

Advanced Timers

This wiki page will explain the advanced usage of timers.

Timers are set up to wait for the selected amount of time to pass before doing specific commands. In introductions for custom stories, I've seen some that are too quick. For example, the player is in his bed and gets up. Naturally, it should take a little more time unless purposely done so. Like if the player wakes up to the breaking of glass or strange sounds.

Here is what a simple timer could look like:

```
void OnStart()
{
    SetPlayerCrouching(true);
    SetPlayerActive(false);
    FadeOut();
    FadeIn(3);
    AddTimer("", 3, "Intro");
}
void Intro(string &in asTimer)
{
    SetPlayerCrouching(false);
    SetPlayerActive(true);
    PlaySoundAtEntity("", "react_sigh.snt", "Player", , false);
}
```

This script doesn't have any sort of “flavor” to it. An introduction is supposed to make the player feel interested or worried of what is to come ahead. I'm not trying to say that the people who script their custom story like this isn't experienced, I'm saying that they should try to take a risk with something bigger and better. This all leads to the advanced usage of timers.

This is what an advanced timer could look like:

```
void OnStart()
{
    PlaySoundAtEntity("", "break_glass_bottle.snt", "Player", , false);
    SetPlayerActive(false);
    SetPlayerCrouching(true);
    FadeOut();
    FadeIn(3);
    AddTimer("T1", 3, "Intro");
    AddTimer("T2", 6, "Intro");
    AddTimer("T3", 8, "Intro");
    AddTimer("T4", 10, "Intro");
    AddTimer("T5", 12, "Intro");
}
void Intro(string &in asTimer)
{
    string x = asTimer;
```

```
if (x == "T1")
{
    PlaySoundAtEntity("", "react_sigh.snt", "Player", , false);
    FadeOut(3);
}
else if (x == "T2")
{
    FadeIn(3);
    PlaySoundAtEntity("", "react_breath.snt", "Player", , false);
    StartPlayerLookAt("ScriptArea_1", 2, 2, "");
}
else if (x == "T3")
{
    StopPlayerLookAt();
    StartPlayerLookAt("ScriptArea_2", 2, 2, "");
}
else if (x == "T4")
{
    PlaySoundAtEntity("", "react_breath_slow.snt", "Player", , false);
    StopPlayerLookAt();
}
else if (x == "T5")
{
    SetPlayerCrouching(false);
    SetPlayerActive(true);
}
}
```

Lets take this script apart. In "void OnStart()", the player can't move or look where he pleases, it is completely dark and will fade back in 3 seconds, and it adds 5 timers. These 5 timers have the local timer names of "T1" through "T5". They will be used to tell apart from each timer when used in the "Intro" function. In "void Intro(string &in asTimer)", a string variable x is equal to what the timer is. So when 3 seconds pass, the "Intro" timer function is called having the "asTimer" be "T1". That's why it has to be string x because it's a word or letter. It makes an "if" statement to see which timer called the function, so if the local timer name is "T1", it will play the sound and fade out the screen, reaching completely black in 3 seconds. The rest of them are done in the same way.

This advanced timer helps organize timers under one function instead of many.

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

Permanent link:
<https://wiki.frictionalgames.com/hpl2/tutorials/script/advancedtimers?rev=1310572887>

Last update: **2011/07/13 17:01**

