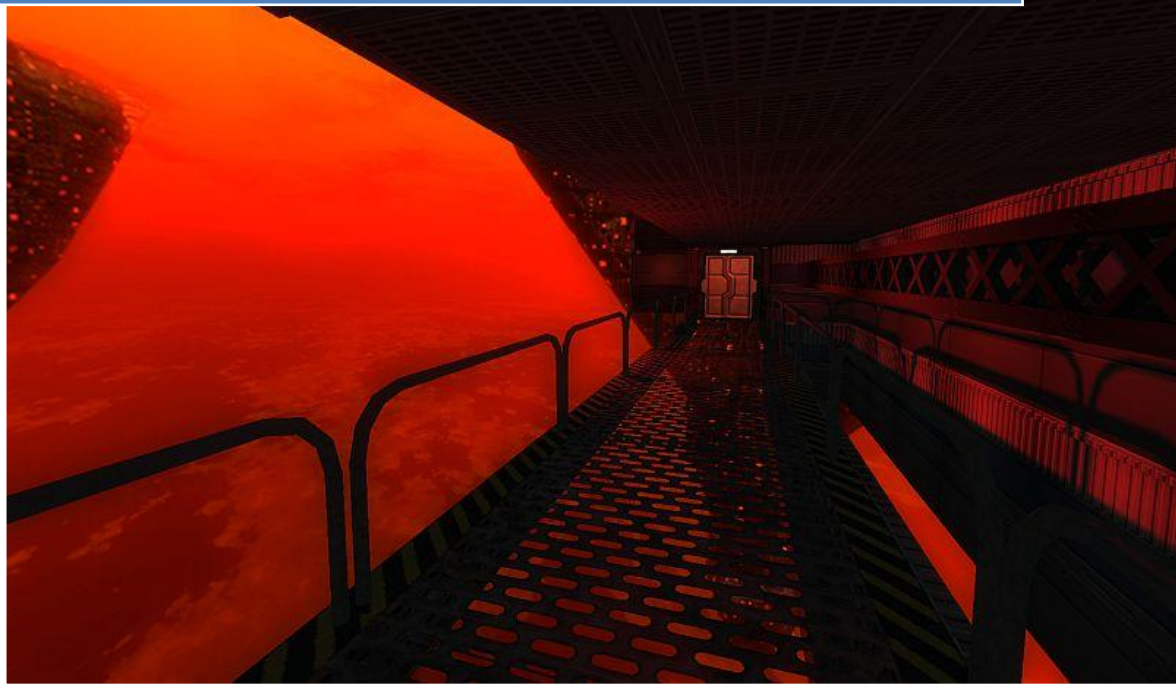




Models created by Mark Blosser (Bond1)

21 February 2011

The Official FPS Creator Community Guide – Part II



Nickydude

MadLad Designs

For FPSC V1.18 and up

Contents

Introduction.....	4
Scripting.....	5
ACTIONS.....	5
CONDITIONS	6
Internal variable recognition.....	7
GUI -X9 Commands.....	8
Tutorials.....	19
So you want to make a game?	19
FPSC: Enemies and You	28
LIST 1: The Basics - The Core Placement Methods Often Used In Games.....	29
Numerical hud using gimp	35
Setting Up Polygon Hit Detection in FPSC.....	41
LEVEL DESIGN TUT #1	41
"0 AMBIENCE" LIGHTING TUTORIAL	45
Particle Decal System.....	47
Getting Characters To Climb Stairs	47
Getting Characters To Go Through Certain Doors.....	48
Getting Characters To Talk.....	49
Getting Characters To Think.....	49
DARK AI.....	51
Dark AI FPI Reference.....	51
Model Animation list – Normal and TF341 (model pack 53).....	53
Hints & Tips.....	60
FPSC Game Engine Optimization.....	60
General Optimisation Tips.....	60
Level Design Tips.....	60
Save Polies	61
Screen Resolution	61
Textures/ Pride	61
Lighting Techniques	62
Use Few Characters.....	62
Enemies	62
Save Framerate.....	62
Delete Dynamic Objects When Player Leaves Vicinity	63

Entity Collision	63
Static Objects	63
Invisible Walls	63
Use of Skyboxes	63
Indoor Level Skyboxes	64
Physics	64
Model Pack 3 (Boxes) Tip	64
Lightintensity/Strength Command	64
GamePlay Tips/Hints	65
Sun in FPSC	65
There aren't Pure Colors in the World	66
The Use of Modifications	66
Texture Quality-Add-on to Prides	66
Keep player moving forward-add-on to Hockeykids	67
An interview with John Zuur Platten, a top-notch game writer.	67
Q: How do you beat writer's block?.....	67
Q: What are some good, FPSC worthy ways, to create a climax to your story?	68
Q: How should writers warm up before they sit down to start typing or writing away at a story?	69
Q: Help! I'm a game designer who just found out a feature isn't capable in the engine's limitations! What do I do?!.....	69
Q: Are there any screenwriting programs you recommend for a small budget? We're talking about, say, \$50 of a budget.....	69
Q: If you could give one message to every aspiring game designer out there, what would you say?	70
Q: Do you need to come out with the next Call of Duty or Half-Life to be a successful game developer? .	70
Q: What is the best way to get noticed as a game developer?.....	70
Q: If all someone can come up with is a story made clichés, what is, in your opinion, the best way to get out of the cliché state of mind?.....	71
Q: What's an effective way to convey story through level design without overpowering FPSC?	71
Side Question: So having the player actually experience the event is more compelling than a cinematic cutscene?.....	72
Questions & Answers	73
Q: How do I change the death animations of a character?	73
Outro	74

Introduction

Well it's been a long time coming, and FPSC has moved on in leaps and bounds since the last Community Guide was wrote so I thought it was about time I updated the guide with all the new stuff that's been added to FPSC. This isn't a rewrite of the first guide but more of a follow on so you won't find stuff repeated in here, eventually I might combine the two to make one mighty guide!

I'd also like to take this opportunity to thank the FPSC Community itself for driving forward the advancements of FPSC and taking it from a virtually unknown small hobbyists' tool to something that can now make commercial games... and things can only get better!

Scripting

Below is a full list of the additional actions and conditions that extend the original version of the FPI script system. Many thanks go to the endeavours of the FPSC community who took the original FPSC source code and added their own cool features over the years:

ACTIONS

FPI COMMAND	DESCRIPTION
<code>dimvar = X</code>	Declares a Global variable with the name of X
<code>setvar = VarName Y</code>	Sets VarName equal to Y
<code>addvar= VarName Y</code>	adds Y to VarName
<code>subvar=VarName Y</code>	Subtracts Y from VarName
<code>mulvar=VarName Y</code>	Multiplies VarName by Y
<code>divvar=VarName Y</code>	Divides VarName by Y
<code>modvar=VarName Y</code>	Stores the Modulus of VarName and Y into VarName
<code>wrapvar=VarName</code>	Wraps the variable's value around from 0 to 360 and from 360 to 0, Example: Variable "AngX" is 365, using <code>wrapvar=AngX</code> , it's value will become 5
<code>selectshadervariable =X</code>	X is the entity's shader variable (0 to 4)
<code>setshadervariable=X</code>	Sets the entity's current shader variable value to X
<code>incshadervariable=X</code>	Increases the entity's current shader variable value by X
<code>decshadervariable=X</code>	Decreases the entity's current shader variable value by X
<code>freeze</code>	Stops the entity from moving
<code>animationnormal</code>	This will set the entity's animation back to normal so animations will no longer be reversed
<code>animationreverse</code>	This will play any animations called after this in reverse
<code>plrfreeze=X</code>	Freezes the player's movement for X amount of seconds
<code>plrdisable=X</code>	Stops the player from doing anything for X amount of seconds
<code>talk=X</code>	Triggers the Dark Voice sound file specified by X and triggers Dark Voice's facial movement if the entity supports it (X can also be \$0 or \$1 to specify the entity's sound field)
<code>shapedecal=X</code>	Changes the decal mode to X for the entity
<code>newjumpheight=X</code>	Sets the player's jump height to X
<code>musicoverride=X</code>	This will stop any current music and play the music file specified by X
<code>weblink=X</code>	This will launch the webpage (Using Internet Explorer) specified by X
<code>setifused=X</code>	Sets the entity's ifused field to X
<code>setusekey=X</code>	Sets the entity's usekey to X

resetplrweapons	This will drop all of the player's weapons
backdropvideo=X	Sets the backdrop to video file X
bloodspurt=X:	Spurts blood using one of the following conditions defined by X <ol style="list-style-type: none"> 1. Last Damaged Position 2. Head Position 3. Center of Model, Falls Straight Down
nobulletcol=X	Set to 1 to remove from bullet collision, 0 to restore
isaltammo	Sets ammo entity to provide alternate fire ammo
bloodsplash=X	Splashes a blood scorch in 6 directions within distance X
camshake=X	Shakes camera by amount X (max: 20)
camfov=X	Moves camera FOV to default FOV plus X
camfovinc=X	Increases the camera FOV by X
hideplrweapon	Puts the current weapon away
showplrweapon	Takes out the weapon the player was holding when "hideplrweapon" was called
plrpointatobject=X	Points the player at the entity, X defines a Y offset
fpgcrawtexttr=X	Sets the raw text's red color value to X
fpgcrawtextg=X	Sets the raw text's green color value to X
fpgcrawtextb=X	Sets the raw text's blue color value to X
fpgcrawtextx=X	Sets the raw text's X position to X
fpgcrawtexty=X	Sets the raw text's Y position to X
fpgcrawtextsize=X	Sets the raw text's font size to X
fpgcrawtextfont=X	Sets the raw text's font type to X
fpgcrawtextoff	Turns off/resets the raw text
fpgcrawtext=X	X specifies what you want the raw text to say
etimerstart	Reset the entity's personal FPI script timer to zero, allowing timing to take place

---oOo---

CONDITIONS

FPI COMMAND	DESCRIPTION
plrblocking	True if player is performing a block action
inview	True if entity is within the current view of the camera
pickobject	True if entity is immediately in front of camera (centre screen)
velocitygreater	True if physics based entity velocity greater than value
etimergreater	True if entity timer greater than value
keypressed	True if value representing key is pressed
hasweapon	True if entity has a weapon

<code>varequal=VarName Y</code>	True if variable defined as "VarName" is equal to X- You can now use real values (Given by \$ or % recognition or by numbers) in place of VarName. Ex. varequal=\$PL 1, varequal=10 %VarName
<code>vargreater=VarName Y</code>	True if VarName is greater than or equal to X- You can now use real values (Given by \$ or % recognition or by numbers) in place of VarName. Ex. vargreater=\$PL 1, vargreater=10 %VarName
<code>varless=VarName Y</code>	True if VarName is less than or equal to X- You can now use real values (Given by \$ or % recognition or by numbers) in place of VarName. Ex. varless=\$PL 1, varless=10 %VarName
<code>varnotequal=X Y</code>	True if user variable named X is not equal to Y

---oOo---

INTERNAL VARIABLE RECOGNITION

Just as applies to the Named Variable Recognition, this will allow you to use values that are stored in internal variables.

- Player Health [**\$PH**]
- Player Lives [**\$PL**]
- Player Right-Hand Ammo (Total) [**\$RA**]
- Player Ammo (total)[**\$A**]
- Player Right-Hand Ammo (In Gun) [**\$RCA**]
- Player Ammo (In Gun) [**\$CA**]
- Entity Health [**\$EH**]
- Entity Ammo (In Gun) [**\$EA**] (There is no "Total" ammo for entities since their ammo stock is unlimited.)
- Entity Position X [**\$EPX**]
- Entity Position Y [**\$EPY**]
- Entity Position Z [**\$EPZ**]
- Entity Angle X [**\$EAX**]
- Entity Angle Y [**\$EAY**]

- Entity Angle Z [**\$EAZ**]
- Camera Position X [**\$CPX**]
- Camera Position Y [**\$CPY**]
- Camera Position Z [**\$CPZ**]
- Camera Angle X [**\$CAX**]
- Camera Angle Y [**\$CAY**]
- Camera Angle Z [**\$CAZ**]
- Mouse Movement X [**\$MMX**]
- Mouse Movement Y [**\$MMY**] - Mouse movement variables are used to determine how far the mouse has moved since last loop.

