3D2Sprite Help Document
by GameTemplates

Website: https://gametemplates.itch.io/

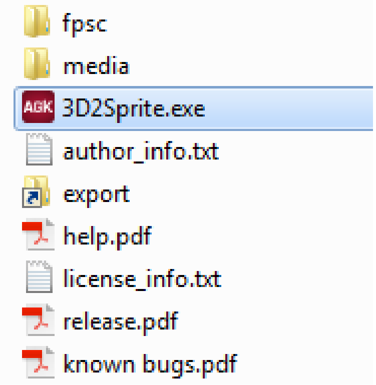
Table of Contents

1. Launch 3D2Sprite:

First you need to download the latest version of the software from here:
http://3d2sprite.blogspot.com

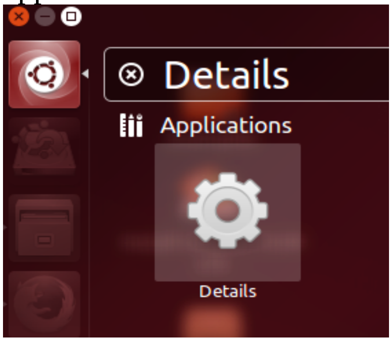
Once you have the zip file on your computer, you need to extract the files. I do not wish to get in to details of how to extract files from an archive, if you have difficulties please use google to find details about how to extract an archive.

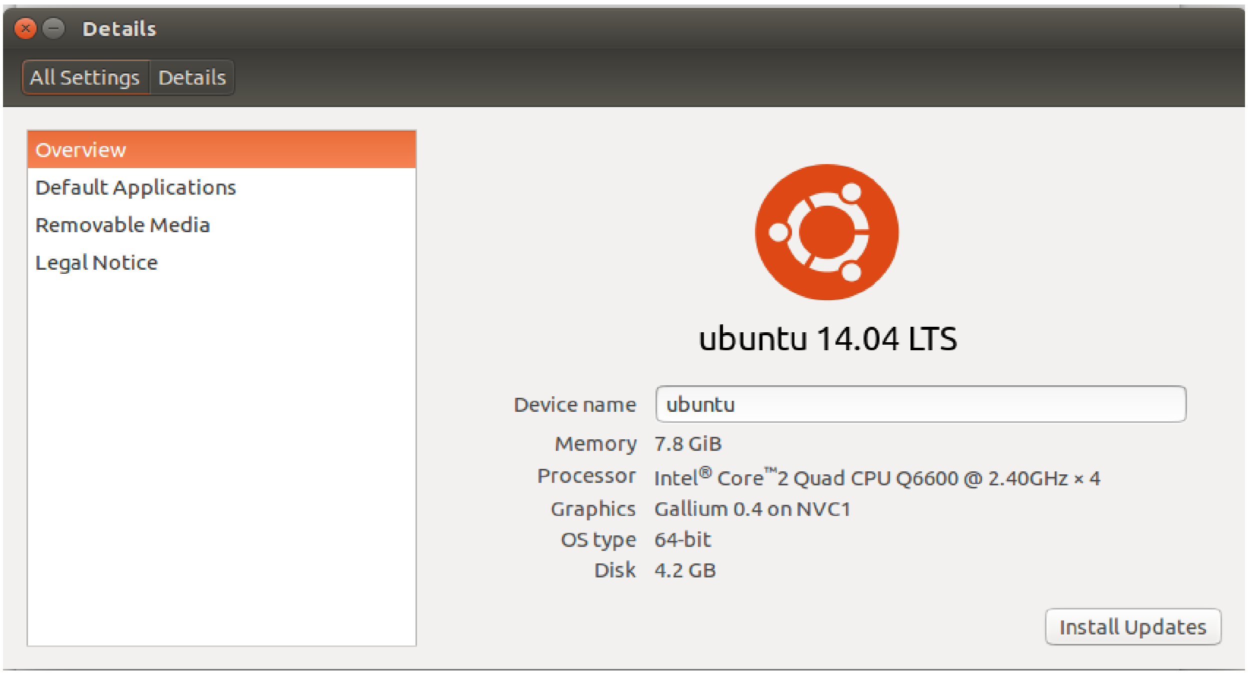
Once you have extracted the files, open the 3D2Sprite directory, and run the 3D2Sprite application launcher.



fpsc
media
3D2Sprite.exe
author_info.txt
export
help.pdf
license_info.txt
release.pdf
known bugs.pdf

On Ubuntu, you may going to find 2 application launcher. One for 32bit systems and one for 64bit systems. In case you don't know if your Ubuntu is 32bit or 64bit version, you can run the Details application to find the details of your system the one you are interested in is the OS type:





ubuntu 14.04 LTS

Device name ubuntu
Memory 7.8 GiB
Processor Intel® Core™2 Quad CPU Q6600 @ 2.40GHz × 4
Graphics Gallium 0.4 on NVC1
OS type 64-bit
Disk 4.2 GB

2. Load a 3D mesh:

To load a 3D mesh in to the software, use left mouse click on the first icon in the top left corner.

In the drop down menu, we are going to see 5 options to choose from:

- Load Parent mesh
- Load Child mesh
- Load FPSC Level
- Remove Child mesh
- Add point light

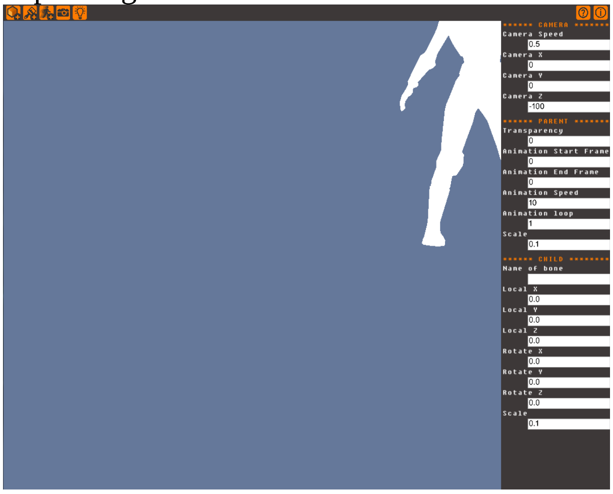
A Parent mesh can be an animated 3D mesh, for example a character. A child mesh can be a static 3D mesh, for example a weapon that we can assign to a bone of the parent mesh to follow it movement during animation. This way we can add the weapon (child mesh) to the hands of the character (parent mesh).

