

Using Message Pop-Ups Properly

Today I'll be teaching you how to make those little messages pop up on the bottom of your screen. These can be perceived as thoughts, or if you really want to annoy people, saying things like: "OMG, a monster! D:"

So what exactly do we use them for? If you don't want to use memos, which the Player may not even care about to check, you can use messages. When colliding with a script area, a message will pop-up on the bottom of the screen giving the Player a hint or expressing Daniel's thoughts. (i.e.: Approaching a barricade and using a message that hints the Player they "saw another way out back there.") Messages are very helpful, and it's difficult for the Player to ignore them, as they appear quite prominently on the screen. Let's learn how to create them.

Creating Messages

First things first: open your extra_english.lang. Inside, if the category named Messages is not present, you'll need to add it.

```
<LANGUAGE>
  <CATEGORY Name="CustomStoryMain">
    <Entry Name="Description">YOUR DESCRIPTION HERE</Entry>
  </CATEGORY>
  <CATEGORY Name="Inventory">

  </CATEGORY>
  <CATEGORY Name="Messages">                                <Entry
Name="Popup1">YOURMESSAGEHERE</Entry>                        </CATEGORY>
    <CATEGORY Name="Descriptions">

  </CATEGORY>
    <CATEGORY Name="Levels">

  </CATEGORY>
</LANGUAGE>
```

The category that needs to be added has been highlighted. I named the message Popup1, but you can name it whatever you'd like. [KEEP THIS NAME IN MIND AS YOU CONTINUE!] This is also an example of a standard, blank language file.

Once that's been added, open up your map and place your script areas. Re-name them things like **Message_1** since that's their sole purpose. Don't forget to hit ENTER!

After that, the last thing to do is open your .hps. We're going to use the function AddEntityCollideCallback.

```
void OnStart()  
{  
AddEntityCollideCallback("Player", "Message_1", "Message1", true, 1);  
}  
  
void Message1(string &in asChild, string &in asParent, int alState)  
{  
SetMessage("Messages", "Popup1", ); //This stands for ("Name of category",  
"Name of entry", time to be displayed on the screen);  
}  
  
void OnEnter()  
{  
}  
  
void OnLeave()  
{  
}
```

We use the function SetMessage. The first parameter, "Messages", will always stay the same so long as you both shall live. The second parameter, "Popup1", is the name of the Entry I told you to keep in mind that we used earlier. It's subject to change. The last parameter is a value. It stands for how long the message is to be displayed. By using 0, it estimates on how long the message is and makes a decent pause so you can read it. However you can put your own values in. And that's it! Enjoy using as many messages as you want.

Created by JenniferOrange

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

Permanent link:
https://wiki.frictionalgames.com/hpl2/tutorials/scripting/messages_jenniferorange?rev=1327804927

Last update: **2012/01/29 02:42**