---oOo---

GUI -X9 COMMANDS

MAIN
Use GUI (Action)
<pre>usegui=(1 or 0)</pre> <p>This command lets the FPSC exe know you want to use GUI-X9. It must be turned on before calling any GUI-X9 commands.</p>
Force GUI Update (Action)
<pre>updategui</pre> <p>This command forces the GUI system to update, useful for when the game is paused and you want to use the GUI.</p>
IMAGE
Load Image (Action)
<pre>loadimage=(image name) (image path)</pre> <p>This command loads an image into the GUI system, tagging it with the name given by the command. This image can then be used later in another script to make a hud or other GUI component.</p>

Replace Image (Action)
<pre>replaceimage=(image name) (image path)</pre> <p>This command replaces a loaded image with a new image, any GUI components and huds using this image will automatically update to the new image.</p>
Heads-Up-Display (HUD)
Make Hud (Action)
<pre>makehud=(Hud Name) (Normal Image Name) (Mouseover Image Name) (X Position) (Y Position) {Width} {Height}</pre> <p>This command makes a new hud to use in the GUI system. The mouseover image name can be set as 'None' if not required.</p>
Set Hud X (Action)
<pre>sethudx=(Hud Name) (X Position)</pre> <p>This command sets the X position of a hud, must be used after creating a hud.</p>
Set Hud Y (Action)
<pre>sethudy=(Hud Name) (Y Position)</pre> <p>This command sets the Y position of the hud, must be used after creating a hud.</p>
Set Hud H (Action)
<pre>sethudh=(Hud Name) (Height)</pre> <p>This command sets the height size of the hud (value in pixels, not percent).</p>
Set Hud W (Action)
<pre>sethudw=(Hud Name) (Width)</pre> <p>This command sets the width size of the hud (value in pixels, not percent).</p>
Set Hud Colour (Action)
<pre>sethudcolour=(Hud Name) (R) (G) (B) OR sethudcolor=(Hud Name) (R) (G) (B)</pre> <p>This command sets the colour overlay for the hud, it will change the overall colour to whatever colour is specified.</p>

Set Hud Alpha (Action)
<pre>sethudalpha=(Hud Name) (Alpha)</pre> <p>This command sets the transparency/alpha of the hud.</p>
Set Hud Over Image (Action)
<pre>sethudoimage=(Hud Name) (Image Name)</pre> <p>This command sets the image shown when the mouse <u>is</u> over the hud.</p>
Set Hud Normal Image (Action)
<pre>sethudnimage=(Hud Name) (Image Name)</pre> <p>This command sets the normal image for the hud when the mouse <u>is not</u> over the hud.</p>
Set Hud Numeric (Action)
<pre>sethudnumeric=(Hud Name) (1 or 0)</pre> <p>This command will turn the hud into a numeric value hud. The image used for the numeric value display is the normal hud image specified when making the hud.</p>
Set Hud Value (Action)
<pre>sethudvalue=(Hud Name) (lives/health/ammo/frags/time/your own value)</pre> <p>This command sets the value of a numeric hud. You have a choice of 5 pre-made values or you can enter your own value.</p>
Set Hud Clickable (Action)
<pre>hudclickable=(Hud Name) (1 or 0)</pre> <p>This command turns a non-clickable hud into a clickable hud and vice versa. Can be used to enable and disable clickable huds.</p>
Show Hud (Action)
<pre>showhud=(Hud Name)</pre> <p>This command shows the hud.</p>
Hide Hud (Action)
<pre>hidehud=(Hud Name)</pre> <p>This command hides the hud.</p>

Hud Mousedown (Condition)
<pre>hudmousedown=(Hud Name) 1</pre> <p>This command checks whether the left mouse button is being held down over a hud.</p>
Hud Mouseup (Condition)
<pre>hudmouseup=(Hud Name) 1</pre> <p>This command checks whether the user has clicked a hud, releasing the left mouse button.</p>
Hud Mouseover (Condition)
<pre>hudmouseover=(Hud Name) 1</pre> <p>This command checks whether the mouse is over the hud.</p>
CHECKBOX
Make Checkbox (Action)
<pre>makecheckbox=(checkbox name) (unchecked image name) (checked image name) (X position) (Y position)</pre> <p>This command makes a new checkbox GUI component.</p>
Set Checkbox X (Action)
<pre>setcheckboxxx=(Checkbox Name) (X Position)</pre> <p>This command sets the X position of the checkbox.</p>
Set Checkbox Y (Action)
<pre>setcheckboxxy=(Checkbox Name) (Y position)</pre> <p>This command sets the Y position of the checkbox.</p>
Set Checkbox W (Action)
<pre>setcheckboxwx=(Checkbox Name) (Width value)</pre> <p>This command sets the checkbox width (Value is in pixels, not percent).</p>
Set Checkbox H (Action)
<pre>setcheckboxhx=(Checkbox Name) (Height value)</pre> <p>This command sets the checkbox height (Value in pixels, not percent).</p>

Set Checkbox Alpha (Action)
<pre>setcheckboxalpha=(Checkbox Name) (Alpha Value)</pre> <p>This command sets the transparency/alpha of the checkbox.</p>
Set Checkbox Checked Image (Action)
<pre>setcheckboxcimage=(Checkbox Name) (Image Name)</pre> <p>This command will change the checked image of a checkbox to a new image.</p>
Set Checkbox Unchecked (Action)
<pre>setcheckboxchecked=(Checkbox Name) 0</pre> <p>This command will set the value of a checkbox to 'unchecked'.</p>
Set Checkbox Colour (Action)
<pre>setcheckboxcolour=(Checkbox Name) (R) (G) (B)</pre> <p>OR</p> <pre>setcheckboxcolor=(Checkbox Name) (R) (G) (B)</pre> <p>This command sets the colour overlay for the checkbox, it will change the overall colour to whatever colour is specified.</p>
Set Checkbox Unchecked Image (Action)
<pre>setcheckboxnimage=(Checkbox Name) (Image Name)</pre> <p>This command will change the unchecked image of a checkbox to a new image.</p>
Set Checkbox Unchecked Image (Action)
<pre>setcheckboxnimage=(Checkbox Name) (Image Name)</pre> <p>This command will change the unchecked image of a checkbox to a new image.</p>
Set Checkbox Checked (Action)
<pre>setcheckboxchecked=(Checkbox Name) 1</pre> <p>This command will set the value of a checkbox to 'checked'.</p>
Checkbox Checked (Condition)
<pre>checkboxchecked=(Checkbox Name) 1</pre> <p>This command will check if a checkbox is checked.</p>

Checkbox Unchecked (Condition)
<pre>checkboxchecked=(Checkbox Name) 0</pre> <p>This command will check if a checkbox is unchecked.</p>
Checkbox Mouseover (Condition)
<pre>checkboxmouseover=(Checkbox Name) 1</pre> <p>This command will check if the mouse is over the checkbox.</p>
SLIDER
Make Slider (Action)
<pre>makeslider=(Slider Name) (Background Image) {Fill Image} (Slider Image) (X Position) (Y Position)</pre> <p>This command makes a slider component with a background bar image, optional fill bar image and slider image.</p>
Set Slider X (Action)
<pre>setsliderx=(Slider Name) (X Position)</pre> <p>This command will set the X position of the slider.</p>
Set Slider Y (Action)
<pre>setslidery=(Slider Name) (Y Position)</pre> <p>This command sets the Y position of the slider.</p>
Set Slider W (Action)
<pre>setsliderw=(Slider Name) (Width Value)</pre> <p>This command sets the width of the slider.</p>
Set Slider H (Action)
<pre>setsliderh=(Slider Name) (Height Value)</pre> <p>This command sets the slider height.</p>
Set Slider Colour (Action)
<pre>setslidercolour=(Slider Name) (R) (G) (B) OR setslidercolor=(Slider Name) (R) (G) (B)</pre>

This command sets the colour overlay for the slider, it will change the overall colour to whatever colour is specified.

Set Slider Alpha (Action)

```
setslideralpha=(Slider Name) (Alpha Value)
```

This command sets the transparency/alpha value of the slider.

Show Slider (Action)

```
showslider=(Slider Name)
```

This command shows the specified slider.

Hide Slider (Action)

```
hideslider=(Slider Name)
```

This command hides the specified slider.

Slider Value (Condition)

```
slidervalue=(Slider Name) (Value To Check)
```

This command checks the user selected value of the slider.

CHOICE SLIDER

Make Choice Slider (Action)

```
makechoice=(Choice Slider Name) (Background Image) {Fill Image} (Slider Image) (Choice 1) (Choice 2) (X Position) (Y Position)
```

This command makes a choice slider component with a background bar image, optional fill bar image and slider image. You must provide two choices when making a choice slider.

Set Choice Slider X (Action)

```
setchoicex=(Choice Slider Name) (X Position)
```

This command sets the X position of a choice slider.

Set Choice Slider Y (Action)

```
setchoicex=(Choice Slider Name) (Y Position)
```

This command sets the Y position of a choice slider.

Set Choice Slider W (Action)
<pre>setchoicew=(Choice Slider Name) (Width Value)</pre> <p>This command sets the width of a choice slider.</p>
Set Choice Slider H (Action)
<pre>setchoicelh=(Choice Slider Name) (Height Value)</pre> <p>This command sets the height of a choice slider.</p>
Set Choice Alpha (Action)
<pre>setchoicealpha=(Choice Slider Name) (Alpha Value)</pre> <p>This command sets the transparency/alpha value of the choice slider.</p>
Set Choice Slider Value (Action)
<pre>setchoicevalue=(Choice Slider Name) (Choice Value)</pre> <p>This command sets the selected choice of a choice slider to the specified choice.</p>
Set Choice Slider Colour (Action)
<pre>setchoicecolour=(Choice Slider Name) (R) (G) (B)</pre> <p>OR</p> <pre>setchoicecolor=(Choice Slider Name) (R) (G) (B)</pre> <p>This command sets the colour overlay for the choice slider, it will change the overall colour to whatever colour is specified.</p>
Add Choice Slider Option (Action)
<pre>addchoicevalue=(Choice Slider Name) (Choice Value)</pre> <p>This command adds a new choice option to a choice slider. The option will be added to the right side of the choice slider, pushing all existing choices left slightly.</p>
Show Choice Slider (Action)
<pre>showchoice=(Choice Slider Name)</pre> <p>This command shows the specified choice slider.</p>
Hide Choice Slider (Action)
<pre>hidechoice=(Choice Slider Name)</pre> <p>This command hides the specified choice slider.</p>

Choice Slider Value Equals (Condition)
<pre>choicevalue=(Choice Slider Name) (Value)</pre> <p>This command checks the value of the specified choice slider.</p>
STOPWATCH
Make Stopwatch (Action)
<pre>makesw=(Stopwatch Name) (1/0)</pre> <p>This command makes a new stopwatch. The 1/0 at the end sets whether the stopwatch automatically starts timing when the stopwatch is made.</p>
Start Stopwatch (Action)
<pre>startsw=(Stopwatch Name) 0</pre> <p>This command starts a stopwatch.</p>
Stop Stopwatch (Action)
<pre>stopsw=(Stopwatch Name)</pre> <p>This command stops a stopwatch.</p>
Restart Stopwatch (Action)
<pre>startsw=(Stopwatch Name) 1</pre> <p>This command restarts a stopwatch.</p>
Stopwatch Time Greater (Condition)
<pre>swgreater=(Stopwatch Name) (Value)</pre> <p>This command checks if a stopwatch time is greater than the specified value. (1 second = 1000)</p>
Stopwatch Time Less (Condition)
<pre>swless=(Stopwatch Name) (Value)</pre> <p>This command checks if the stopwatch time is less than the specified value. (1 second = 1000)</p>
Stopwatch Running (Condition)
<pre>swrunning=(Stopwatch Name) 1</pre> <p>This command checks if a stopwatch is running.</p>

