various settings can be altered.

5.2.1 Resource Directories

The HPL-Engine doesn't use paths when searching for files, it only searches after specific filename. The option lets you set what directories the engine will search for the files in.

Add:	Adds a new directory to the list. This MUST be a subdirectory of the folder that HPL Helper is in!
Remove:	Simply just removes a directory from the list.
Save To File:	Saves all the directories to a file, this button must be pressed for the resource directories to be used by the applications.

5.3 Models

This part of the program is used to view and convert model files.

5.3.1 Model File

Click browse and choose the file you want to work with. *You can open a .dae file directly or you can use a .ent file to get some/most of the ingame functionality for the model.*

5.3.2 View

Click this button to open the model viewer. Inside the viewer hold the left mouse button and move the mouse around to change the view. Press middle button and move the mouse up and down to zoom in and out. If the model has colliders use the right mouse button to hook onto the model and then move the mouse while holding the button down to drag it around.

For the time being there are no nice menus so one has to deal with using keys to toggle features.

1:	Turn shadows on /off.
2.	Show/hide colliders.
3.	Turn physics on /off.
4.	Change animation.
5.	Start/stop spinning lights.
6.	Show/hide bounding box.
R	Turn ragdoll on/off if present.

5.4 Scenes

Under this tab a scene can be loaded and viewed. This works exactly the same way as with models (described the previous chapter) apart from a few small things.

Beside the View button you can fill in the start pos. This will specify at what start point the camera will start at, you are using the `_start` entities for this, if you have a "`_start_location1`" entity in you map you enter "location1" to make that your starting point. If this is blank or the start position doesn't exist the camera will be placed at (0,0,0).

The following keys are used in side the viewer:

W,A,S,D	Movement.
Mouse	Look around.
1.	Toggles shadow modes, on, static only, off
2.	Turn room/portal debug graphics on and off. In this mode the rooms are randomly colored according to the order they where created. The portals are colored red and have a normal in the same color to witch the room it belongs to. At the end of the portal normal is a sphere in the same color as the room the portal connects to.
3.	Turns light bounding boxes on/off.
4.	Turns collision geometry on/off.
5.	Turns bounding boxes for entities on and off.
6.	Turns sound colliders on / off.
7.	Turns line rendering on / off. This is useful for optimizing portals.
8.	Turns show areas on / off.
B	Bloom on / off

5.5 Materials

There shouldn't be anything strange in this part, the different settings are all explained in [chapter 4](#).

Just remember that the changes aren't updated until the file is saved!

5.6 Physics Materials

Later on there should be an editor for physical material, right now one has to write by hand. In the `redist` folder you'll find the **`materials.cfg`** file which you can edit and add materials to.

Physics materials are different from normal materials in that they describe physical properties of a surface/object rather than the visual. They describe what sounds are heard when two objects hit each other, how much they bounce, etc. If no material is set "Default" is used.

There are several properties that need some explanation:

Name

Simply the name of the material, entities and texture refer to this.

Elasticity

How much of the energy that is conserved when the object hits another object. 0 = all is lost and 1 = no energy loss. This means 1 creates a bouncy surface or object and 0 creates a something that just stops at impact.

StaticFriction

The amount of energy required to get an object moving. The value can be between 0 and 1 (even up to 2 in special cases) and MUST be larger or equal to kinetic friction.

Priority

Gives materials different priorities, materials with the same priority will blend together while a higher priority for one material will make it dominant and the only one used.

KineticFriction

How fast the object stops sliding once it is moving (static describes how hard it is to get it start moving). This is a value between 0 and 1.

ElasticityMode

When two objects collide this describes how the elasticity values blend. See BlendMode for more information.

FrictionMode

When two objects collide this describes how the friction values blend. See BlendMode for more information.

BlendModes:

Each blendmode have a weight and when two objects collide the mode with the highest weight is used. Weight is written in front of the mode.

0. Average:	The average of the two values is used ($r = (x+y)/2$)
1. Min:	The smaller of the two values is used.

2. Multiply	The two values are multiplied together. ($r = x * y$)
3. Max	The largest value is used.

StepType

A game specific string meant to help specify what step sound to use. There are 3 kinds of steps: sneaking, walking and running. Make sure to name the sound entity files you want to use with the same ending, for

example:player_step_sneak_dirt.snt,player_step_walk_dirt.snt,player_step_run_dirt.snt.

