

# Level Editor - Useful tricks (in progress)

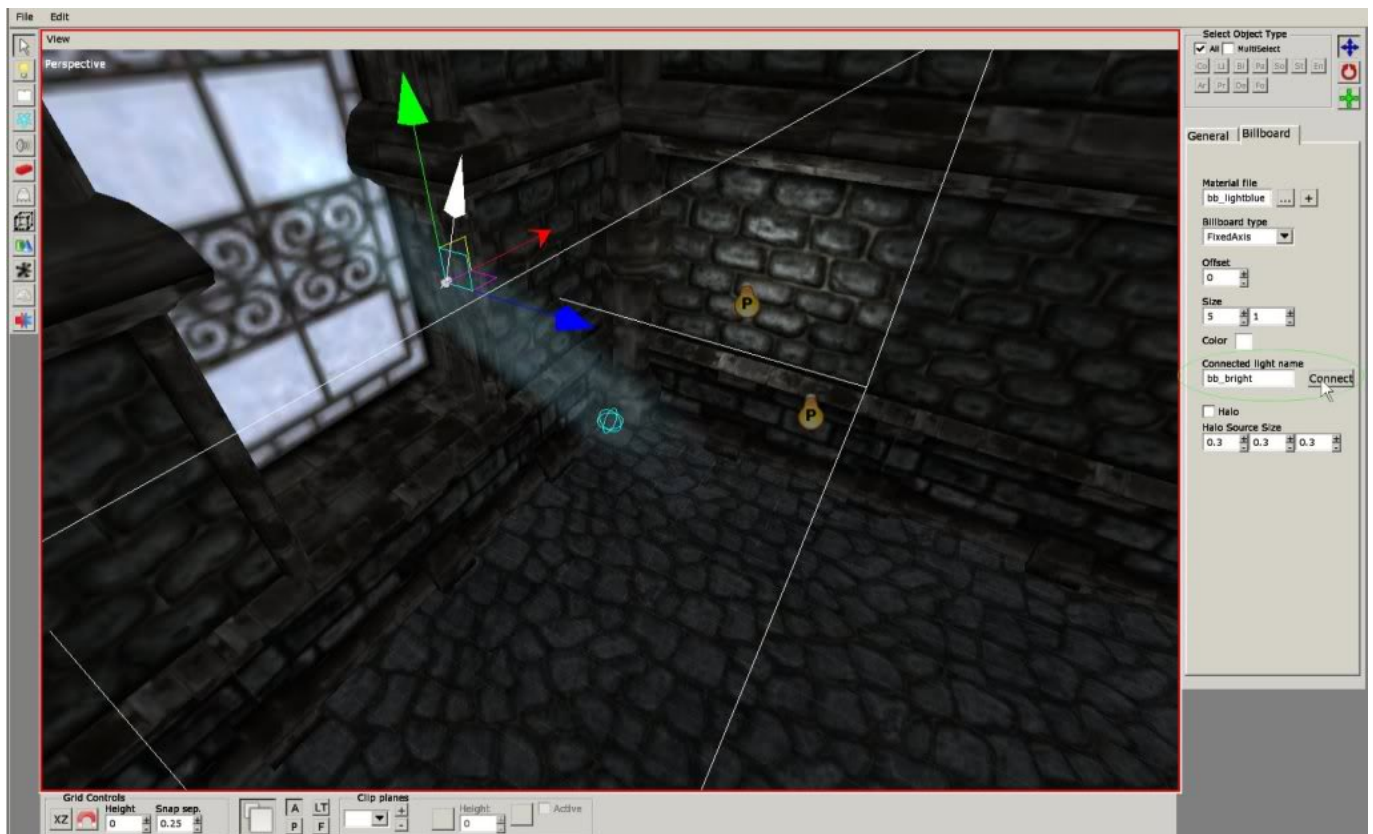
While The Basics and Light tutorial are great enough to create atmospheric maps, there are some things that can easily help achieve interesting results.

## Billboards

Step by step tutorial. Starting with easy way to setup and control billboards for your map, and continuing to more advanced effects.