Stopwatch Not Running (Condition)
<pre>swrunning=(Stopwatch Name) 0</pre> <p>This command checks if a stopwatch is not running.</p>
SET UP VARIABLE
Make Setup Variable (Action)
<pre>makesvar=(Variable Name) ('setup.ini' Line) (Default Value)</pre> <p>This command makes a new setup variable with a specified line in 'setup.ini' to write to and a default value. The 'setup.ini' line can be an existing line or a new line.</p>
Set Setup Variable Value (Action)
<pre>setsvarvalue=(Variable Name) (Value)</pre> <p>This command will set/change the value of the specified setup variable to whatever value is given.</p>
Set Setup Variable Line (Action)
<pre>setsvarline=(Variable Name) ('setup.ini' Line)</pre> <p>This command sets/changes the 'setup.ini' line to write to, to the specified line. The 'setup.ini' line can be an existing line or a new line.</p>
Read Line To Value (Action)
<pre>readsetupline=(Variable Name) ('setup.ini' Line)</pre> <p>This command will read the value of a line in 'setup.ini' and store the value in the specified setup variable.</p>
Set Value To Component (Action)
<pre>setsvartogui=(Variable Name) (Component Name)</pre> <p>This command reads the value of a GUI component and puts the value into the specified setup variable. Perfect for allowing options to be changed, saved and remembered. The order it searches for the named component is: Choice, Slider, Checkbox.</p>
Save All Setup Variables (Action)
<pre>Savesvars</pre> <p>This command saves all setup variables to the 'setup.ini' file. It will write in the following format: <i>variable line=variable value</i> For example: <i>TestLine=TestValue</i></p>

Setup Variable Value Equals (Condition)

`svarequals=(Variable Name) (Value To Check)`

This command checks the value of a setup variable.

Tutorials

SO YOU WANT TO MAKE A GAME?

By raymondlee306

Who is this for? This document is made for beginners wanting to get their feet wet in the world of game creation and design.

Well, what is a game? A game considered to be an application which initiates an engine, waits for and interprets inputs and displays a result based on the inputs. The inputs can be either from a user through an input device such as a game controller or a keyboard and mouse, or the inputs can be the end result of a script or small program running alongside the game. Please note this is a very vague and made up description of a game. Do not answer a test question in school with this answer. It is merely my layman's definition.

So what does that mean? If you have just gotten First Person Shooter Creator (FPSC), then at this point it does not mean a whole lot. But stay with me and we will continue and create our very first game.

For starters we will be using only stock media. We will get into using and creating custom content soon enough, but let's work on using the software first. While clicking around and playing with the commands is the best way to learn the software, I think we need to have an idea of where to start. Let's begin with picking a story line for our game. Who is the character? Why is he here? Why do we care about this character? What is the "win condition"? When and where does it take place? Let's answer each of those one by one.

Who is the character (player)? – This defines the role that the player is going to see your game world through. For our example I have named him Jack (Totally Random).

Why is he here/Why do we care about the character? – This is the plot. Are we taking part in an invading army? Are we defending from the invading army? Are we searching for lost artefacts? The possibilities are endless, use your imagination? For our game I have chosen that Jack must save his girlfriend from Bad guys with guns. This would be the same as finding an artefact.

What is the win condition? - The win condition is the list of events that need to be completed in order to win the game, complete the level, or unlock the next area. In our game we will have 1 level. The win condition is when the bad guys are killed and jack's girlfriend is saved.

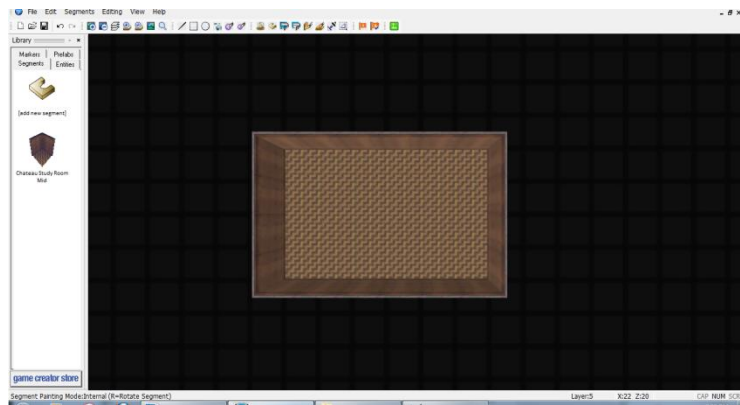
When and where does the game take place? - This is the year, time of day, location, country, spacecraft, etc that our game will take place. For our example I have chosen a 2 story house in Berlin, Germany 1943 9:00am. No real reason other than there is plenty of WWII stock media for us to choose from.

So we have Jack trying to save his kidnapped girlfriend from Badguys in Germany at 9am. So this also tells me that the level will take place during the day. That tells me any lighting that will be needed inside or out doors. Okay we have our story so let's start the software.

Look at your window. In the middle is your working area. This is where you will place all of your objects in the game. On the top are the tool bars which we will discuss as we need them. On the left is the Library (browser panel). This is where you will select the data you wish to place in the game. First and foremost – save, save, save, save, save. Save often, save always and learn to love your save button. I hope I made my point about saving. If the software crashes, and it will because all software crashes, you can control how much data you lose based when the last time you saved was. Okay so let's start by saving. Let's go to file then save and create a new folder called "Jack" in the default file location the FPSC brings us to. Open the new "Jack" folder and let's call this level 1.

Let's make our first room. If you look at the bottom of your screen you can see what mode you are currently in, Layer, X, Z locations. The mode tells you if you are drawing segments, placing entities, editing way points, etc. Layer tells you which Vertical layer (floor) you are on and the X, Z location tells you where you mouse is on the grid. So you can draw a 40 x 40 map that can be up to 20 layers tall. That would be a huge game level. Plus by default you can have 9 game levels, giving you almost infinite possibilities of level and game creation. But we're new to this so let's keep it small right now. The game Takes place in a 3 story building, so the first level will be the first 2 floors and the second level, which we will make later, will be the second and final level. A great idea would be to use graph paper where 1 square on the paper will equal 1 square on the screen. But we will just wing it right now. I picture the 1st floor will be 4 main rooms, 1 staircase, and a hallway. So how do we make the first floor?

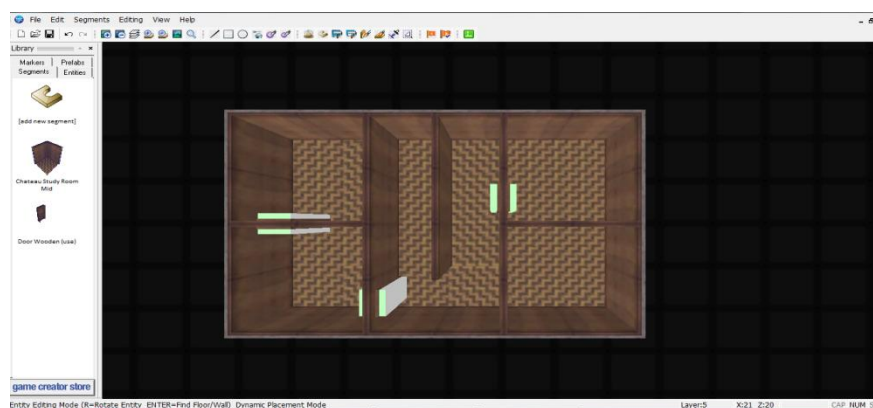
On the left side of the screen in the Library let's select the segments tab. A segment is considered what will make up the terrain walls, floors, and ceilings of your game. A staircase is a terrain and therefore a segment, but a table in a kitchen or bed in a child's room is not. We will tell you what a table is later. In FPSC the default size is 100 inches (or units) long x 100 inches wide x by 100 inches tall. You see there is only "add new segment" So let's hit it. Now the right side of our screen has changed and we can select our media from the expanding folders on the left. Let's pick the WWII folder then open the rooms folder and select the "Chateau Study Room Mid" Segment and press OK. You can also double click the icon. Let's stay on level 5 (you can change levels by pressing your – and + keys) and move to the middle by using your Arrow keys. Now hold down your left mouse button and let's draw a rectangle 4 segments by 6. If you mess up let go of the left mouse button, move your cursor to the bad part and press the right mouse button and you will delete it. You can play around a little here to get used to drawing segments.



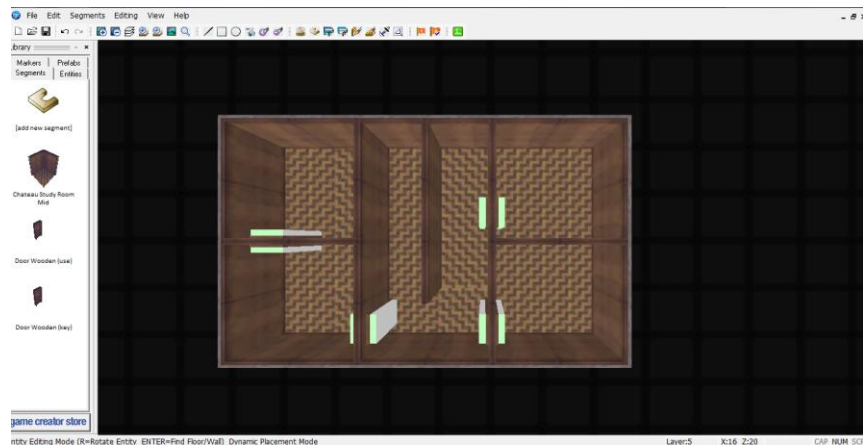
Now we have the main first floor of our house. Let's add some interior walls. If you no longer have the Chateau segment selected, you can just click on it in the browser to re-enter segment draw mode. Press letter "M" or select "Paint only Segment walls" from the tools at the top and draw your interior walls. You can see that the arrow points to the wall that appears and once again you can right click to delete any mistakes. Press letter "R" to rotate the arrow to the place you would like to press it. Go ahead and make it look like this:



Okay so we now see the room layout taking shape but we need to create a way to move between the rooms...Like doors. Doors are also considered segments so let's add a new segment from the WWII Scenery file called "Door Wooden (use)". This particular door requires that the player (Jack) presses enter to open the door. You will see again that the arrow points to where the door will be located. Add the doors to the locations shown here:



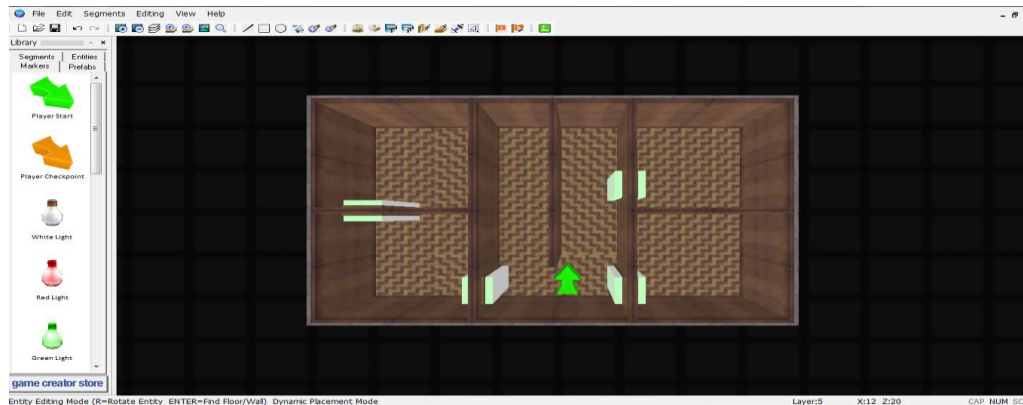
You notice that there is one room we can't enter yet. That is because that will be the way down to the basement where the bad guys have your girlfriend. That door will require the key from the 2nd floor we will create in a few minutes. Let's place the door here now though. Go back and select "add new segment" and select "Door Wooden Key" and place it in here:



Move your mouse over the door that requires the key and right click on it. See this menu on the side?