FPSC Level is a level made in FPS Creator, I'm going to talk about this option later, for now let focusing on loading a parent and child mesh

In order to load a Child mesh, first we need to load a Parent mesh with animations and the Child mesh must be a separated mesh file and not just a 'child' in hierarchy of the Parent mesh.

To load a Parent mesh, select the first option, to load a Child mesh, select the second option from the menu.
In the next window we need to browse our file system on the computer to find the 3D mesh we wish to load. Currently, 3D2Sprite supports the following formats:
.fbx .3ds .dae .obj .x .blend .lwo (technically many others also supported by the library but not tested)

Once we have loaded a 3D mesh, we should see it in the 3D window. It may not fully visible at first depending on the size of our mesh and also where the origin point of the mesh is.



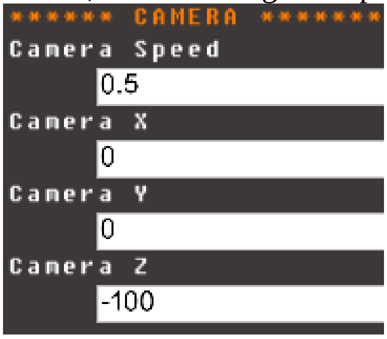
Camera Speed 0.5
Camera X 0
Camera Y 0
Camera Z -100

Transparency 0
Animation Start Frame 0
Animation End Frame 0
Animation Speed 10
Animation loop 1
Scale 0.1

Name of bone
Local X 0.0
Local Y 0.0
Local Z 0.0
Rotate X 0.0
Rotate Y 0.0
Rotate Z 0.0
Scale 0.1

3. Navigation in the 3D window:

Once we have a 3D mesh loaded, we can see it or at least some part of it in the 3D window. To navigate in the 3D window we can hold the Right mouse button and drag the mouse to look around. We can use the W,A,S,D keys to move forward, backward left and right. We can use the Q,E keys to move up and down. We can use the Arrow keys to move forward, backward, left and right in global space regardless the actual rotation of the camera and finally we can use the R key to reset the camera angle to look straight in global space. In case we would like to move around a bit faster or slower, we can change the speed of the camera on the right hand side.



****** CAMERA *******

Camera Speed

0.5

Camera X

0

Camera Y

0

Camera Z

-100

The speed can be as low as 0.01 or even less, and as high as we wish even 100 or 1000. We can also see the Camera X,Y,Z position in the 3D space, we can change it to whatever we'd like in case we need or just take a note of the current position of the camera.
If we look further we can see more options on the right side but I'm going to talk about it later.

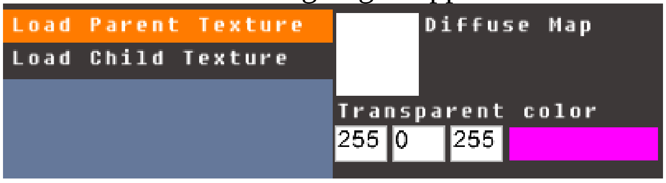
4. Load a Texture:

To load a texture for our 3D mesh, we need to click on the next icon in the top.

In the drop down menu, we are going to see 2 options to choose from:

- Load Parent Texture
- Load Child Texture

By using the first option, we can load a texture file for the parent object and by using the second option we can load a texture file for the child object. Just move the cursor over the option, and a texture selection menu going to appear on the screen.



Load Parent Texture

Load Child Texture

Diffuse Map

Transparent color

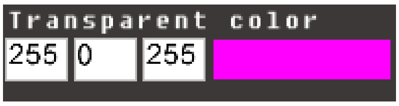
255 0 255

In the Texture selection menu, we can choose a Diffuse texture to load for our 3D mesh:

To load a texture we need to click on the Diffuse Map option using the left mouse button. In the next window, we need to browse the file system on our computer to find the texture we wish to load. 3D2Sprite supports the following formats to be used as textures: .png .jpg .jpeg .bmp
The texture must be a single texture file, multiple textures does not supported at the moment.

5. Transparent color of texture:

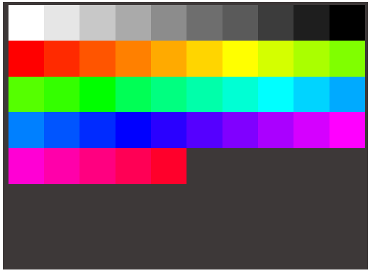
If we look carefully, we can notice the Transparent Color option on the bottom of the Texture selection menu.



Transparent color

255 0 255

Here we can choose a color that is going to be transparent on our texture. By default this color is pink and it RGB code is 255 0 255. We can change to color to anything we'd like by using the left mouse button on the pink box which is representing the actual color we have set.
When we click the box, a simple color picker should appear with some basic colors to choose from



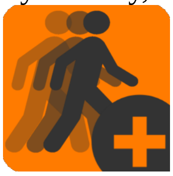
To select a color, just use the left mouse button in the color picker.
If we can't find the color we would like to use in the color picker we can use the RGB code of the color in the boxes. In case we don't know we can use any paint program to find the rgb code of the color.The first box from the Left is the value of the Red component, the second is the value of the Green component and the last box is the value of the Blue component.