This way the game will know what sound to use for the different kinds of steps. In the materials.cfg file make StepType = Last word in the name, meaning StepType = "dirt". This will point the game to the player_step_xxx_dirt.snt files and automatically load them all 3.

Scrape sound properties

The following properties are used to set what sound is played when an object slides against some other object.

Speed is in m/s and frequency is in percent / 100. This means that if frequency is 1 then the sound is played at normal (100%) speed. If it is 0.5 then the sound is played at half speed and 2 means double speed.

MinScrapeSpeed	The minimum speed at which the sound is played. 0 - infinity.
MinScrapeFreq	The minimum frequency the sound will have. 0 - 1.
MinScrapeFreqSpeed	The object speed at which the minimum frequency will be used. Frequency never goes lower once this speed is reached. 0 - infinity.
MaxScrapeFreq	The maximum frequency the sound will have. 0 - infinity.
MaxScrapeFreqSpeed	The object speed at which the minimum frequency will be used. Frequency never goes higher once this speed is reached. 0 - infinity.
MiddleScrapeSpeed	The object speed at which normal frequency (1) is used
MinScrapeContacts	The minimum number of contacts for a the scrape sound to play. 0 - infinity.
ScrapeSoundName	The name of the sound entity played.

Roll sound properties

The following properties are used to set what sound is played when an object rolls.

Speed is in m/s and frequency is in percent / 100. This means that if frequency is 1 then the sound is played at normal (100%) speed. If it is 0.5 then the sound is played at half speed and 2 means double speed.

MinRollSpeed	The minimum speed at which the sound is played. 0 - infinity.
---------------------	--

MinRollFreq	The minimum pitch frequency the sound will have. 0 - 1.
MinRollVolume	The minimum volume the sound will have. 0 - 1.
MinRollFreqSpeed	The object speed at which the minimum frequency will be used. Frequency never goes lower once this speed is reached. 0 - infinity.
MaxRollFreq	The maximum pitch frequency the sound will have. 0 - infinity.
MaxRollVolume	The volume the sound will have at maximum. 0 - 1.
MaxRollFreqSpeed	The object speed at which the maximum frequency will be used. Frequency never goes higher once this speed is reached. 0 - infinity.
MiddleRollSpeed	The object speed at which normal frequency (1) is used
RollSoundName	The name of the sound entity played.
RollAxis	Along which axis the roll is for, X,Y,Z.

Impact properties

The impact properties are used to define effects at impact. They are written like:

```
<Impact MinSpeed="2.1" [...] />
<Impact MinSpeed="1.1" [...] />
[...]
```

The impacts must be written with the highest minimum speed first and then in dropping order.

MinSpeed	The minimum object speed at which this effect is used.
SoundName	The name of the sound entity played.
PSName	The name of the particle effect to use.
PSPrio	What priority to use on the effect, the same uses all effects for all involved materials and higher uses only the highest.

Hit properties

The hit properties are used to define effects from hitting on an object. They are written like:

```
<Hit MinSpeed="3" [...] />
<Hit MinSpeed="1" [...] />
[...]
```

The hits must be written with the highest minimum speed first and then in dropping order.

MinSpeed	The minimum object speed at which this effect is used.
SoundName	The name of the sound entity played.
PSName	The name of the particle effect to use.
PSPrio	What priority to use on the effect, the same uses all effects for all involved materials and higher uses only the highest.