First make sure you have read and understood [this](#).

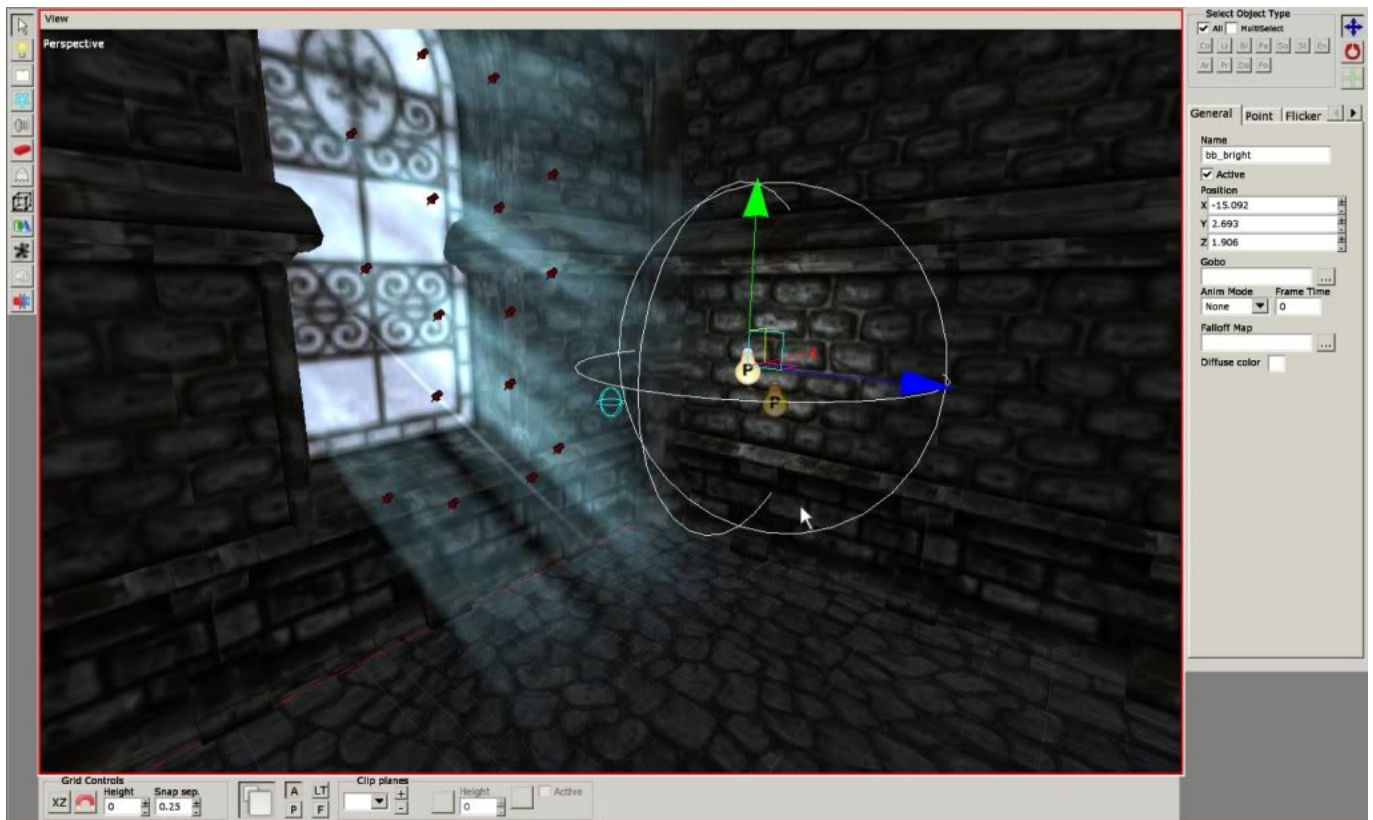
**Start off by creating a billboard and a point light and connect the billboard into the point light.** This helps you control the color value of every billboard you duplicate from the original connected billboard. Keep all of the billboards colors at 1 value. (That one extra point light in the picture is useless for the first parts)