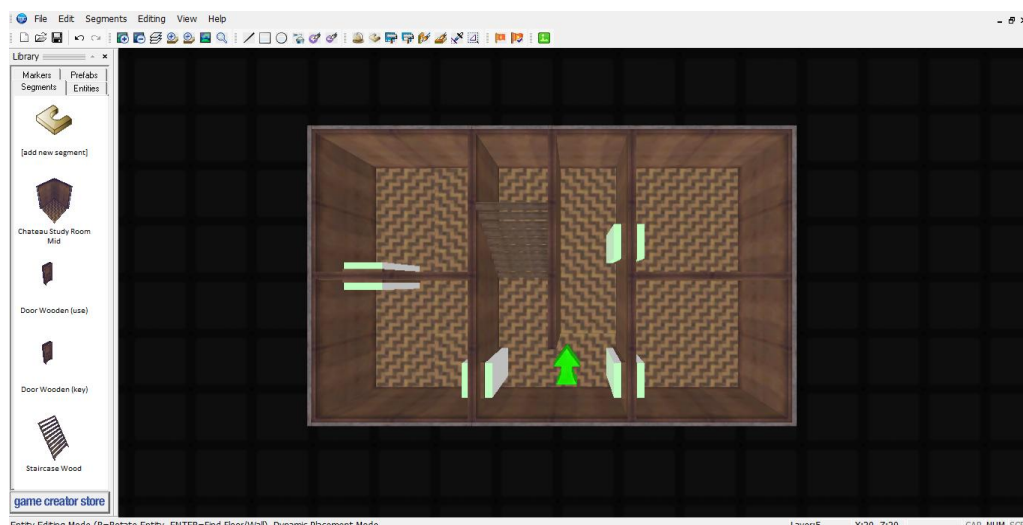
General	
Name	door_g_k
Static Mode	No
AI Scripts	
Start	appear1.fpi
Main	doorkey.fpi
Destroy	disappear1.fpi
AI Automated	
Use Key	Key
If Used	
Respawn	
Spawn At Start?	Yes
Max at any time	0
Maximum Spawn	0
Spawn after delay?	No
Spawn when dead?	No
Delay	0
Delay Random	0
Number to spawn	0
Qty Random	0
Velocity	0
Velocity Random	0
Angle	90
Angle Random	0
Spawn Life	0
Optional Visuals	
Texture	texturebank\...
Alt Texture	
Effect	
Transparency	0
Reduce Texture	0

Change the "Use key" entry to "key1". I always number my keys in the order that the player will pick them up. I use simple names like key1. I find it easier to keep track of than a name like "Jim's only door key he got from his grandma in 2nd grade." I would hate to have misspelled that and have to diagnose the problem later. We are done editing the door and can hit "apply changes". Now would be a good time to test you game. Let's change tabs to the Markers tab in the Library and select the player start arrow. Place it here:

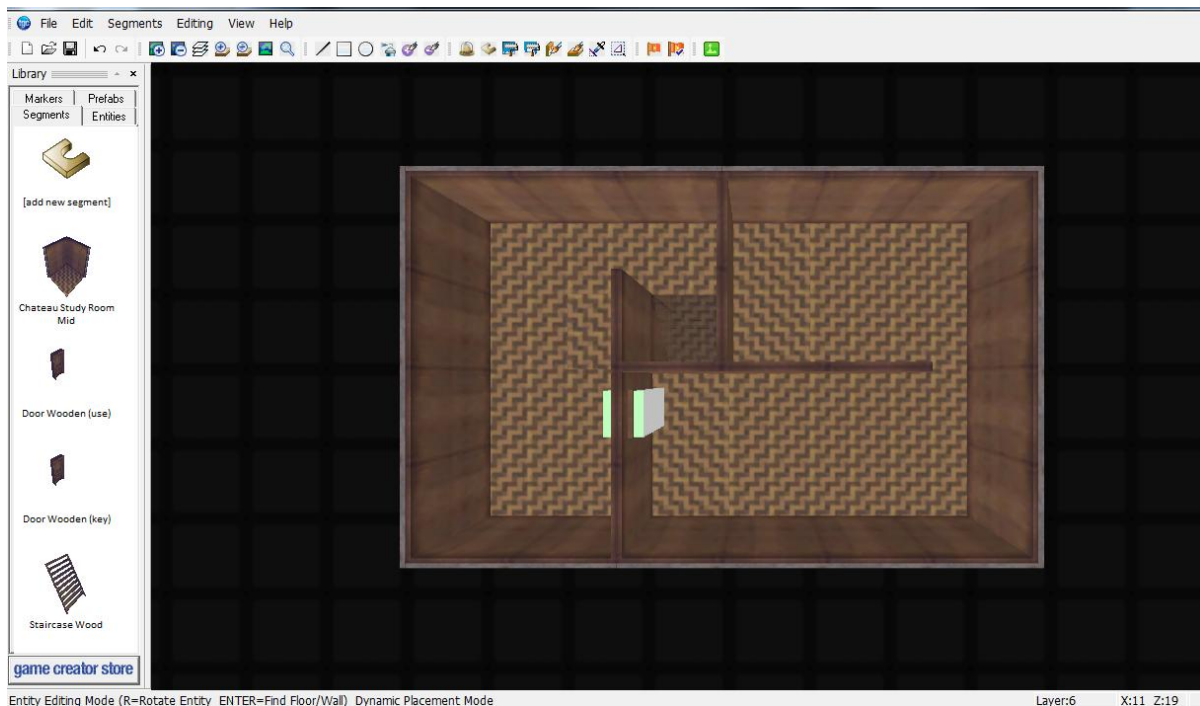


Let's also right click on the player and in '**the has**' weapon let's give it the mp40 and 4 clips.

Now save and hit the Green arrow button at the top right of the tools. You will see the waiting bar come up and the game will "compile." when this is finished press the ok button and explore the house we made a little. See how you can open the doors that do not require a key, but not the one that does? Good, because this is what we want. Hit escape when you're done testing and let's add some stairs and the second floor. Adding the stairs is easy, Go back to the segments tab in the Library and go to the WWII Platforms folder and select the "Staircase wood" and place them as seen below. The arrow will point from the lower step to the upper step.

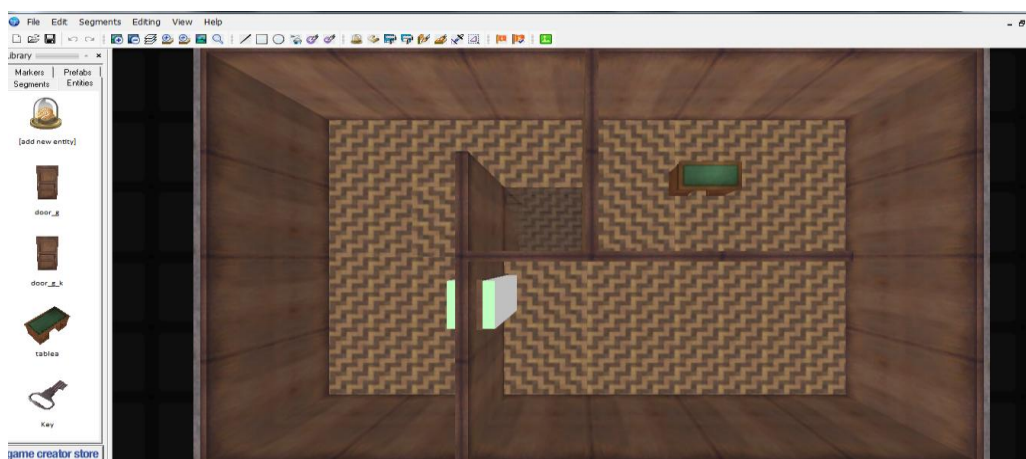


Now let's draw the 2nd floor. Press the "+" button once and go to the next floor (level 6 in our current case). Use what you've learned so far to make it look like this. Note no doors on the second floor should be locked. They are all just "Door Wooden Use."



Hey wait! How did you make the hole in the floor? Well remember how pressing the "M" key drew only walls? Well pressing the "F" key only draws floors. Move your mouse over the spot where the stairs are and right click the mouse and the floor will disappear (left clicks adds, right click deletes). Now save and run the test again and you can see you can move up stairs and throughout the whole house. Now we need to make a key to open the door.

Simply go to the left side of screen and select the entity tab and navigate to the WWII folder and **furniturec** and select **tablea** and place it in the room in the top left. Then go to the WWII-items folder and select key_1. Now move your mouse over the table and press the enter button. Pressing enter while being below the table will make the entity you have selected move to the next available surface. We should see this:

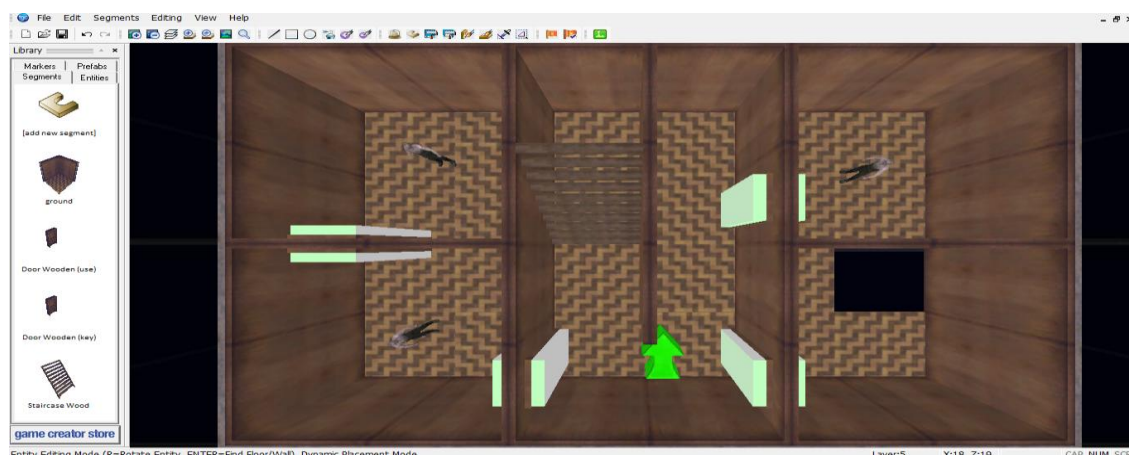


Now right click the key and edit the name to key1 so that it matches what we typed in the "if used" category of the door.