The transparent color is important for two reasons. If our texture is using the selected transparent color anywhere on the texture, that area of the texture going to be transparent after exporting the sprite even if we don't want it to be transparent. So it is very important to make sure the transparent color is not something wed like to keep visible on our sprites and also to make sure it is the color we'd like to have it transparent on our sprites. The background color of the 3D window is set to be transparent by default.

The second reason this option is important is that if we would like to render the 3D mesh with transparency to see what is behind the transparent area, it is require a special rendering option that we are going to talk about later, but the success of that option also depends on this transparent color option. I'm going to come back to this later...

6. Load animations:

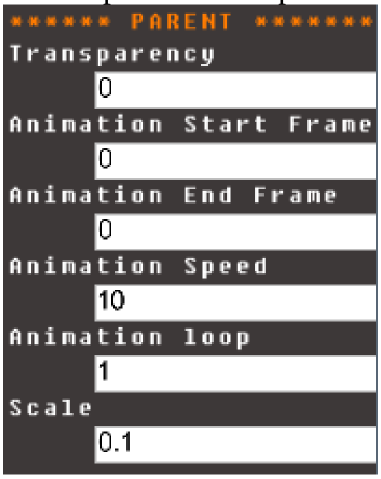
Since, 3D2Sprite does not load the animations along with the mesh, and there is a reason for that, we need to load it manually. To load an animation of a 3D mesh which can be only the Parent mesh by the way, we just simply need to use the left mouse button on the next icon on the top.



Once we have clicked the icon, 3D2Sprite going to load all the animations and display a message if the animations successfully loaded or there is an error. The 3D mesh we are using must have the animations by default, otherwise 3D2Sprite fail to load the animations. Once we have done loading the animation, we need to select the animation frames to play and being captured.

7. Select animation frames to play and to be captured on sprites:

Once we have the animations loaded, we can select the animation frames we would like to play and to be captured on to sprites. On the right side we can see several options for the parent mesh.



****** PARENT *******

Transparency

0

Animation Start Frame

0

Animation End Frame

0

Animation Speed

10

Animation loop

1

Scale

0.1

The ones we are interested in right now;

- Animation Start frame
- Animation End frame

I suppose, this is not the first time you see those options, but in case it is the first time and you don't know what a start frame or end frame is just read on.

Animations basically are stored within frames also called keys and every animation must have a start frame where the animation starts and an end frame where the animation ends. For example walk animation may starts at frame 236 and ends at frame 252. To play the animation, you need to enter the start frame number and the end frame number in to the relevant boxes on the right side. The value of the end frame must be higher than the start frame or equal but can't be less.

As soon as you enter the value of the end frame and it is higher than the start frame or equal, the 3D mesh start playing the animation. Remember, the animation of the 3D mesh must be loaded.

You can also change the animation speed and loop on the right side. The higher the speed value is, the faster the animation going to play. It is 10 by default, but in some cases, it need to be as low as 1 or as high as 20 or even higher, it is depends on how the actual animations designed by the artist and also depends on the mesh format we are using.
In some unfortunate cases, animations may not going to play properly because the animation speed need to be lower than 1, something like 0.2 for example, but at the moment it is not possible, the smallest value we can use for animation speed is 1 but we can use as high as we wish.

The loop value should be also familiar. The value can be 0 or 1. If we set the value to 1 which is the default by the way, the animation going to repeat forever, never stop. If we set the value to 0. the animation going to stop at the last frame.

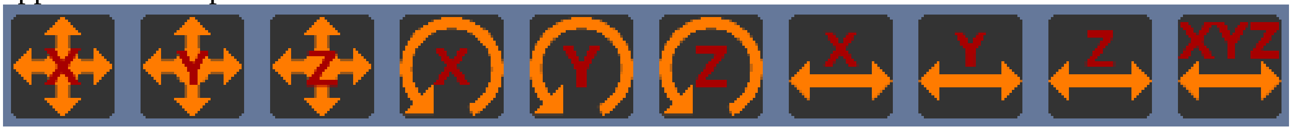
Important to know, the animation speed and loop value does not effecting the capturing process. During capturing, every single frame get captured regardless the speed and loop value we use here

8. Move, Rotate and Scale models

Once we have loaded our child mesh in to the software, the position, rotation and scale of the parent and child mesh may not fit perfectly and we need to change it manually.

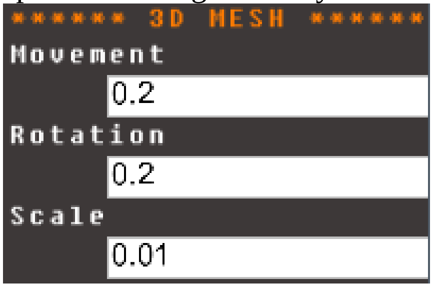
To change the position, scale or rotation of a model in 3D window, first we need to select the model by using the left mouse button on the model in the 3D window. If the selection success, the model should turn red or at least something different, it is effected also by the actual ambient light options.

Right after we have selected a model using the left mouse button, we should see some buttons appear on the top of the screen:



From left to right, the first 3 buttons moves the selected object on X, Y or Z axes. The next 3 buttons, rotates the selected object on X,Y or Z axes and the last 4 buttons, scales the object on X,Y,Z or equally on all XYZ axes. To move, rotate or scale the object on a specified axes, just move the mouse over the relevant icon, hold the left mouse button and drag the mouse left or right, while you keep the left mouse button down.

In case you find the movement, rotation or scale speed too slow or too fast, you can change the speed on the right side by the 3D MESH section:



****** 3D MESH ******

Movement

0.2

Rotation

0.2

Scale

0.01

The movement and rotation value is a float value and must be greater then 0. Can be as low as 0.1 or even less and as high as we wish. Lower value is slower and higher value is faster. I suggest a value somewhere between 0.1 and 1

The rotation value is also a float value but must be a value between 0 and 1. Can be as low as 0.01 and as high as 0.99. Lower value is slower and higher value is faster. I suggest a value somewhere between 0.01 and 0.3

To deselect a model just press the Escape key or click away from the model.