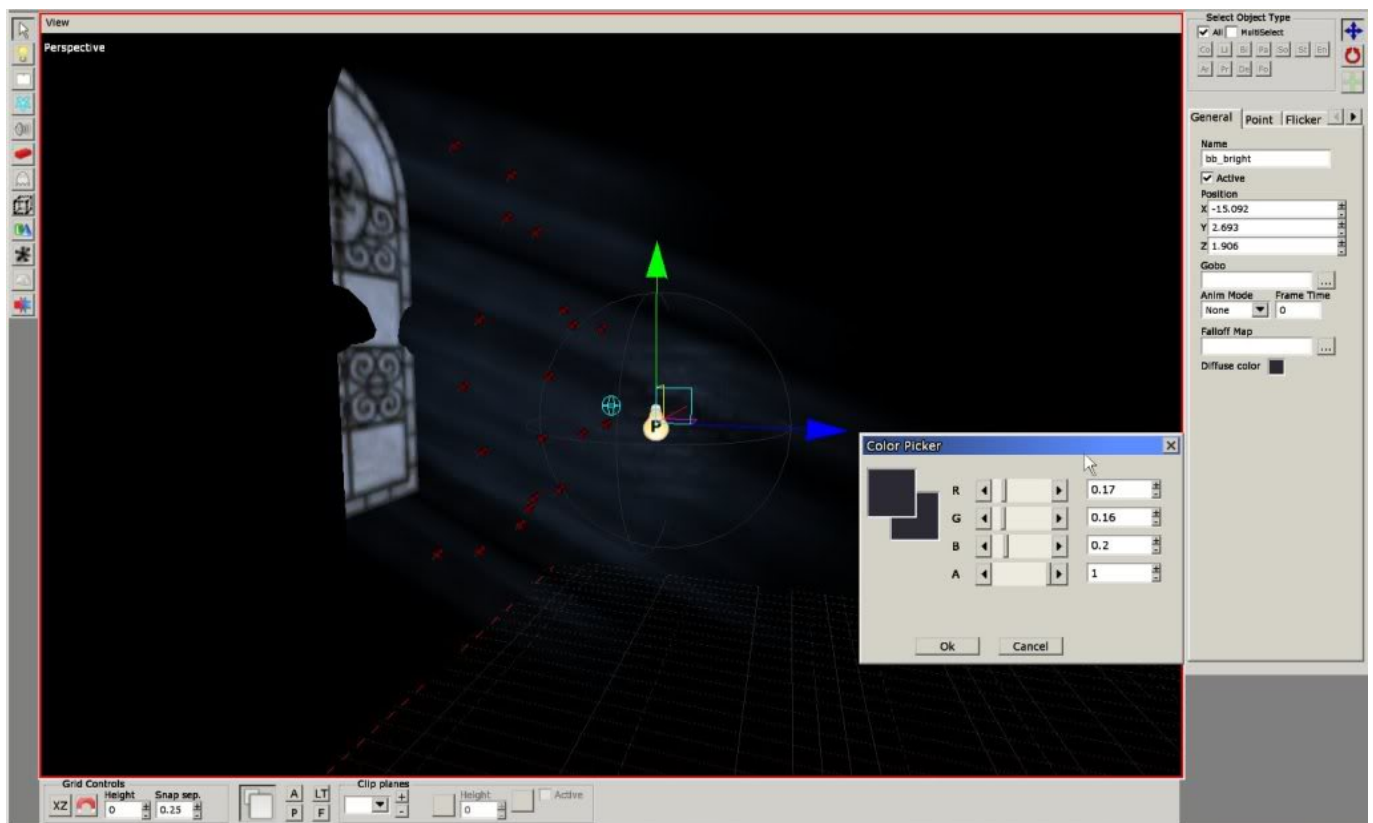
Then align your billboards properly into your window.