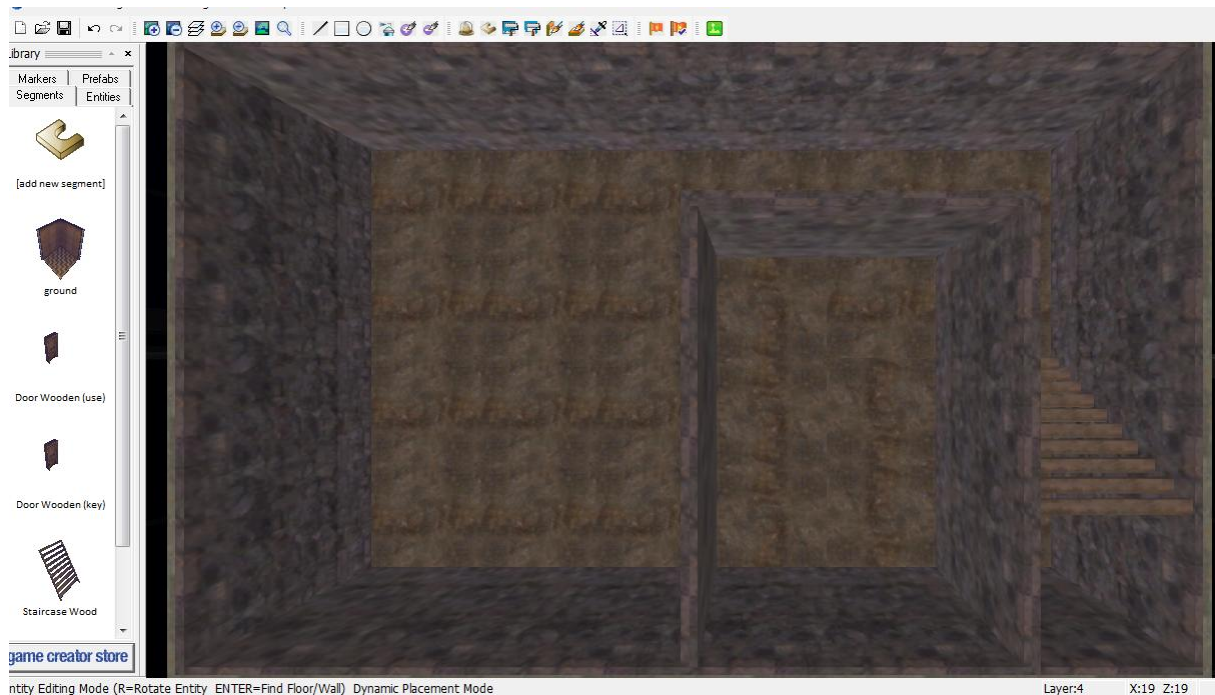
Now we need some bad guys. That's why our character is here. Use your underscore (minus) button to go back down to the 5th level (our first floor). Then go to the characters folder and select the Thug (machine pistol) and place a few here.



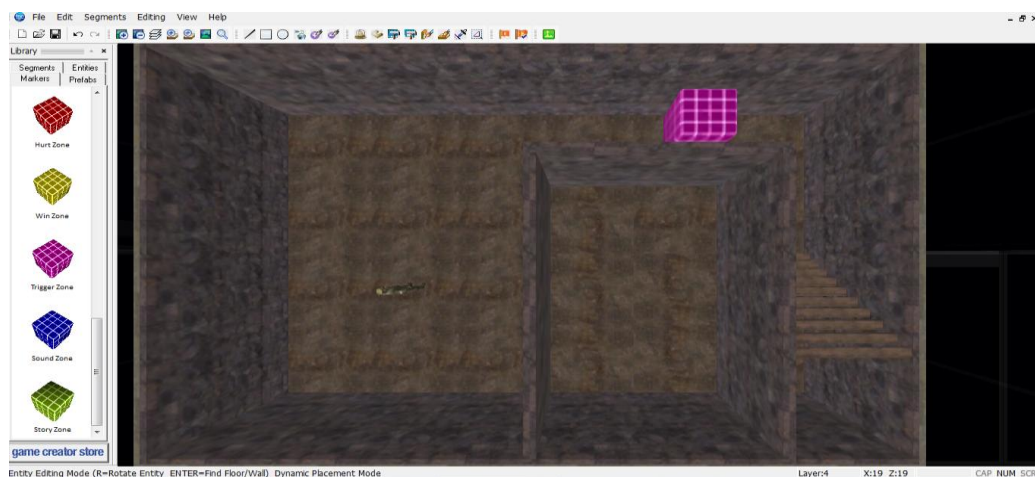
Now we have some action going on. You can add more if you want but I'm going to go ahead and start showing you how to add the basement and the boss and your girlfriend. Let's delete a segment of the floor by selecting the room segment we have been using, press the letter **F** and right click on the spot we want to delete.



Press the minus button and go to level 4 (our basement) and let's pick the chateau cellar mid segment and the wood stairs and draw this as our basement.

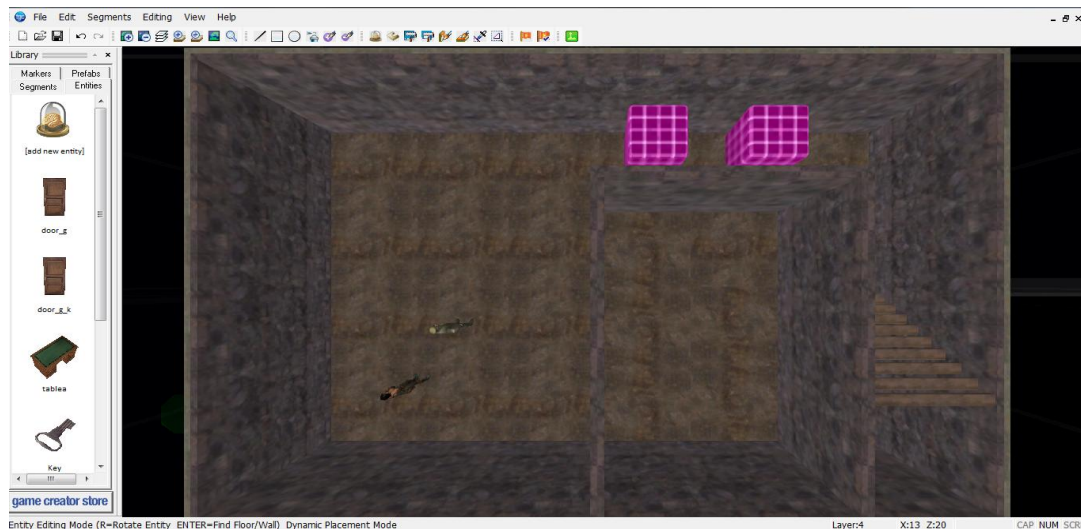


Now let's add our Boss and a trigger to spawn him. Why do we want a trigger? If we were to render all of the entities and enemies at the same time we would use a lot of resources on things we won't see for a while. Instead, by using triggers to spawn enemies, we can have more entities and stuff in the areas we are currently in and load them when we actually need them. Place the officer (machine gun) in the center of the room and go to markers and select trigger marker and place it in the "hallway" we have created.



Right click the officer and change the name to "boss1", the spawn at start to "no" and the spawn after delay to yes. Let's also change the health to 200 to make it harder for the player than the previous enemies. Click '**apply changes**' and right click on the trigger zone and change the main script to `playerinzoneactivateused.fpi` and in the if used box type boss1 for the name of our boss. I also delete the sound in the sound0 so it doesn't make any noise when the player hits it. Your own sounds can be added if needed to suit your game. Let's now add the Aiko (unarmed) from the scifi characters folder and change her name to "girl1", the main script to `people-coward.fpi`, and all the other settings to match the officer. Then

add another trigger zone next to the first one and use what we learned in the officers trigger to make her spawn after we walk into the zone also.



Hit the test button and make sure everything works the way we want it to. The player should start with the gun, be able to shoot the 3 grunts, go upstairs get the door key, go back downstairs, open the locked door and go down in the basement. The officer and the girlfriend should spawn before you round the corner and if all goes well she should run and cower from the gun shots while the officer fights you. But how does the game end? Let's go back into the officers menu by right clicking on it and change the is objective to 1. Now we have 1 objective and when that objective is completed. The level will be over. If we had made multiple levels we would then go to the next level. In our case when we have only one level, the player has won the game.

Congratulations. You have just made you first First Person Shooter. The next step would be to add more entities, light's etc. Try playing around with what already have and right clicking on the entities to see what parameters can be changed. See what you can turn this into, and most of all have fun.

FPSC: ENEMIES AND YOU

By Bugsy

Most casual FPS players don't make sure to enter every room, just to marvel at the detailed textures, they don't make note of ever box on a shelf and the time it likely took to be placed there. Most casual FPS players don't fire their guns at a wall over and over to hear the beautiful sound it makes (unless your game is bad company 2) and most casual FPS players don't play a game just because they hear it has amazing level design.

FPS players play FPS for the FPS aspect- not anything else. It's a sad truth you'll need to understand. FPS, as you REALLY SHOULD know, stands for First Person Shooter, it is a game, where the player sees from the eyes of the character they play, and that character, these days, usually stays within the stock character template for the FPS genre- Tougher than a refrigerator made of reinforced steel, grizzled enough to fit in with bears and sharks playing cards at a salty spittoon, and able to reload literally any gun in under a second, as important as that character is, it's not as important as the experience they get.

The character in the game is meant only to be a medium for the player to connect to the game world, he is that player in the game world itself, this is why usually, the less personality you give the player character, the more personality is required on the part of the player. Players are almost always happy to mix their own personality into the equation, it's why games like call of duty 4 and half life 2 did so well, the players know who they are in the world, but that's really just them under a name-de-game. Keep that in mind, while that covers the "you" aspect, the more important aspect of a SHOOTER is what I shall now move on to.

Enemies are what is usually given to the player in a game to shoot at, be they human, alien, or undead (which is a little TOO common around the forums lately), these enemies each have artificial intelligence that determines how they behave when it's killing time again. Some games have incredible AI that knows how to take cover, flank, work as a squad, and even brush their teeth in the morning to minimize plaque buildup on that pearly white teeth texture, other games (like ones made in FPSC) have less focus on the actual AI of the AI characters instead, these types of games compensate with large masses of enemies, or enemies that take a whole lot of bullets to kill.

Whether you have an amazing AI system who've memorized the Spanish dictionary and studied being cannon fodder at oxford university, or a bunch of mentally challenged, man-shaped chimpanzees on extra-drowsy cold medicine, this small guide will focus of the best ways to maximize the "fun factor" of your games firefights- be they with people or otherwise.

LIST 1: The Basics - The Core Placement Methods Often Used In Games.

THE ARENA: Usually a large room, with lots of cover and flanking routes, with enemies all about, ready to kill at all costs. This one's commonly used in tonnes of games, and it's a really great way to stretch out gameplay.

WITH GOOD AI: with good AI, try to keep this area long, but thin, maybe 35-50 feet wide and 60-90 feet long. At the far end of the room, the opposite end that the player enters, spawn about 2-4 enemies and tell them each to respawn at least 2-3 times each, this keeps the player pinned down in the room until it feels safe, at which point it usually is. It's best to make them fast movers so they can move from where they spawn and fight the player quickly, rather than a lull in combat because they take their time leisurely wandering to cover.

WITH POOR AI: poor AI makes this more difficult to create, and craft it to be convincing. It's a little more taxing to use poor AI in this case as well, as there will need to be more actual characters, instead of re-used spawns. To do it well, spawn the poor AI closer to the player, and put more spawns with less respawns, this will give the poor AI an easier time of navigating to the player, and won't allow the player to spawn camp them if they don't respawn.

PROS:

- Usually can last 1-2 minutes, which is a good amount of time.
- Fun over and over, with just subtle changes in layout.
- Works wonders when allied forces are involved.

CONS:

- Players can spawn camp the enemies if they don't pin the player down effectively. To fix this, make them do lots of damage, move fast, and spawn obstacles like explosions to go off when the player reaches the spawn area.
- Taxing- requires 4 enemies at one time. If you're using low polygon characters, it's not bad, but with high polygon characters, it adds up. To fix this, make sure to use lower polygon enemies, and keep the real detail of the area to the textures and the cover, which you will be focusing most on.
- Bodies must disappear. To fix this, keep the players eye focused away from the floor... perhaps on a timer?

AS USED IN: Tom Clancy's Rainbow 6 Vegas 2:

<http://www.youtube.com/watch?v=n5WRRzXdBSY> (go to 4:23)

THE AMBUSH (player): usually, the enemies have the drop on the player, and the player faces over whelming odds to kill them, but in this situation, the player is forced to sit and wait for his gratifying lineup of unsuspecting enemies to mow down.

WITH GOOD AI: really, this is more of a bad AI strategy, because the enemies aren't supposed to notice the player until it's too late, give them low view angles, and make sure to give the player a cue of exactly when to attack, and when not to.

WITH POOR AI: simply set waypoints, and make sure the player is out of their sight. Above them usually works well, give the player a cue of when to attack, and make sure to not let the enemies live long enough to show how awful their AI is.