Note: At the moment there is a bug with object selection. After loading animation it is become difficult to select the object because it does something to the collision mesh. In case you experience the problem, try clicking different places on your object to find the right spot to be able to select it. For example one case after loading the animation, I was able to select the model only if I was clicking below the knees. In other case, I was able to select it only if I was clicking on it hands. Weird bug, if you experience it, just try clicking different places on your model.

9. Transparency of parent mesh:

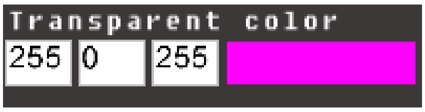
If we take a look at the right side, we can see number of options for the Parent, and there is 2 more options that we haven't talked about it yet, one of them is:

- Transparency

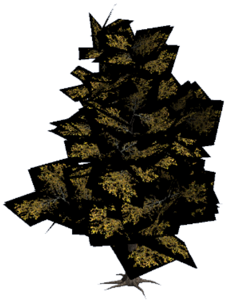
Transparency is something I have already mentioned earlier in connection with the transparent color. In some cases, we need something more then textures and sprites with transparency and transparent colors. When we wish to see and capture what is behind the transparent areas of a 3D mesh we need to apply a special rendering technique that allow it to render transparency in real time so we can see behind the transparent areas.

The default value of Transparency is 0 which means, we do not use this special rendering technique, the reason for that is, it is a very resource hungry process, so better to keep it disabled and enable it only when we need it. To enable it, we simply need to set the value to 1 and to disable it, we need to set the value to 0

When we enable the Transparency, basically what is happening, 3D2Sprite gong to render transparency using the transparent color.
To make it work, it is very important to make sure the Transparent color option in Texture selection menu of the Parent is set to the color that we want to be rendered as transparency.

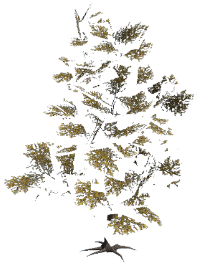

Transparent color
255 0 255

To demonstrate, here is an example:



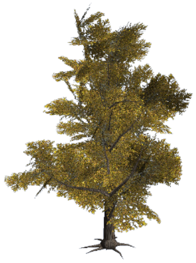
On the above image, what you see is a tree with black color between the leafs. The black area should be transparent but because the default transparent color in 3D2Sprite is pink, the black color on this sprite did not become transparent.

But transparent color itself is not everything:



On this image, what you see is the same tree but this time the transparent color is set to black, and it is sort of worked because the black area on the sprite become transparent but not the way we wanted because in this case we would like to see what is behind the transparent color and not just make the sprite transparent. So what is missing?

The Mesh Transparency:

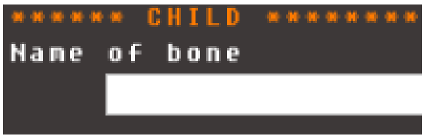


What you see here is the correct transparent color (which is black in this case) used along with the Transparency enabled (set to 1). As you can see the transparency is rendered in real time so we can see what is behind the transparent areas and we can make a nice sprite.

10. Assign child mesh to a bone of the Parent mesh:

So, we can load our child mesh just the way we load a parent mesh. You might noticed that you can not load a child mash before you have loaded the Parent mesh and also loaded the animations of your Parent mesh, the reason is because the real purpose of the child mesh is to assign it to a bone of a Parent mesh, so we can for example add a weapon in to the hands of a character.

So how we are going to assign it to a bone?
If you take a look at the options on the right side, the Child section got only a single option to choose from:



****** CHILD ********

Name of bone

- Name of bone

This is the place where we need to enter the name of the bone that we wish to use to assign the child mesh to it. We can not use the ID of the bone here, the bone must have a name in order to be able to assign a child mesh to it. The name can be anything, and it is depends on the artist who created the model so we need to have a name and need to know what the name is to use this option.

After we have entered a name, just need to press Enter to confirm it, and in case the name is correct, our child mesh going to take it relative position by the selected bone.
From now on the child mesh going to stay by it relative position to the bone and going to follow the movement of the bone during animation and rotation We can change the bone any time by simply changing the name of the bone in the box.

11. Capturing the mesh and export the Sprites:

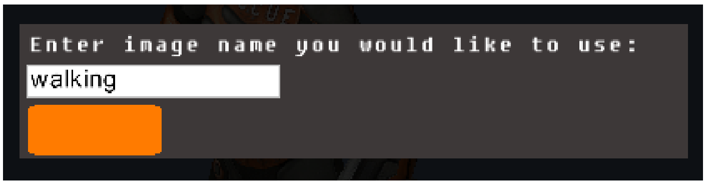
Finally, the reason this whole application was born for. Once you have set and loaded everything you need to position the view in the 3D window to look the way you want. For example, in case you need a sprite for a side scrolling platformer game, position the view to look at the 3D mesh from it side:

In case you need sprite for an isometric game, position the view to look at the 3D mesh from Isometric perspective and so on:

Once you have positioned the view, you can also (and recommended to) select the area you wish to capture. You can select the area by holding the Space key and Left mouse button in the 3D window and drag the mouse. In case of animated mesh make sure the mesh fit in to the selected area with all animations. Once you have selected the area, just press the little camera icon on the top to get the process started.

📷

First you need to enter the name you wish to use for your sprites. For example in case you are capturing a walking animation you may need to consider to add a name like 'walking' so the sprites going to be exported with the file name “walking”.
Once you have entered the name, just press the orange button on the bottom.



Enter image name you would like to use:
walking

Now, 3D2Sprite technically going to capture the selected area of the screen with the mesh and every single animation one by one. Very important, do not minimize the window or open an other window during the process. 3D2Sprite application window need to remain focused and open until the whole process is finished, otherwise you are going to get empty sprites.