Last update:  
2011/10/05  
13:47

hpl2:tutorials:level\_editor:tutorial\_6 [https://wiki.frictionalgames.com/hpl2/tutorials/level\\_editor/tutorial\\_6?rev=1317818847](https://wiki.frictionalgames.com/hpl2/tutorials/level_editor/tutorial_6?rev=1317818847)

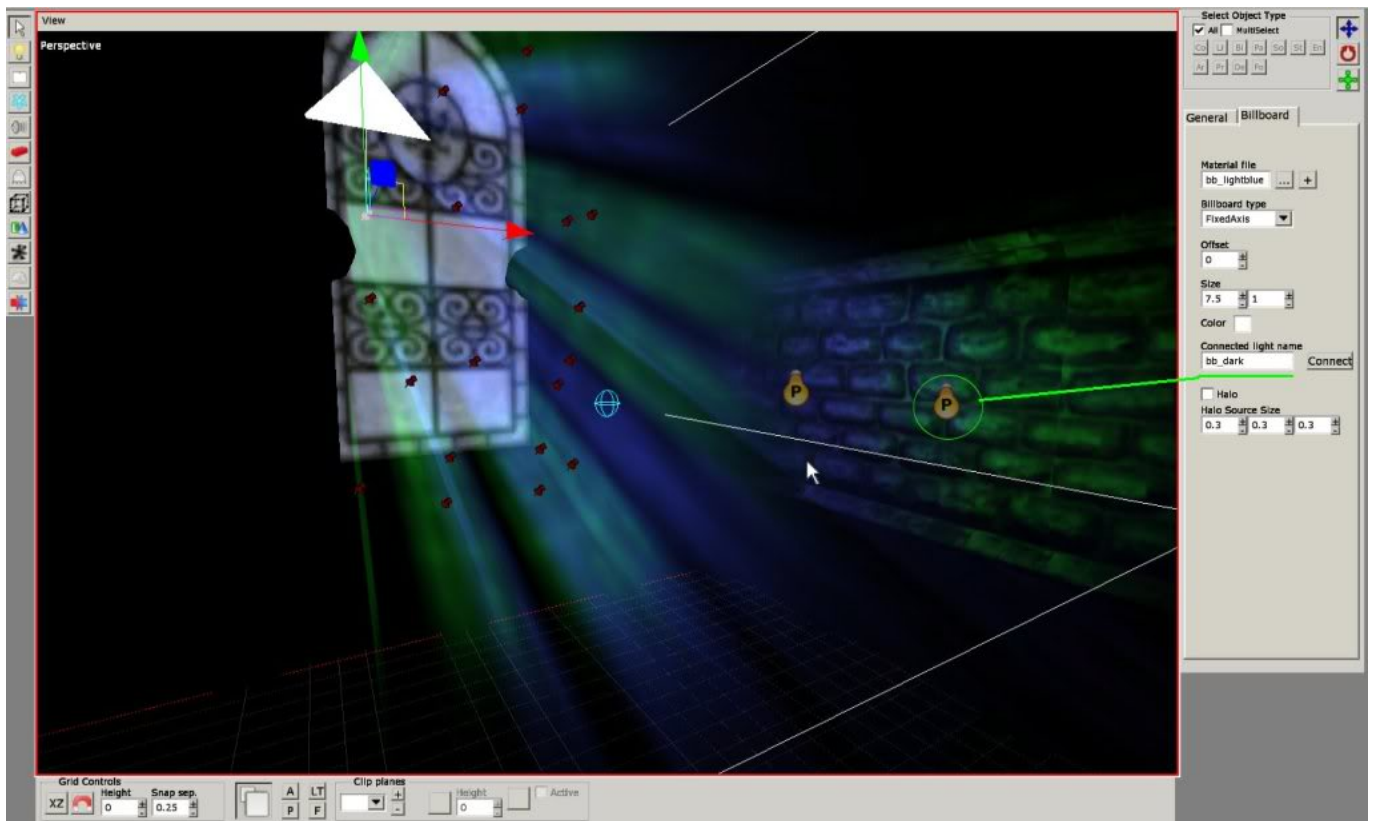


Then tinker with the color of the connected point light and adjust the length and width of the billboards. This is the hardest and time consuming part about billboards. Make sure to often check in game because billboards look different there and you can see if they look good from every angle.



