BO3 | Purchase Loops

It is suggested that if you are familiar with World at War scripting and/or Black Ops scripting, that you check out the BO3 | What's changed from WaW scripting?

Overview

This tutorial will show the steps to setting up a Custom Script to include a "Purchase Loop" which can be implemented to allow traps, wallbuys, and other custom scripts to have a way of purchasing the feature.

Including _zm_score.gsc

The _zm_score.gsc contains the function minus_to_player_score, which will allow the user to remove points from a given player as well as display the change of points on the HUD.

Since this script is not a part of your specific script, your must include the using line for this file so the script as access to minus_to_player_score.

```
#using scripts\zm\_zm_score; // Add this line at the top of your script
```

Trigger

To check if the player is press the use key on a trigger, you will have to create a trigger_use in radiant and give it a kvp of targetname with your own name for the value (this tutorial named the trigger "test_trigger").

The script will get the trigger using the GetEnt function as well as use the waittill function to wait until the trigger has been activated. Notice how there is a variable in the waittill function named player;

the player variable here is being defined as the entity that activated the trigger. This is important, since it can be used to minus the score from the player later.

```
function testfunction(parameter)
{
	trigger = GetEnt("test_trigger", "targetname");
	trigger waittill("trigger", player);
}
```

If Statement

Checking if the player has enough points to actually purchase the feature requires an If Statement. This will allow the script to compare the integer value of the player's score to the cost of the feature.

This cost has to be set before hand as its own variable. As for the player's score, it is defined as a variable on the player - player.score

```
function testfunction(parameter)
{
	trigger = GetEnt("test_trigger", "targetname");
	cost = 1000;

	trigger waittill("trigger", player);

	if(player.score >= cost) // checking if the player's score is greater than or equal to the cost
	{
		// Code to perform if the player has enough points
	}
}
```

While Loops

The script will require 2 While Loops to achieve the logic for the Purchase Loop. The first While Loop is to allow the script to perform your code multiple times such as a Trap or purchasing an Item more than once.

The second While Loop is inset into the first one to allow it to be used multiple times. This second While Loop will loop until the player has enough points to activate the trigger.

Note: while loops require a form of "waiting" so that it doesn't loop multiple times on the same frame. This can be in the form of a waittill or wait function.

```
function testfunction(parameter)
{
	trigger = GetEnt("test_trigger", "targetname");
	cost = 1000;
	coolDownTime = 30; // time in seconds before the trigger can be activated again

	while(1)
	{
		while(1)
		{
			// Purchase Code
			trigger waittill("trigger", player);

			if(player.score >= cost)
			{
				// Code
			}
		}

		// Your Code
		wait(coolDownTime);
	}
}
```

Purchase Code

The final step is to allow the inset While Loop to break (meaning it will allow the code to continue outside of the loop) so if the player does have enough points it will perform the rest of the code.

Using the player variable you can now call the function from _zm_score.gsc to minus the points from the player. Also adding the break keyword to the if statement will allow for the code to break out of the most immediate loop and continue the code below.

```
function testfunction(parameter)
{
	trigger = GetEnt("test_trigger", "targetname");
	cost = 1000;
	coolDownTime = 30;

	while(1)
	{
		while(1)
		{
			// Purchase Code
			trigger waittill("trigger", player);

			if(player.score >= cost)
			{
				player zm_score::minus_to_player_score(cost);
					break;
			}
		}

		// Your Code
		wait(coolDownTime);
	}
}
```

Sounds

An extra step would be to include sounds for the purchase loop. When the player purchases the feature using a sounds like a "cha-ching" will solidify the purchase, as well as doing a deny sound for when they don't have enough points.

These sounds are played on the trigger, but can also be played on the player. Note the names "name_of_purchase_sound" and "name_of_deny_sound" are just made up alias names. If you want to include your own sounds or know some in game aliases then you can include those.

```
function testfunction(parameter)
{
	trigger = GetEnt("test_trigger", "targetname");
	cost = 1000;
	coolDownTime = 30;

	while(1)
	{
		while(1)
		{
			// Purchase Code
			trigger waittill("trigger", player);

			if(player.score >= cost)
			{
				player zm_score::minus_to_player_score(cost);
				trigger playsound("name_of_purchase_sound");
				break;
			}
			else
			{
				trigger playsound("name_of_deny_sound");
			}
		}

		// Your Code
		wait(coolDownTime);
	}
}
```