PROS:

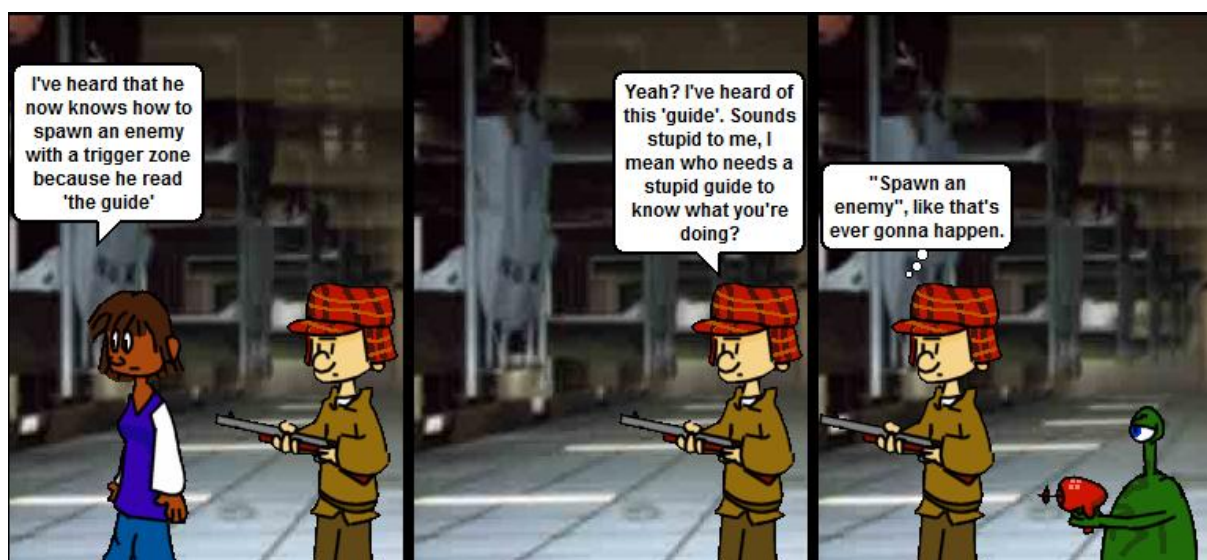
- Easy to do
- Very fun if done correctly
- Can last a while if you make the player wait

CONS:

- Not challenging. To fix this, perhaps make the player have to find a good place to hide BEFORE the enemies arrive.
- Not fast paced enough. To fix this, make sure that the pace of your game slows down to fit the scene; don't throw this in the middle of an arena firefight.
- Doesn't last. To fix this, make the anticipation outlast the encounter, this is a scene for buildup, not a gunplay skill test.

AS USED IN: BAD COMPANY 2

<http://www.youtube.com/watch?v=veYeGsvgDI0&p=96131DC6B8FCCDC2&playnext=1&index=2> (go to 2:40)



THE APPROACH: This is a fun one; it's great with good or bad AI, but best with good AI. The basic premise is that enemies are in a house, and the player is FORCED to move slowly up to the house, under heavy fire, and clear the house, it's especially fun to play.

WITH GOOD AI: the player starts outside the infiltration building, the enemies spawn inside, and fire out the windows, spawn them at the windows, where they can plainly see the player. If there are multiple stories, than it makes this scene even more epic, these enemies should keep respawning, at least twice or 4 times each, and when the player finally gets the chance to enter the house spawn more inside. Another way, is to simply make them stop spawning when the player gets inside, this is done often in call of duty 4, but make sure not to spawn them unlimitedly, as this gets annoying, and without a lull in action, the player will never get the idea that it's time to go in.

WITH POOR AI: do the same thing, however simply stopping the spawns will not work, as they most likely will spawn once at the window, and keep popping up right there, the player will then come in the house, and they won't understand to come to the player inside, making it easy for the player to walk right by, leaving enemies alive, taking up resources.

PROS:

- Epic fun that can be reused multiple times.
- Varies indoor and outdoor combat.
- Can use a relatively small area.

CONS:

- Very taxing- FPSC especially hates indoor-outdoor maps. To fix this, make the entire map indoors, but perhaps a large chamberlock wall with windows needs to be breeched.
- Difficult to construct. There is really no fixing this, if at first you don't succeed, try, try again.
- Enemies may not all die. To fix this, give them limited spawn lives.

AS USED IN: CALL OF DUTY 4

http://www.youtube.com/watch?v=kFw_NnkD0IE (go to 6:15)

THE ADVANCE/RETREAT ARENA: to basically see the general layout of this, see arena, but at the very end of the arena, the player has to go all the way back to the start. It's like having to make it to the end of a room, just to get a key to the door next to where you entered the room. It uses the same basic "arena" method.

WITH GOOD AI: see "the arena" and make sure to place a spawn trigger right over where the player will ultimately need to get to, half way through the encounter, when they have to retreat.

WITH POOR AI: see "the arena". Spawn the characters one at a time on the way back, to give them an easier time of being found by the player or make them immobile, where they'll automatically see the player.

PROS:

- Can take up an entire level's worth of gameplay.
- Can stretch out one level into a level twice as long.
- Great climax events.

CONS:

- REALLY TAXING. To fix this, make sure to optimize the static level as much as humanly possible.
- No one likes backtracking. To fix this, perhaps make different routes available to them when they go back to the beginning of the level)
- What keeps the player from just running right to the winzone? To fix this, place destroyable dynamic blockades)

SEE ALL OF THESE BASIC AI TECHNIQUES USED IN ONE LEVEL IN:

CALL OF DUTY 4

<http://www.youtube.com/watch?v=EapRJ0SndIo>

in the beginning, there's an ambush, then comes the massive arena up the mountain littered with small approaches, finally, you reach the top and learn you have to go all the way back down, which is an advance/retreat setup.

---oOo---

All of these AI techniques and more can be used, and combined for amazing scenes in your games. Here are few combo pictures:

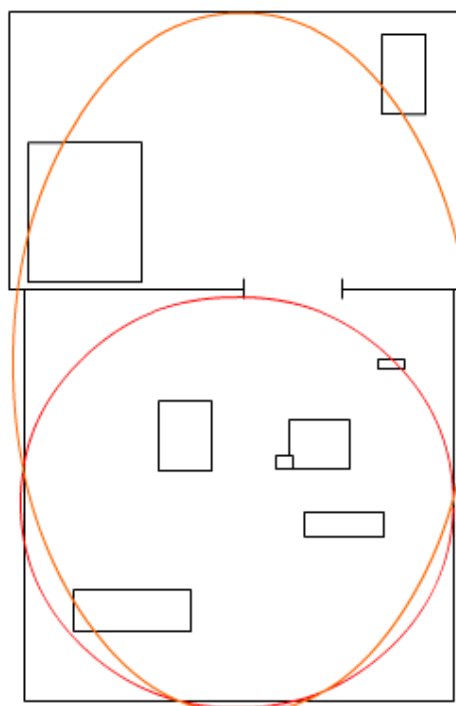
THE APPROACH/ARENA HALLWAY: This is a long arena, surrounded on both sides by approaches, giving the player a choice on how they wish to enter the encounter.

Used in: Cold war mission- BFBC2, War pig mission- COD4



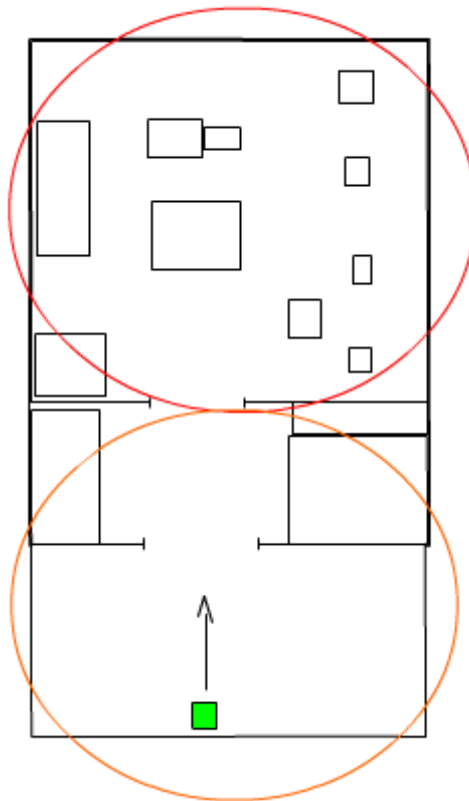
THE ARENA-APPROACH: an arena that is also an approach to an area the player needs to be, this is often a little more diluted in games, and not so plain and simple, but the basic technique is still present.

Used in: The boneyard airplane graveyard mission- MW2, Makarov's estate mission- MW2



THE APPROACHING ARENA: a simple approach to a building that contains an arena, there's a fight into and inside the building.

Used in: The Gulag mission- MW2



NUMERICAL HUD USING GIMP

By Starmind

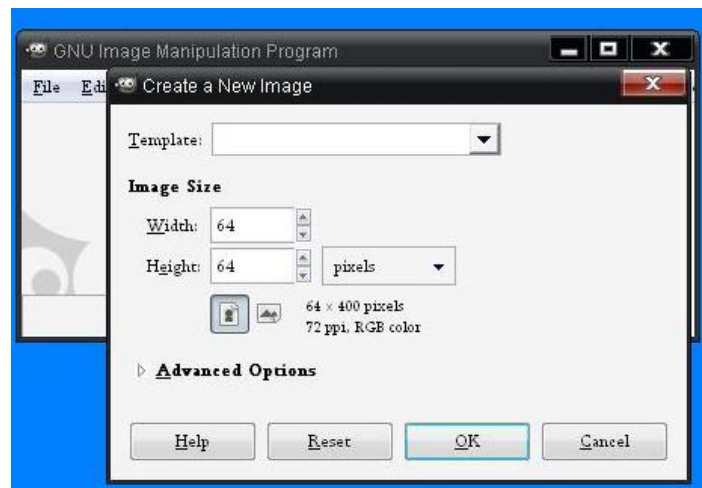
I have been asked over and over again on how to pull off a new numeric hud text. This tutorial should help those that want to do this or are new to FPSC.

For this tutorial I am using Gimp. I use this because it is free and available for everyone. You can download it [here](#).

Once you have Gimp downloaded, installed, and opened up Gimp we can begin the tutorial.