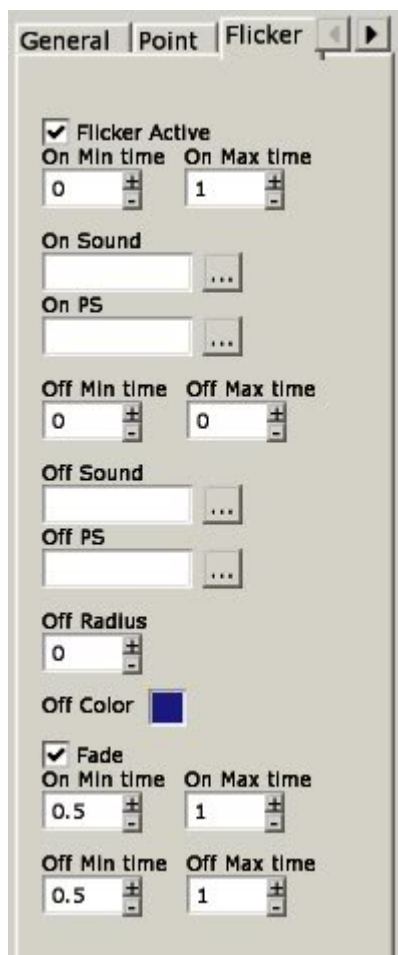
**That was it for simple billboards, but if you continue, you get to the interesting parts!**

To create better looking billboards, not all of them should be of the same color, so create another point light and connect the billboards that you think need different colors to other point lights. For simplicitys sake I keep them connected to only 2 point lights. (Colors changed so you can see better what I did)



## Flicker effect

You can create the flickering effect with various methods. First and easiest one would be