In the next step, you can see the actual size of your sprites along with a preview to show you how is your sprite looks like.

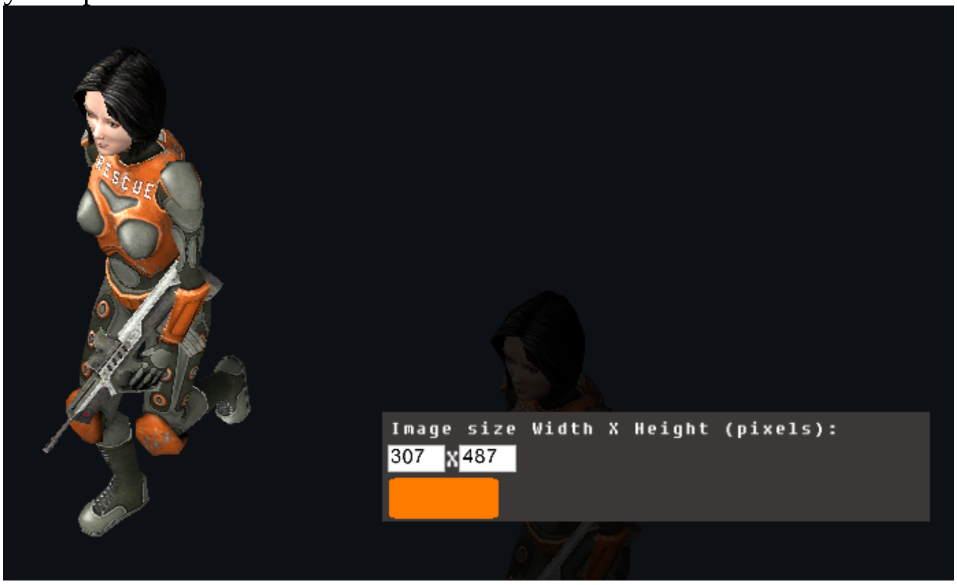


Image size Width X Height (pixels):
307 X 487

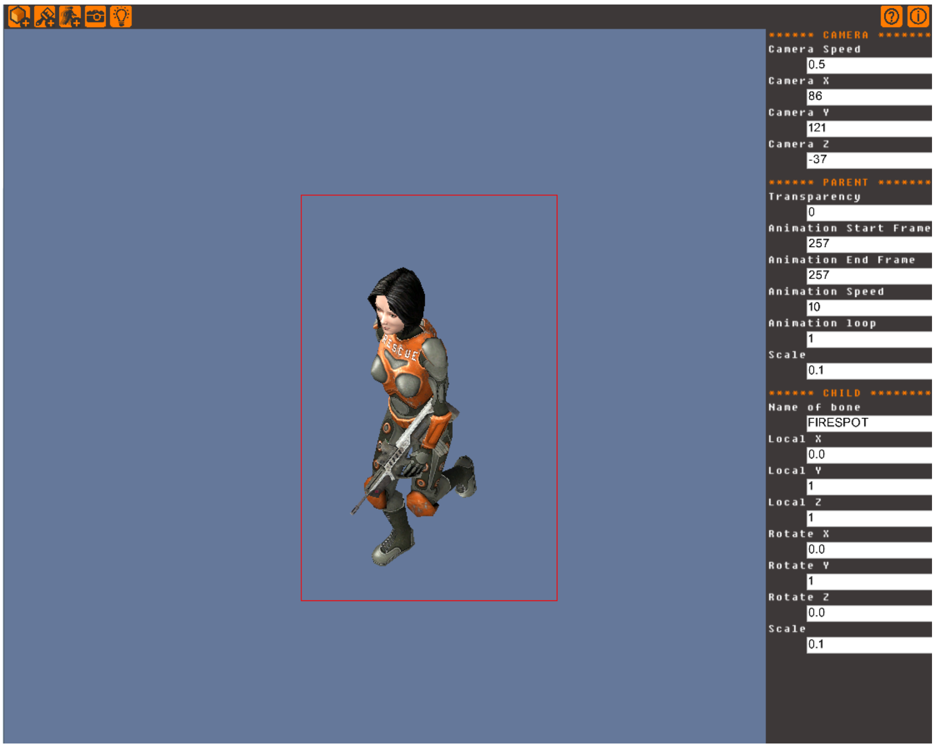
Because, technically 3D2Sprite just making screen shots, depends on your resolution you may get huge sprites and you wish to resize them.

To resize the sprite just select the box displaying the current size of the sprite, change the value and press Enter to confirm. The box displaying pixels so you need to think for a reasonable size in pixels. The size of the preview sprite also going to change to show you how is your sprite looks like at the selected size.

If you have changed the size and you are happy with the look of the preview, just press the orange button on the bottom.

Now, 3D2Sprite going to resize every single sprite exported earlier, you can see them in the process in the top left corner. Just like before, very important to keep the window in focus and open until the process complete.

Once the resizing process is completed, you are going to see to usual screen with the menus, the mesh and selected area.



****** CAMERA *******

Camera Speed
0.5

Camera X
86

Camera Y
121

Camera Z
-37

****** PARENT *******

Transparency
0

Animation Start Frame
257

Animation End Frame
257

Animation Speed
10

Animation loop
1

Scale
0.1

****** CHILD ********

Name of bone
FIRESPOT

Local X
0.0

Local Y
1

Local Z
1

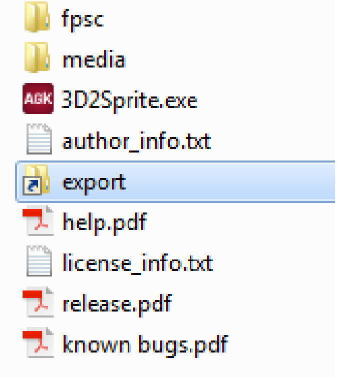
Rotate X
0.0

Rotate Y
1

Rotate Z
0.0

Scale
0.1

Okay, so where are the sprites actually you may ask. If you open up the folder of 3D2Sprite, you are going to find a shortcut in the root folder of 3D2Sprite called “export”.



fpsc
media
3D2Sprite.exe
author_info.txt
export
help.pdf
license_info.txt
release.pdf
known bugs.pdf

By opening the shortcut you are going to find all the sprites exported earlier.
You can copy, and move this shortcut to the desktop, to favourites, anywhere you wish for easy and instant access and you can also rename it to whatever you prefer.