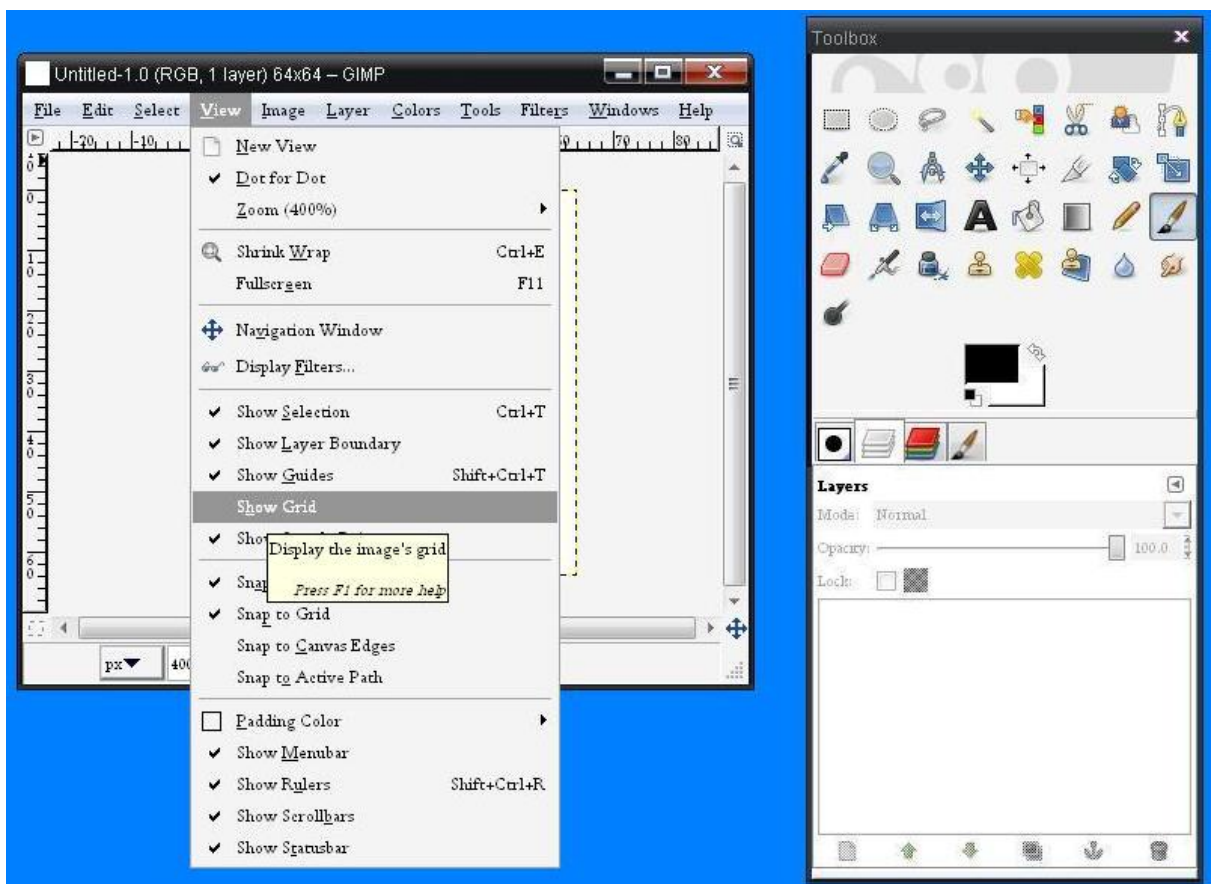
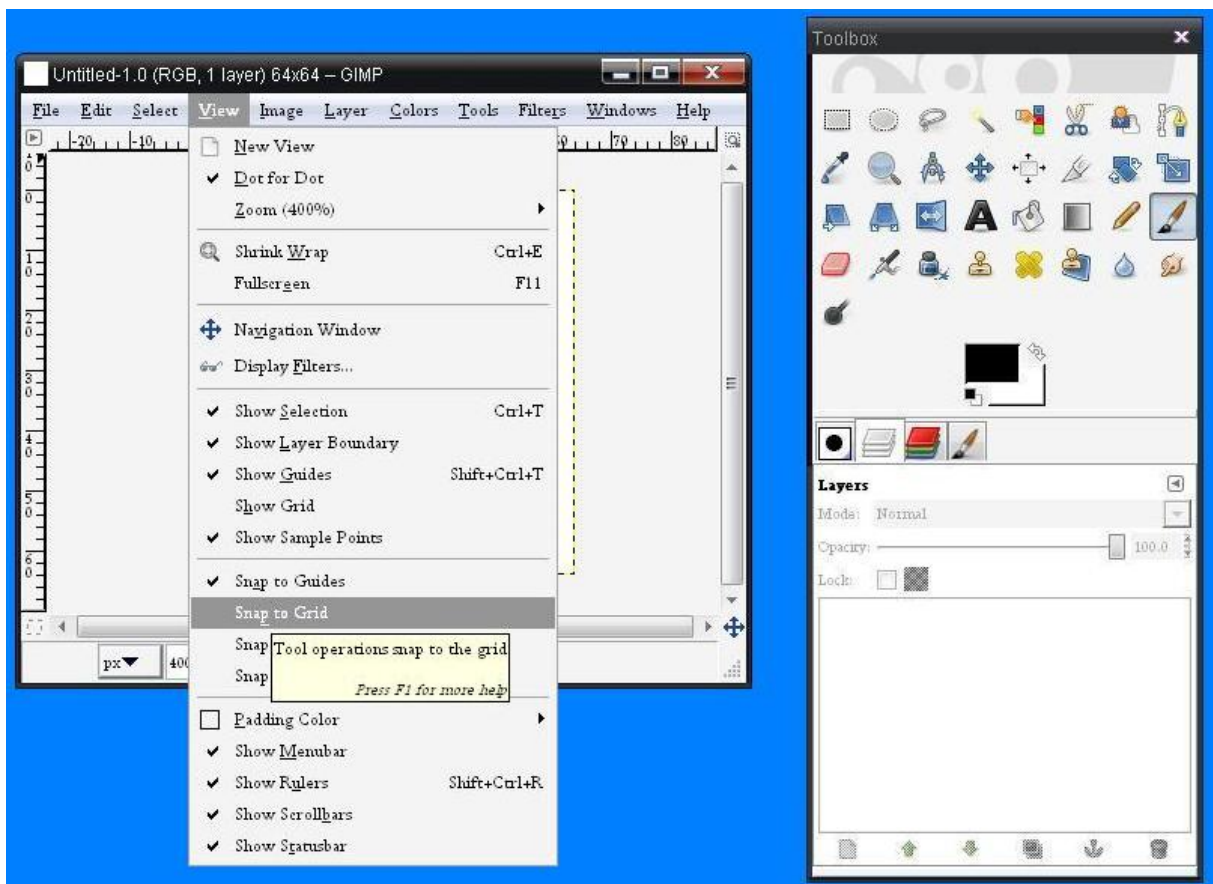
Step 1:

You will need to create a new image and scale it to 64 x 64. This will open up a new window with a white background.



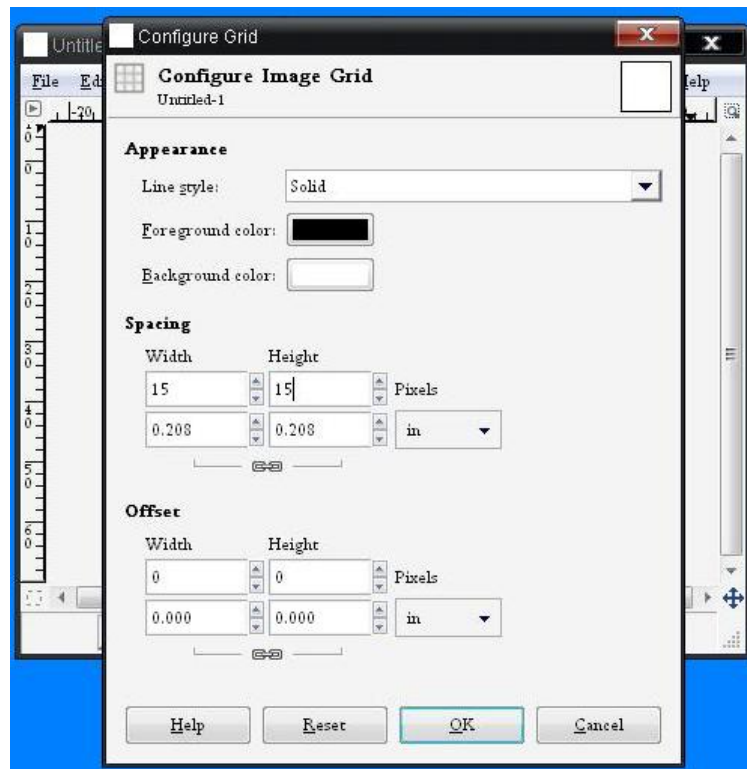
Step 2:

Go to View/Snap to Grid and click on it and it should show a little check mark. Next, in the same tab, click on Show Grid.



Step 3:

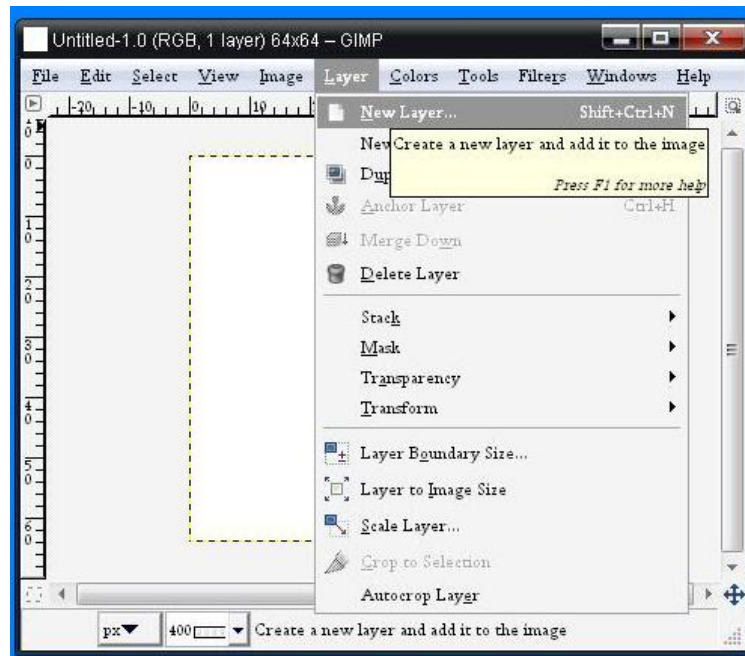
Next we need to configure the grid for it to work properly. Go to "Image -> Configure Grid". A window will open up and you want to look for "Spacing". In the width and height boxes put 15 x 15.



Step 4:

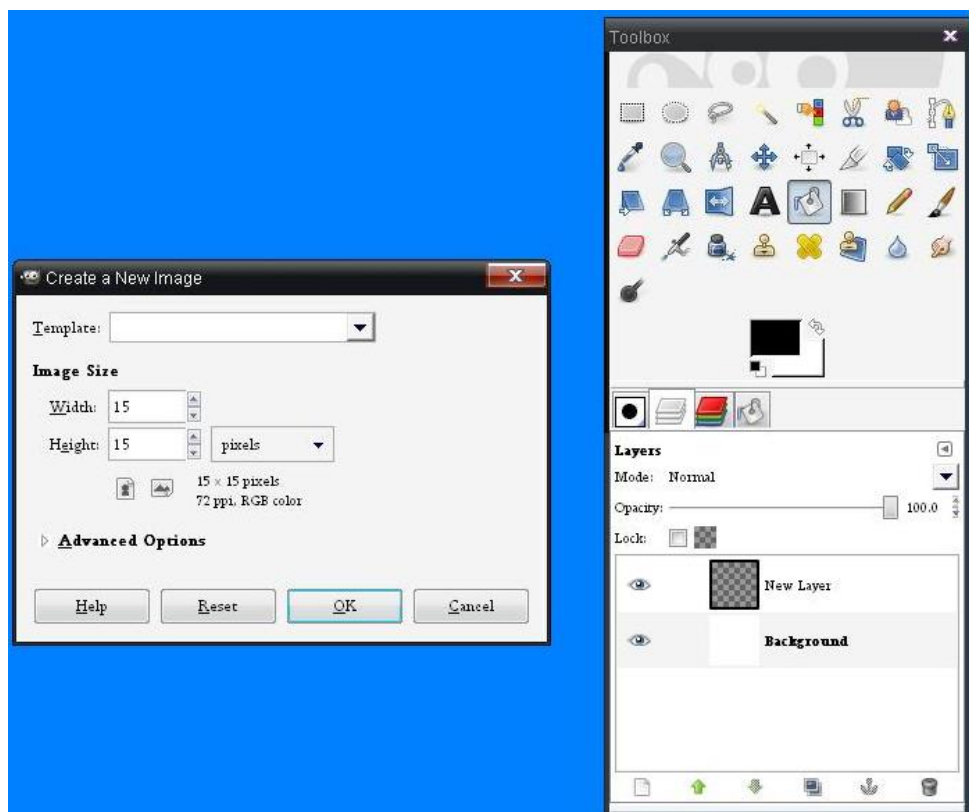
Now, you should have an image with black lined boxes and a white background. If we try to put text on a white background the numbers will look like white boxes in-game unless we add a colour to the text or we remove the background. For this tutorial we will use a transparent background.