to use the point lights own flickering settings inside the level editor. Tweak the Off Color and Diffuse Color to suit your liking. Example settings:



Using the level editor only is not as interesting as randomly changing colors of billboards and every light that acts as the windows shine in sync with script.

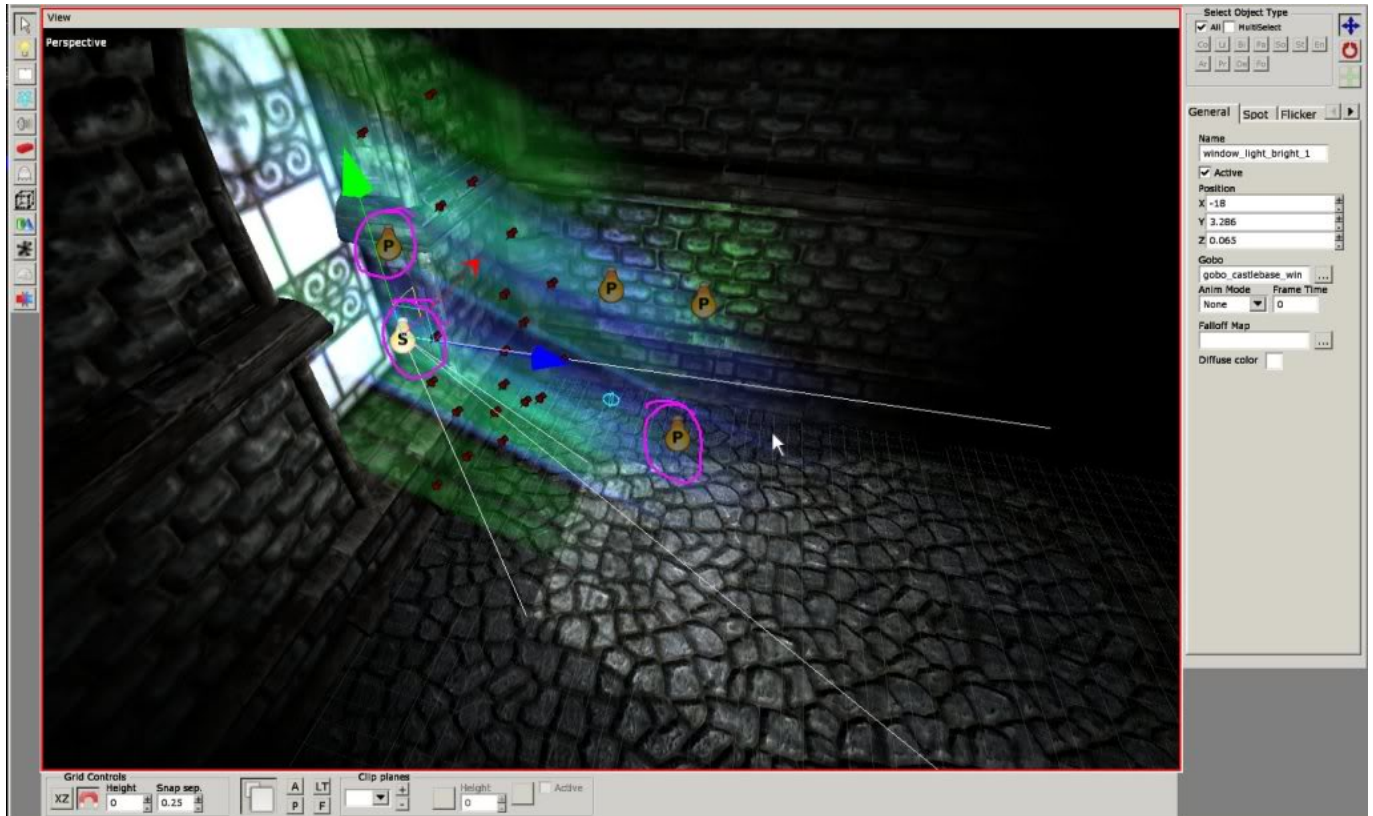
So next phase would be to set up additional lights to the window, spot light and some ambient point lights in this case. Name them so you can easily track what they are. In this case I named mine like this:

