

Welcome back folks ☺ glad to see you enjoy the basics of game programming. Let's continue with this fun stuff.

The use of the if command.

Remember the variable we made? we will try to use this to play a sound. First we need to define the sound as we learned before. We use the file bonus.wav.

So place the following line above the main function:

```
SOUND* bonus_snd = "bonus.wav";//// « define the sound we will use
```

Then in our pickup_ammo action we will add these 3 lines.

```
if(ammo_count == 30){////« so if the ammo number is 30 it must do the sound beam play wait(-2); //«« wait 2 seconds before playing the sound. snd_play (beam_snd, 100, 0);////«< play the sound file we defined
```

if starts always with a comparison and then followed by instructions what to do. In this case on the number 30 it will play the sound.

Here are the comparison used in the programming with Lite-C

==	« when the number should be exactly the same
>	« when the number is higher
>=	« when the number is higher or the same
<	« when the number is lower
<=	« when the number is lower or the same
!=	«when the number is not the same and so is not equal

We call these also operators. This is used very much as it is simple to use and activate actions or functions when needed.

Your code should now be: action pickup_ammo()

```
action pickup_ammo()
{
set(my,SHADOW | PASSABLE);// set the model passable and give it shadow
wait(1); //wait one frame
c_setminmax(my);// set proper collision BBOX
while(!viper) //wait till the player is created the hereiam action
{
wait(1);// wait one frame
}
while(vec_dist(my.x,viper.x) >50)// the distance between the viper and the pickup object
{
wait(1);/// wait one frame
}
ammo_count +=10;// yes we add 10 points to our variable. Use - if you want to subtract.
snd_play (ammo_snd, 100, 0);////"< play the sound file we defined
ent_remove(my);// we remove the pickup model
if(ammo_count == 30){////" so if the ammo number is 30 it must do the sound beam play
wait(-2); //" wait 2 seconds before playing the sound.
snd_play (bonus_snd, 100, 0);////"< play the sound file we defined
}
}
```

Run it and if the number hits 30 of the ammo_count variable it will play the beam sound after 2 seconds. Yes. We learned another important part of programming.

This is cool Realspawn but now another cool thing we need. Can we play music during the game?
If so how?

Ah you want to spice things up well let's do this then. But we do it the Realspawn way so pay attention.

Using ingame music. Handle.Media plav.pause.stop

I was kind enough to create an instrumental song (yes i do make music) The file is called: ingame.wav A simple way to make a music file loop during the full game is this.

Add this line to your main function :

```
media_loop("ingame.wav", NULL, 30); // « start the song and sets the volume on 100. It will loop.
```

Run the level and there you have your music. This is the easy way but it will be hard to control. If you want to be able to control the song like pausing it or stopping it we have to do it in another way. Here is how.

First we create a variable for the song (Remember the variables ? well here it is again)
place this code above your main function

```
var soundtrack_handle;
```

Then replace the line in the main function with this line :

```
soundtrack_handle = media_loop("ingame.wav", NULL, 100);
```

Run it to see if it works. Next chapter you will learn how to stop the soundtrack and control it.

How to use key's to control the music and the use of functions.

Next thing we do is adding a function. A function is nothing more than a collection of lite-C commands that use variables and stuff like that. They can be called whenever needed. We also create a new variable for the pause function. (See again we are using variables.)

We will create a function to pause the music on press M key and resume it when we press M again.

Add this function above the main function

```
var pause_music;
function pause()//""the name of the function
{
if(pause_music == 2) //" if our variable is not pressed
{
media_start(soundtrack_handle);//"" it resumes playing our variable (the song)
pause_music = 0; // " set variable to 0
return;
}
Else//<when the m key is pressed
{
pause_music = 2; // pause //""< when the M key is pressed
media_pause(soundtrack_handle); //""then we pause our variable (the song)
}
}
```

In our main function we add this line :

`on_m = pause;`

Now on the M key press you will call the pause music function. So you know now also how to make a function or action work on any key press. You can change it into any key you want. See if you can replace it for another key and see if it works.

When we run the level we can now pause the music by pressing the M key. If we press it again the music resumes. Good work people slowly but surely we are learning and using the knowledge we gathered these first 2 workshops.

Panels (graphics) on screen.

Tjeez we did a lot and so many questions left like how do we add graphics to our game like a HUD or message panels? Well let's see how we use this and how to handle it all. With the sounds we needed to define them first, we will do the same with our graphics. In this workshop we will use the picture gm.pcx
Add these lines above your main and your pause function:

```
BMAP* gm_map = "gm.pcx";///  
PANEL* gmp_pan = ///  
{  
    bmap = gm_map; // calling our picture.  
    pos_x = 10; ///  
    pos_y = 50; ///  
    layer = 10; ///  
    flags = SHOW; ///  
}
```

Run your level and you will see the picture on screen. Note that the black is also visible. If you want to remove all the black add OVERLAY to the flags line this way

```
flags = SHOW | OVERLAY; ///  

```

Run it again and you will see the picture again but without the black parts.

This is the way you can add your graphics to your game. Experiment with the positions see if you can add graphics yourself and try it all out. You will need this in the future.

Making panels visible and invisible on key press.

The first thing I need you to do is remove the flags show line from our picture. This way when we run the level the picture is not shown. Remember the function we created to pause our music and restart it? Aaaahhh that's why you gave us a picture with the text music is paused. Yup you got me let's see how to use this.

We are going to use these 2 lines of code ;

```
set(gmp_pan,SHOW | OVERLAY); ///  
reset(gmp_pan,SHOW | OVERLAY); ///  

```

We insert these 2 lines in our pause music code like this :

```
function pause()//""the name of the function
{
if(pause_music == 2) //"" if our variable is not pressed
{
reset(gm_pan,SHOW | OVERLAY); //""< to turn the panel on and make it visible
media_start(soundtrack_handle);//"" it resumes playing our variable (the song)
pause_music = 0; // "" set variable to 0
return;
}
else
{
pause_music = 2; // pause //""< when the M key is pressed
set(gm_pan,SHOW | OVERLAY); //""< to turn the panel on and make it visible
media_pause(soundtrack_handle); //""then we pause our variable (the song)
}
}
```

Run it and press M to pause the music. Yesssss the picture is shown and on the second key press it is invisible again. Amazing stuff isn't it ? Well we learned a lot for now here is some advice what you could try out. Create your own graphics and placed them on your screen Create a new function to make a screen visible and invisible See if you can use M for pausing music and another key for you graphic on/off Have fun I'll see you next time.

René Pol aka Realspawn.

Realspawn@live.nl