

Workshop 00

Welcome to the first of a new workshop series I created for non coders like me.

First for those that don't know me, my name is René Pol and I live in the Netherlands. I been toying with 3D Gamestudio from A5 times and I always enjoyed trying to make things easier to understand and share it with others. Some people remember WDL Gold? a collection with codes. Well yes that was me to. I travel under the name Realspawn and although people think sometimes I am an expert coder I am definitely not. What I do is make simple short examples with a lot of help from forum friends. These simple examples help people that are new and want to explore their own creativity.

Although people try hard to make good manuals and tutorial workshops, I noticed they are never really complete or have a sense of creativity. Most of the time its dry math material and codes are never fully explained. I tried to redo the original online workshops in my own Realspawn way. This means I show what it all does and make you test it out yourself. Seeing things actually happen is easier to learn and understand from is my point of view. A non coder like me should get the feeling he or she is actually creating something while learning. Also I tried to insert stuff that is asked most of the time in the forum but never really answered to. You see half-baked answers or stuff like: look in the manual.

For example how to control media during game play? Well it should all become clear now as I finished the first 9 workshops. In part 08 you should be able to create a small fun game with all you have learned in the previous workshops. I made sure I commented al coded lines so you see what it does and you can change things to see what impact it has. Professional coders know what they are talking about and are sometimes too focused on explaining the math. I know it's important but it motivates if you actually see what you and your code is doing.

Now don't think I can make you a great programmer because I simply can't. Hell even I learned all this stuff while I am making these workshops. But when you know all the basics you have a firm start to let your creativity do the rest. Make sure you read al and test all. Don't try to do more until you understand all you have learned. Most project fail because people want to do too much. The game in workshop 8 is simple and could have more fun things in it but from there you can expand your knowledge as all you learned in the first workshops is right in there. So it's important to take all step by step and don't rush thing. I have faith in you all that you will succeed. Trust me this 41 year old dude is just beginning to learn how to program and I am sharing this information gladly with you all.

Special thanks and credits are going out to :

- George Pirvu for his wonderful work on the AUM magazines and the original online tutorials.
- Kartoffel for his box models I use and his helping solve my problems I posted in the forum.
- Devon helping solve my problems I posted in the forum.
- 3dgs_snake helping solve my problems I posted in the forum.
- Superku helping solve my problems I posted in the forum.
- Rayp helping solve my problems I posted in the forum.
- 3Run helping solve my problems I posted in the forum.
- Malice helping solve my problems I posted in the forum.
- Joozy for his space-cube I have used in this workshop.

And all I might forgot. Please send me a mail so I can name you in the updates.

I know you are motivated and want to start otherwise you would not have taken the time to download the workshops right?

You should make yourself familiar with 3D game studio 8A. I will not explain how to build levels as there is enough good material to find. I will give you some links at the end of this chapter you can explore. For now I will try to explain the basics of creating a new script (programming) for your own level.

First things first.

You have installed a free or full Game studio A8 version on your computer; the program should have an entry in Windows' start menu. In the program group you can see shortcuts to the Lite-C Script editor (**SED**), to the Model and Terrain editor (**MED**), to the level editor (**WED**), and to the **manual**. Let's see what they do:

*The script editor (SED) is the program that helps us write, execute, compile and debug our lines of lite-C code.

*The model editor (MED) which comes with lite-C allows us to import or create 3D objects (models).

*The world editor (WED) is the program that you use to build levels to your own ideas.

When you built a simple level (as a test) you can run it and fly through it with the arrow keys and page up and down. A start camera position can be used if you place a position in your level.

If you start a new project it's important you create a good work folder that has all stuff organized so you won't have to search every time for things you need.

So here is an idea what you can do best:

- Create a workfolder (in this case workshop 00)
- Create in this folder sub folders for all the different parts you use for the game creations
- Example I use : subfolders : 3dmodels / scripts / sounds / graphics
- So now you can put all sound files in the sound folder and your graphics in the graphics folder
- You can always use your own subfolders but make sure it's neat and tight so things stay organized.