window\_light\_bright\_1 (spot light with gobo, I want this bright)

window\_light\_bright\_2 (point light close to window, bright also)

window\_light\_dark\_1 (large point light)

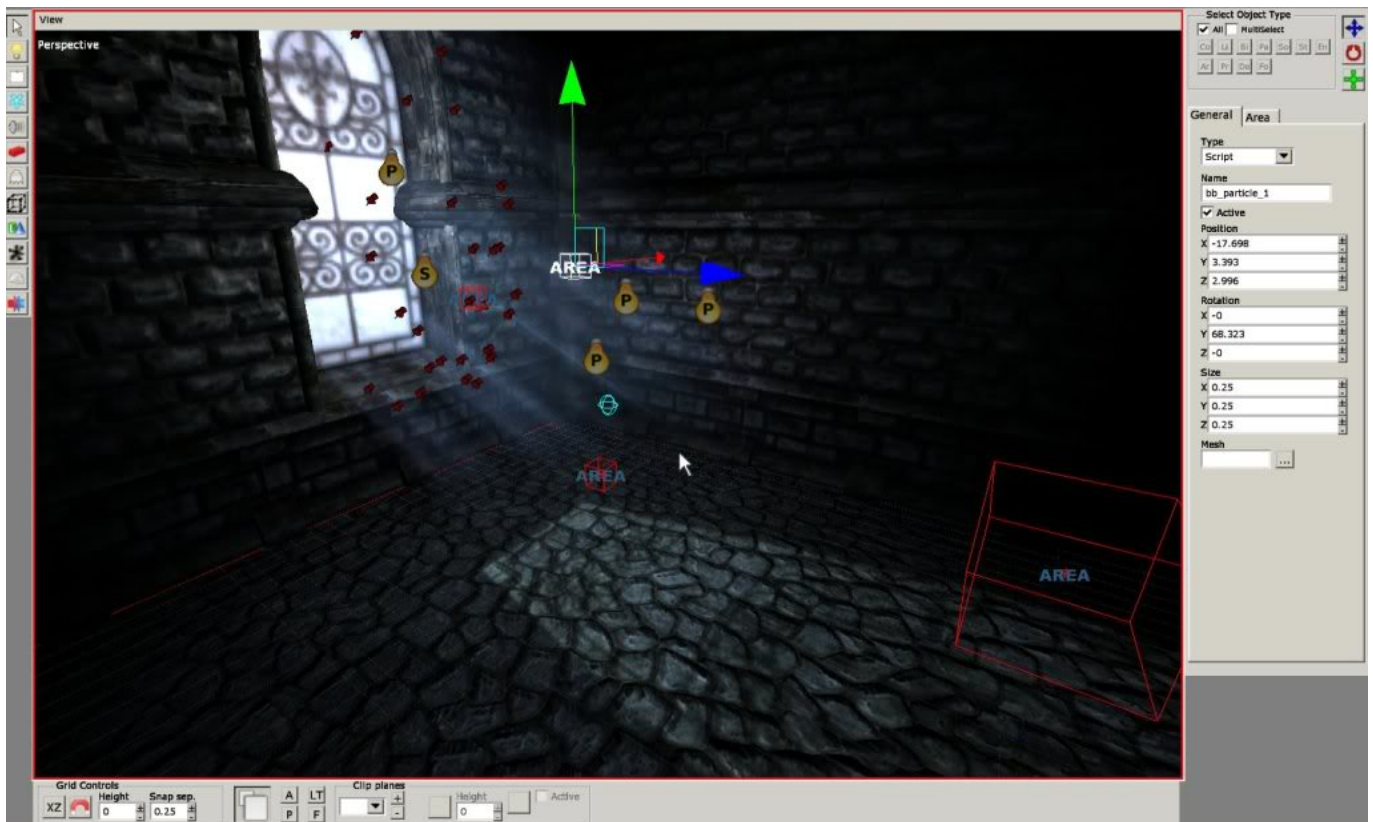




Then the rest is up to script. Example script for controlling, very simple vers. You can tweak the ideas around alot. Like this one does not change the color of the lights into same color as the billboards.

```
void FlickerTimer(string &in asTimer)
{
    float ftimer = RandFloat(0.6f,1.6f);
    FadeLightTo("bb_bright", RandFloat(0.2f,0.3f), RandFloat(0.2f,0.35f),
    RandFloat(0.3f,0.39f), 1, -1, ftimer);
    FadeLightTo("bb_dark", RandFloat(0.15f,0.29f), RandFloat(0.18f,0.35f),
    RandFloat(0.35f,0.39f), 1, -1, ftimer);
    for(int i=i<2;i++) FadeLightTo("window_light_bright_"+i,
    RandFloat(0.55f,0.7f), RandFloat(0.69f,0.8f), RandFloat(0.8f,0.9f), 1, -1,
    ftimer);
    for(int i=i<2;i++) FadeLightTo("window_light_dark_"+i,
    RandFloat(0.4f,0.5f), RandFloat(0.5f,0.6f), RandFloat(0.6f,0.7f), 1, -1,
    ftimer);
    AddTimer("repeat", ftimer, "FlickerTimer");
}
```