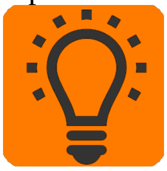
In case you accidentally delete the shortcut, the location of the sprites are the follow:

Windows:
C:\Users\username\AppData\Local\AGKApps\3D2Sprite\export

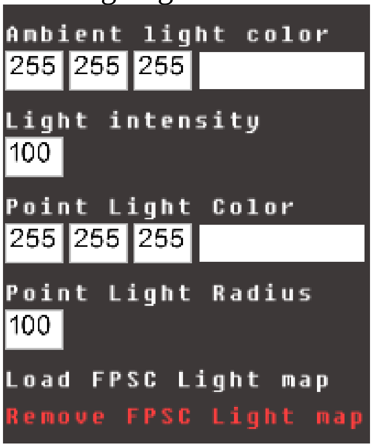
Ubuntu:
/home/.config/AGKApps/3D2Sprite/export

12. Ambient light color and intensity:

In 3D2Sprite we can change the Ambient light color and intensity by clicking on this icon on the top:



We are going to see several options here.



Ambient light color

255 255 255

Light intensity

100

Point Light Color

255 255 255

Point Light Radius

100

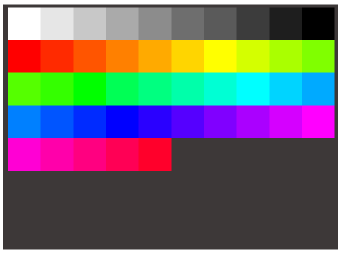
Load FPSC Light map

Remove FPSC Light map

The ones we are interested in:

- Ambient light color
- Light intensity

To change the light color, we can use the left mouse button on the white box on the right side which is representing the currently set ambient color. When we click the box, a color picker should appear with some basic colors to choose from:



To select a color, juts use the left mouse button in the color picker
In case you can't find the color you would like to use, you can use the RGB code of the color and enter the values manually in to the boxes.
In case you don't know, we can use any paint tool to find the RGB code.
R= The red component of the color
G = The green component of the color
B = The blue component of the color

The first box on the left is for the Red component of the color, the next box is the Green component and the last box is the Blue component.
The value can be anything from 0 to 255.

To change the intensity of the light, we just simply need to change the value in the box. The value rang is 0 – 255. The lover the value is, the more intensive the color is, the higher the value is, the less intensive the color is.

In case you wonder why is this useful; to change the color and intensity of the ambient light can be useful to make our sprite look more dark for example when we plan to use them in a dark environment, or for example when we plan to use them inside an environment with red lights all around the place, we can apply some red ambient light on the sprite to make it fit with the environment.

Tip: to make your sprite dark, set Ambient light color to Black, the RGB code is 0,0,0 and make the intensity value something around 120

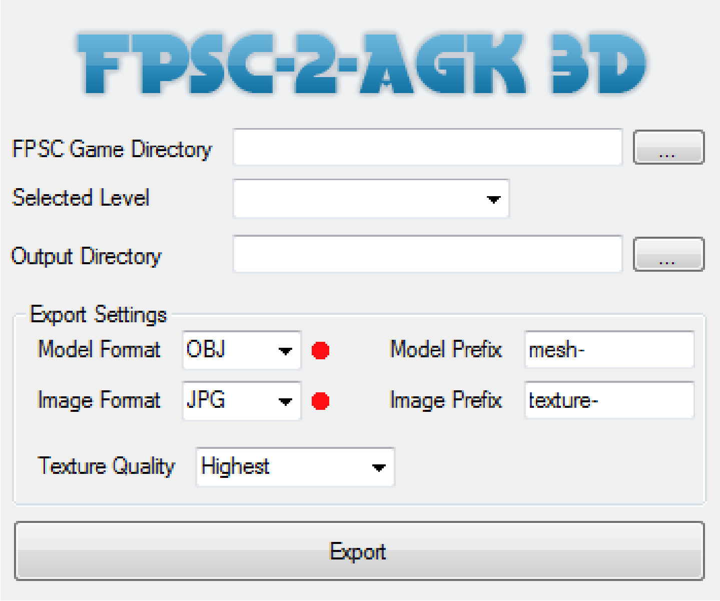
13. Load FPS Creator Level with all models, textures and light map:

In case you don't know, FPS Creator is a fun and easy to use free and open source FPS game maker developed by TheGameCreators. We can download the latest version and source code and many model packs from GitHub:
https://github.com/LeeBamberTGC/FPS-Creator-Classic

By using this feature, you can load a complete level made in FPS Creator in to 3D2Sprite. FPS Creator is not intended to be used as a 3rd party level editor but my vision is that the walls and floors of the rooms we can build in FPSC can be used for example in isometric or a top down view games.

This is an experimental feature, I'm not entirely sure what would be the best practice to make the most out of it, so the way it works and the way 3D2Sprite handles the imported level may change in the future.