Ok so you built this simple level (I used one big flat block for an example) save this into your workfolder under the name you want. In my case I saved it as workshop00. Make sure the level is built before saving it.

Now to give our level a clean C-lite script (That we use to program) we do this:

- Click file (After you have opened your level)
- Go to map properties (a small window will open)
- Choose New script (the second button)
- Choose Lite_C_Project (then close this window)

We now added a clean script for our level let's see what it has created shall we?

In wed go to the left part of the screen en select the resources tab.

Choose script files and you will see the workshop 00 script. Click it 2 times to open up. You will see that the script editor (SED) will open so you can edit your script and start programming ☺

This basic clean script has only these lines:

```
////////////////////////////////////
#include <acknex.h>////<<<< scripts that come with 3d gamestudio
#include <default.c>
#include <mtlFX.c>
////////////////////////////////////
function main()////<<<< important the main function
{
    level_load("workshop00.WMB"); ///<<<<< loads your level
}
```

The include lines are template scripts that come with 3d gamestudio A8.

Next is the main function. The main is important as it is always used to start up all things needed to run your game the way you want. You see it will load our level but nothing else.

Remember that we made a workfolder? right so we need the engine to know where to find models, music, graphics in order to use them. What we do is we will include these subfolders by adding a few lines inside the include lines.

```
////////////////////////////////////
#include <acknex.h>////<<<< scripts that come with 3d gamestudio
#include <default.c>
#include <mtlFX.c>
////////////////////////////////////
#define PRAGMA_PATH "3dmodels" ///<< the path to our 3d models folder
#define PRAGMA_PATH "graphics" ///<< the path to our graphics folder
#define PRAGMA_PATH "scripts" ///<< the path to our scripts folder
#define PRAGMA_PATH "sounds"///<< the path to our sound models folder

function main()////<<<< important the main function
{
    level_load("workshop00.WMB"); ///<<<<< loads your level
}
```

Save your script. Open your level again in wed and ad the ship model ☺ that's right in the 3dmodels folder. Run it and you see it will show up. (Built first with entities selected)

Now delete this model from your level as we will try something else now.

In the main function we will add one line of code that imports the ship model and places it in the level where we want it to be. We will add this line to our main function

```
ent_create("ship.mdl", vector(0, 0, 0), NULL);
```

Let's break this line in parts and see what it does.

Ent_create	= Create an entity (3d model or terrain or graphic)
"ship.mdl"	= Import the ship from our 3d subfolder. MDL is the format of 3d objects 3dgs uses.
Vector 0,0,0	= The place (vector) where the ship model should appear. X-axis,y-axis and z-axis that's right in that order and those 3 numbers. You can give it any coordinates you want.
NULL	= The ship has no action and will do absolutely nothing. Don't worry in next workshops you will learn to create your own actions.

Your main code should look like this now:

```
function main()  
{  
  level_load("workshop00.WMB");  
  ent_create("ship.mdl", vector(0,0, 0), NULL);  
}
```

Run the level and you will see the ship in the level. Change the coordinates to change its position or try to load other objects this way.

You now understand how to work and what the basic scripts contains. Also you already learned one line of coding how to make an object load and place itself in the level.

Well done for now. Curious in what more you can do? then I will see you in the next workshop.

Till then

René Pol aka Realspawn
Realspawn@live.nl

Here are some links to clips that will show you how to build a level in WED. Please mail me if any of this links no longer is present so I can remove it with the next update.

- *<http://www.youtube.com/watch?v=RfveYw9dfUc>
- *<http://www.youtube.com/watch?v=VMW8am7n15o>
- *<http://www.youtube.com/watch?v=sbwy5hoshJQ>
- *<http://www.youtube.com/watch?v=Y0DIasSWP3o>

Make yourself familiar with building a level, placing models, sprites and using all of WED's tricks like scaling, Subtract, rotating and the use of textures in your projects.