In closing a physical material entry looks somewhat like this:

```
<Material
  Name = "Name_Of_Material"
  Elasticity = "0"
  KineticFriction = "0.2"
  StaticFriction = "0.2"

  Priority="0"

  ElasticityMode = "Min"
  FrictionMode = "Min"

  StepType="The_Sound_For_Steps"

  MinScrapeSpeed = "0.04"
  MinScrapeFreq = "0.5"
  MinScrapeFreqSpeed = "0.7"
  MaxScrapeFreq = "0.8"
  MaxScrapeFreqSpeed = "2.5"
  MiddleScrapeSpeed = "1.5"
  MinScrapeContacts = "3"
  ScrapeSoundName = "The_Sound_For_Scrape"

  MinRollSpeed = "0.05"
  MinRollFreq = "0.5"
  MinRollVolume = "1"
  MinRollFreqSpeed = "0.5"
  MaxRollFreq = "0.8"
  MaxRollVolume = "1"
  MaxRollFreqSpeed = "1.5"
  MiddleRollSpeed = "1"
  RollSoundName = "The_Sound_For_Rolling"
  RollAxis = "y"
>
  <Impact  MinSpeed="30"  SoundName="Hardest_Impact_Sound" />
  <Impact  MinSpeed="20"  SoundName="Medium_Impact_Sound" />
  <Impact  MinSpeed="10"  SoundName="Softest_Impact_Sound" />

  <Hit  MinSpeed="3"  SoundName="Hit_Sound_Hard"
PSName="Hit_Particle_Effect_Hard"  PSPrio="10" />
  <Hit  MinSpeed="1"  SoundName="Hit_Sound_Soft"
PSName="Hit_Particle_Effect_Soft"  PSPrio="10" />
</Material>
```

5.7 Sound

This will allow you to create sound entity files, .snt. A sound entity file is a XML file that contains properties for a sound you want to play, for example if it should loop, at what volume, the radius and so on. It also contains information of what sound file to actually use, a valid sound file is in either .wav or .ogg.

IMPORTANT:

Sound names must be written in special way. The files must be named [name][number].[extension], for example: "mysound1.wav" and "test_sound1.ogg". If you have several sounds that you want randomized you name them the same and with an increasing number (9 is maximum though). Example: "testsound1.wav", "testsound2.wav" and so on.

The names above are written **without** the extension and number. So if you have a sound called "testsound1.wav" you would write "testsound". If there are more sounds named testsound with increasing number then writing "testsound" will take one of these at random. This part functions pretty much like Material. Below is an overview of the settings:

Main Sound

The main sound played. *Remember delete the number and extension from the sound you choose, "testsound1.wav" should be entered "testsound".*

Start Sound

If the sound is looping and interval is set to 0, then this will be played before main sounds start playing. *Remember delete the number and extension from the sound you choose, "testsound1.wav" should be entered "testsound".*

Stop Sound

Just like start sound but will be played after the main sound stops. *Remember delete the number and extension from the sound you choose, "testsound1.wav" should be entered "testsound".*

Fade Start / Stop

If the main sound should fade in/out when start / stop sound is playing. (Not implemented yet).

Loop

If the sound should looped.

Use3D

If the sound is heard in 3d or not affected by the player moving. Example: A sound coming from a leaking steam pipe should Use3D, but the ambient rumble in a level should not Use3D.

Stream

If the sound is streamed.

Volume

The volume of the sound. Valid values are 0 -1.

Min Distance

The distance at which the sound starts getting lower. Valid values are 0 - infinitive.

Max Distance

The distance at which the sound is quiet. Must be larger than min distance.

Interval

If looped the sound is played in intervals. The interval is in seconds. Valid values are 0 -infinite.

Random

The chance of the sound playing at each interval. Valid values are 0 - 1.

Blockable

This determines if the sound is blocked by sound blockers.

Block Volume

This is the amount the volume is lowered when behind a sound blocker. Valid values are 0 - 1. 1 will not affect the sound and 0 will kill it completely, usually a value of about 0.5 to 0.8 gives a nice result.

Priority

This sets what priority the sounds has in the game. The engine sets it's own priority depending on where the sounds is.

Outside max distance: 0

Outside min distance: 10

Inside min distance: 100

The priority property is then added to this value. To make some collision sound more important then other collision sound the priority should be set to 0 - 100 depending on you want it to go before sounds inside min dist if it the sound itself is outside. If you have for example a radio message that must be heard and cannot be started over then set the prio to above 200.

Valid values are 0 - 255. Higher value means higher priority.

Note: When looped sounds are overridden by another, the looped sound is started over once a channel gets free. This shouldn't be any problem for most sounds though.

From:

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

Permanent link:

<https://wiki.frictionalgames.com/hpl1/documentation/content.creation.document.chap5>

Last update: **2010/11/04 07:25**