In order to be able to load an FPSC level in to 3D2Sprite, first we need to convert the level to .obj and .jpg format. You can easily do this by using this free tool, called FPSC-2-AGK.
You can download it from here:
https://drive.google.com/folderview?id=0By25hYDe9OvVU2tMVXowUExIZGs&usp=sharing



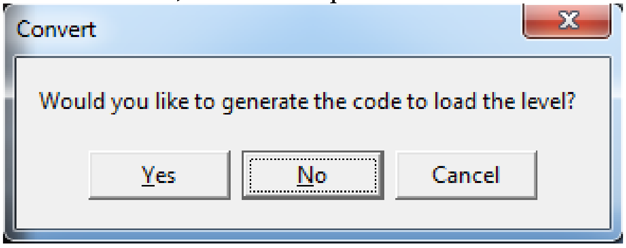
Model Format: OBJ
Image Format: JPG
Model Prefix: mesh-
Image Prefix: texture-
Texture Quality: Highest

This tool is going to convert FPSC own 3d format to a more popular .obj format and the texture format to .jpg format and the light maps to .png format that we can load in to 3D2Sprite, so it is very important, when you run the application, select .OBJ and .JPG to export and keep the Model Prefix and Image Prefix on default otherwise it is not going to work in 3D2Sprite!

For information on how to use the converter, please refer to it help document.

I was implementing the feature to load the walls, floors, doors, windows and different objects in the rooms so I was planning to work with only a single room on a single layer and it works the best that way.
If you try to load a complete level with multiple rooms on multiple layers, it may not going to work as you expect and also it makes no sense really. After all, we need sprites of the walls and floors, individual objects and not the entire level.

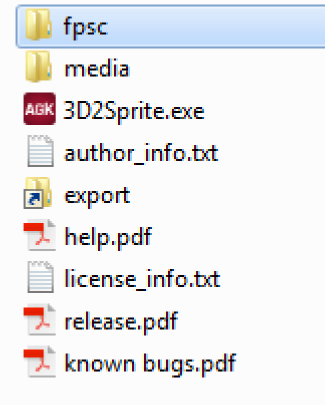
When you convert a level and FPSC-2-AGK ask you if you would like to generate the AGK code to load the level, it is not required to work with 3D2Sprite and you can choose No.



Convert

Would you like to generate the code to load the level?

Once you have everything done and you are ready to load your FPSC room in to 3D2Sprite, first, you need to copy all the files in to the fpsc folder which is located in the 3D2Sprite folder:

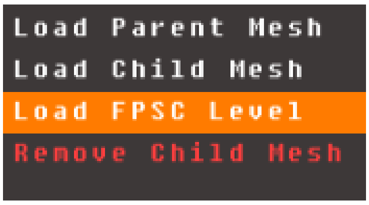


fpsc
media
3D2Sprite.exe
author_info.txt
export
help.pdf
license_info.txt
release.pdf
known bugs.pdf

Make sure you are copy the files in to the root of fpsc folder. Actually, in FPSC-2-AGK you can set the output directory to be this directory to make your life easier.

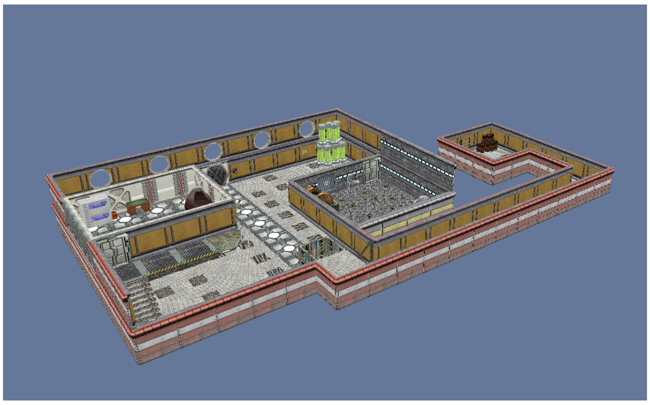
Once you have all the files in place, the only thing you need to do is to click on the load mesh icon in the top.

and select the Load FPSC Level option.



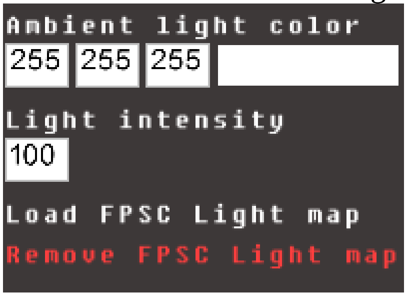
When you load an FPSC Level, you may not going to see it at first, because of the position of the level. To find it, just look around :)
The best practice, when you design your level, always build the level at layer 0 and in the top left corner, this way the level is going to be loaded at position 0,0,0 in 3D space.

The Level should load with all textures and models.



The light map is not loaded at this point. To load the light map, click on the light options icon

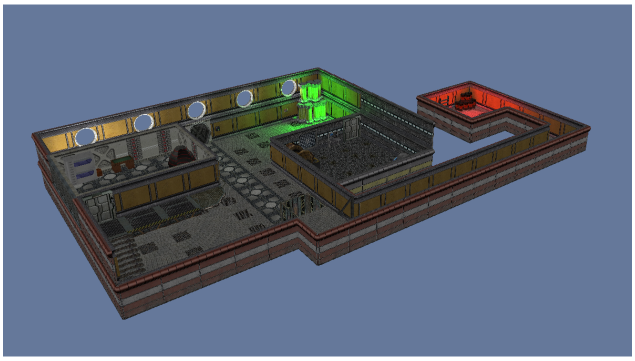
and select the Load FPSC Light map option on the bottom.



Ambient light color
255 255 255

Light intensity
100

The light map should load just fine.



To remove the light map, select the Remove FPSC Light map option.

Keep in mind, when you have a light map loaded, the ambient light options and point light does not effecting the FPSC room any more and the reason for that is because it is using different, custom shaders to render the lightmap and the ambient light and point lights doesn’t work with that shader. In case you wish to use the Ambient light options or point lights you need to remove the light map so the system going to use the default shaders again.