Then here is the one I used for the demonstration video clip. (Can cause performance issues when done with lots of billboards and particles, I haven't tested...). Minor tweaking, added spawn areas for particles and finetuned the script for long. The script could be done more efficient, if someone can, please edit!



```
void OnStart()
```

```
{
```

```
AddTimer("start", , "FlickerTimer");
```

```
SetPlayerMoveSpeedMul(0.7f);
```

```
SetLocalVarInt("EventInt", 1);
```

```
for(int i=i<3;i++) CreateParticleSystemAtEntityExt("blue_particles",
"ps_light_dust_large.ps", "bb_particle_"+i, true, 0.6f, 0.75f, 1.0f, 1,
true, 1, 2, 8, 11);
```

```
for(int i=i<3;i++) CreateParticleSystemAtEntityExt("blue_fog_particles",
"ps_area_fog.ps", "fog_particle_"+i, true, 0.6f, 0.75f, 1.0f, 0.8f, true, 1,
2, 8, 11);
}
```

```
void FlickerTimer(string &in asTimer)
```

```
{
```

```
SetLocalVarFloat("CheckBrightness", RandFloat(0.20f,0.32f));
```

```
float fMainColor = GetLocalVarFloat("CheckBrightness");
```

```
float fBloodColor = RandFloat(0.3f,0.45f);
```

```
float ftimer = RandFloat(0.4f,0.7f);
```

```
switch(GetLocalVarInt("EventInt"))
```

```
{case 1: //This part loops forever when EventInt is 1
```

```
    FadeLightTo("bb_bright", fMainColor+0.08f, fMainColor+0.05,
```

```

fMainColor+0.1, 1, -1, ftimer);
    FadeLightTo("bb_dark", fMainColor+0.01f, fMainColor, fMainColor+0.04, 1,
-1, ftimer);
if(GetLocalVarFloat("CheckBrightness") <0.26f){
    for(int i=;i<2;i++) FadeLightTo("window_light_bright_"+i,
fMainColor+0.22f, fMainColor+0.27f, fMainColor+0.43f, 1, -1, ftimer);
    for(int i=;i<2;i++) FadeLightTo("window_light_dark_"+i,
fMainColor+0.13f, fMainColor+0.19f, fMainColor+0.24f, 1, -1, ftimer);
}
if(GetLocalVarFloat("CheckBrightness")>= 0.26f){
    for(int i=;i<2;i++) FadeLightTo("window_light_bright_"+i,
fMainColor+0.31f, fMainColor+0.36f, fMainColor+0.58f, 1, -1, ftimer);
    for(int i=;i<2;i++) FadeLightTo("window_light_dark_"+i,
fMainColor+0.2f, fMainColor+0.27f, fMainColor+0.3f, 1, -1, ftimer);
}
    break;
case 2: //Changes colors to red when EventInt is 2
    FadeLightTo("bb_bright", fBloodColor+0.08f, fBloodColor-0.3f,
fBloodColor-0.3f, 1, -1, ftimer+0.3f);
    FadeLightTo("bb_dark", fBloodColor+0.02f, fBloodColor-0.3f,
fBloodColor-0.3f, 1, -1, ftimer+0.3f);
    for(int i=;i<2;i++) FadeLightTo("window_light_bright_"+i, fBloodColor+0.28f,
fBloodColor-0.3f, fBloodColor-0.3f, 1, -1, ftimer+0.5f);
    for(int i=;i<2;i++) FadeLightTo("window_light_dark_"+i, fBloodColor+0.15f,
fBloodColor-0.3f, fBloodColor-0.3f, 1, -1, ftimer+0.5f);
    break;
}
AddTimer("repeat", ftimer, "FlickerTimer");
}

void Touched(string &in asEntity)
{SetLocalVarInt("EventInt", 2);
StartScreenShake(0.01f, 0.9f, 0.2f, 0.2f);
//destroyParticles
DestroyParticleSystem("blue_particles");
DestroyParticleSystem("blue_fog_particles");
//create new ones
for(int i=;i<3;i++) CreateParticleSystemAtEntityExt("red_particles",
"ps_light_dust_large.ps", "bb_particle_"+i, true, 1.0f, 0.1f, 0.1f, 1, true,
1, 2, 8, 11);
for(int i=;i<3;i++) CreateParticleSystemAtEntityExt("red_particles",
"ps_area_fog.ps", "fog_particle_"+i, true, 1.0f, 0.1f, 0.1f, 0.9f, true, 1,
2, 8, 11);
}

```

Last update:

2011/10/05

13:47

hpl2:tutorials:level\_editor:tutorial\_6 [https://wiki.frictionalgames.com/hpl2/tutorials/level\\_editor/tutorial\\_6?rev=1317818847](https://wiki.frictionalgames.com/hpl2/tutorials/level_editor/tutorial_6?rev=1317818847)

---

From:

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

Permanent link:

[https://wiki.frictionalgames.com/hpl2/tutorials/level\\_editor/tutorial\\_6?rev=1317818847](https://wiki.frictionalgames.com/hpl2/tutorials/level_editor/tutorial_6?rev=1317818847)

Last update: **2011/10/05 13:47**