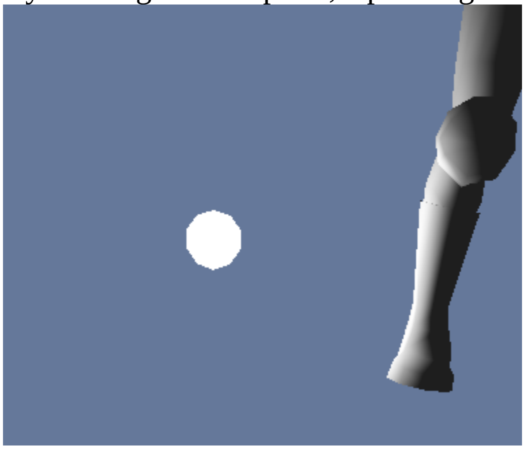
14. Add point light and change it position, color and radius

To add a point light, we need to use the left mouse button on the first icon in the top left corner:

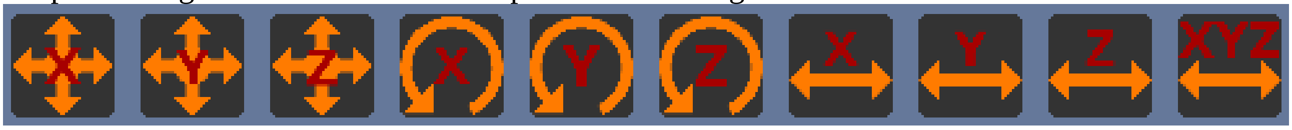
In the drop down menu we need to select the last option:

- Add point light

By clicking on the option, a point light should appear in the 3D window:

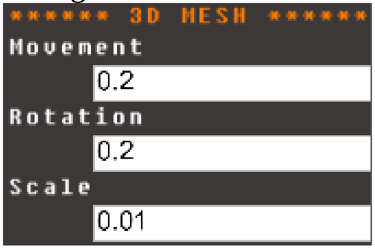


By default it is a white light with radius 100 normally created by 100 pixels in front of the camera. In some unfortunate cases, the light may created inside our 3D mesh, in case you can't see the light mesh in the window after you have created, it most like inside your 3D mesh. Just try to find it and you can select the light using the left mouse button and position it anywhere in the 3D space using the positioning buttons visible on the top when something is selected.



Technically, you can also rotate and scale the light mesh but it does not effecting the actual point light itself. The light mesh is there only to represent the position and color of the point light and to be able to select it and move it around.

When you try to change the position of the point light, remember to change the Movement speed on the right side to be able to move it faster:



****** 3D MESH ******

Movement

0.2

Rotation

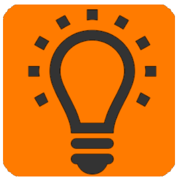
0.2

Scale

0.01

I'm always find my self changing the Movement value to 1 when I'm moving lights.

To change the color and radius of the point light, first, you need to select the point light you want to work with using the left mouse button and after you need to left click on the light options icon:

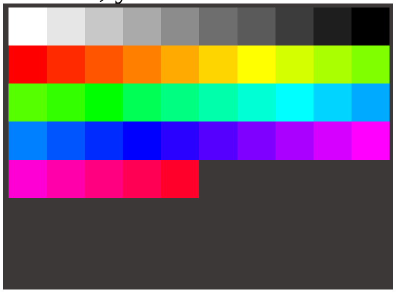


Make sure the light remain selected.

Once you see the drop down menu, the options we are interested in is:

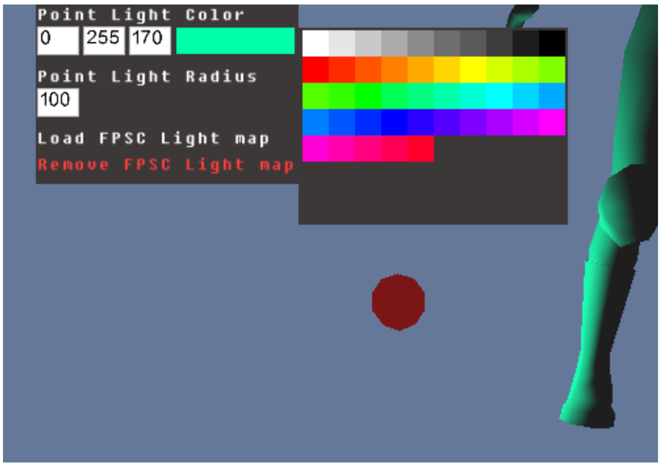
- Point light color
- Point light radius

To change the color of the point light, you can use the left mouse button on the white box on which is representing the currently set color of the point light which is white by default. When you click the box, you should see a color picker with some basic colors to choose from:



To select a color, just use the left mouse button in the color picker.

In case you can't find the color you need , you can use the RGB code of the color you wish to use. In case you don't know, you can use any paint program to find the RGB code of your color. Once you have changed the color, it should immediately change in the light options and also in the 3D window.



Point Light Color

0 255 170

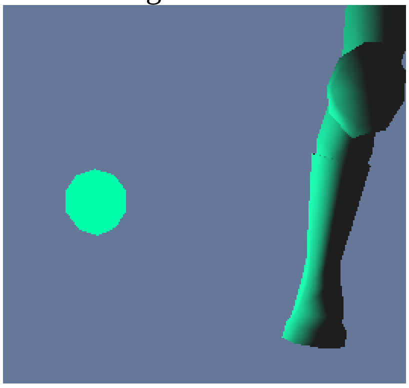
Point Light Radius

100

Load FPSC Light map

Remove FPSC Light map

The color of the light mesh also changes to the selected color but it is going to be noticeable only when the light mesh is not selected:



To change radius of the point light, simply change the value in the box. Lower value is smaller radius, higher value is bigger radius. The value must be greater then 0. The radius of the light is not visually represented in the 3D window but you should notice the change of radius anyway.

To delete a point light, select the point light using the left mouse button and press Delete or Del on your keyboard.

Note: When you add a point light, the ambient light is automatically set to dark to increase the effect of the point light and when you delete the point light the ambient light is automatically set to bright to avoid darkness in the 3D window.