Go to Layer/Add New Layer. Use a transparent layer. Now delete the background.



Step 5:

With our base done, we can work on the text. Go to File and create new image. This time it will be scaled at 15 x 15. Select your font and place one number in the center of that 15 x 15 image. **Be sure that this background is also transparent!** Now save that image as png or tga. The numbers should be 0,1,2,3,4,5,6,7,8,9,./ Once you have each individual image done and saved, then we can start to assemble the finished image.



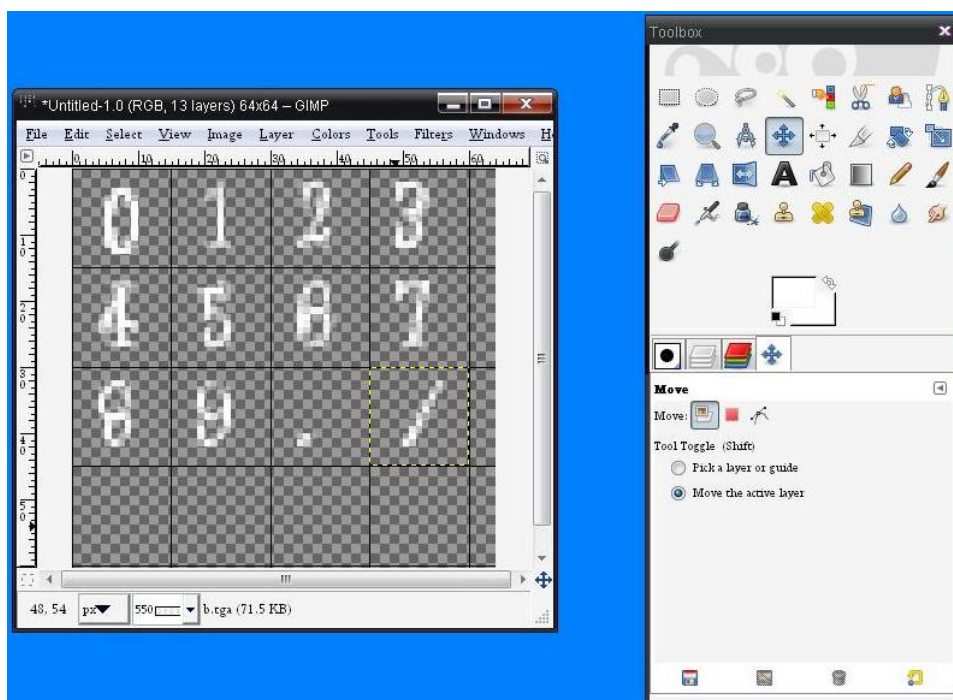
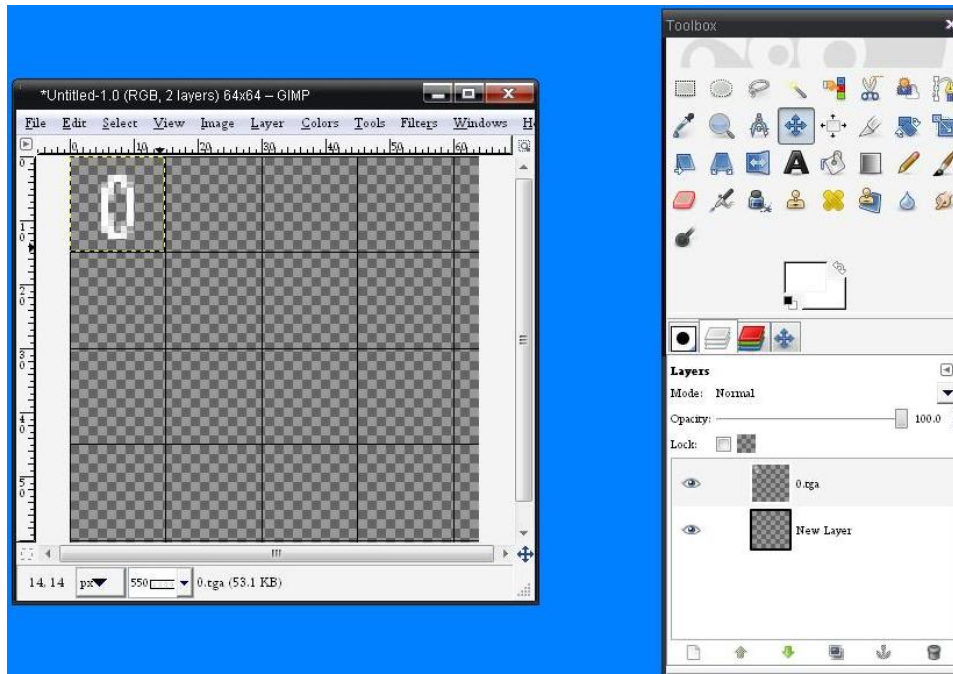
Step 6:

Now switch over to the 64 x 64 base image. Simply drag each 15 x 15 image into the base and arrange the numbers like so.

0123

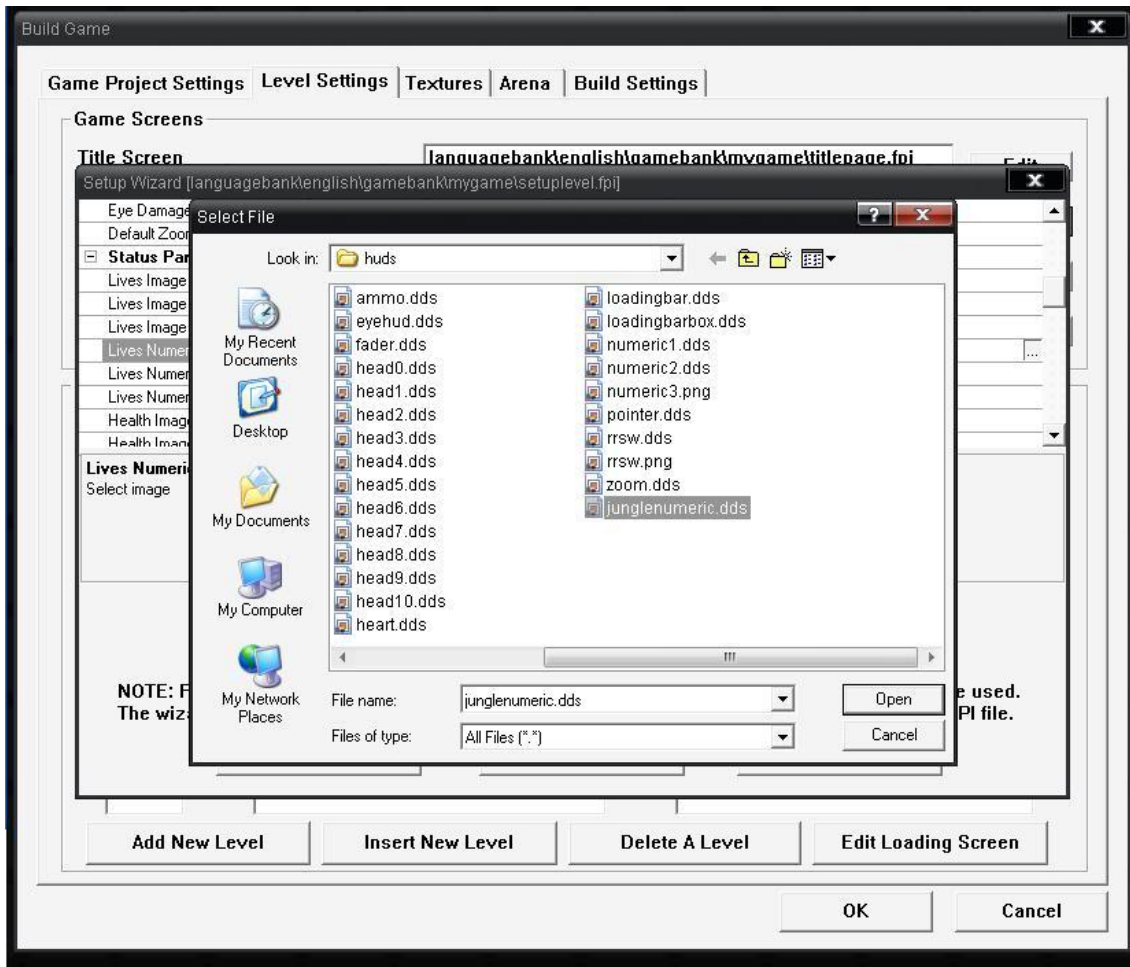
4567

89./



Step 7:

To finish merge all images and export as tga, png or dds(dxt5). Now we are ready for testing in FPSC. Start Fpsc and go to build game/setup level and whichever numeric you want to change. Once the image selection window has copy and pasted your new image into this window and press ok.



Press preview and see your new work. Should your numeric's look to small, you can do this whole process again or you can go to mygame/languagebank/setuplevel.fpi and edit it for your scale. Find your image and edit the hudx and hudy for your scale.

```
1|Script from Wizard
2|ader
3|e = Setup Wizard
4|etup Level Stats
5|ate=0:sky=skybank\BLSky\Pla
6|ate=0:music=audiobank\music\Jurassic+Park+theme+song..ogg,musicvolume=50,soundscale=25
7|ate=0:fog=1,fogred=0,foggreen=25,fogblue=25
8|ate=0:ambience=0,ambiencered=255,ambiencegreen=255,ambienceblue=255
9|
10|ate=0:hudreset,hudx=50,hudy=50,hudsize=1024,hudsizey=768,hudimage=gamecore\huds\fader.tga,hudhide=1,hudtype=3,hudm
11|ate=0:hudreset,hudx=50,hudy=50,hudred=128,hudgreen=0,hudblue=0,hudsize=1024,hudsizey=768,hudimage=gamecore\huds\ey
12|ate=0:hudreset,hudx=50,hudy=50,hudred=128,hudgreen=128,hudblue=128,hudsize=1024,hudsizey=768,hudimage=gamecore\hud
13|ate=0:hudreset,hudx=5,hudy=5,hudimage=blank.dds,hudtype=1,hudmake=display
14|ate=0:hudreset,hudx=4,hudy=8,hudsize=16,hudsizey=16,hudimage=gamecore\huds\numeric1.tga,hudtype=1,hudmake=numeric
15|ate=0:hudreset,hudx=15,hudy=5,hudimage=languagebank\english\gamecore\huds\health.tga,hudtype=2,hudmake=display
16|ate=0:hudreset,hudx=14,hudy=8,hudsize=16,hudsizey=16,hudimage=gamecore\huds\numeric1.tga,hudtype=2,hudmake=numeric
17|ate=0:hudreset,hudx=85,hudy=8,hudimage=gamecore\huds\ammo.tga,hudtype=3,hudmake=status
18|ate=0:hudreset,hudx=88,hudy=12,hudsize=16,hudsizey=16,hudimage=gamecore\huds\numeric1.tga,hudtype=3,hudmake=numeric
19|etup Game Menu
20|ate=0:hudreset,hudx=50,hudy=10,hudimage=,hudname=gamemenuitle,hudhide=1,hudmake=display
```


SETTING UP POLYGON HIT DETECTION IN FPSC

By Bond1

Want your bad guys to react differently depending where you hit them? Then FPSC (from v1.18 -Beta 5) has to ability to detect which limb is shot and react accordingly

Here is the requested tutorial on this new feature.

1. Download the preliminary limb/bone detection tool here:
http://forum.thegamecreators.com/?m=forum_view&t=180608&b=21
2. Load in a character and use your mouse to hover around the character and take note of the limb numbers, which one is an arm, leg, head, etc.
3. Using the included FPE file in the above mentioned tool as a template, find the "bodypartslimbinfo" section and use the limb numbers discovered in step one to assign them to the appropriate body part index number. Note that these are completely arbitrary: ANY limb can be a head, torso, leg, etc!
4. Stopping here will give you polygon perfect hit detection, so your aim must be perfect now! Also, assigning the "appearwithheadshot" script will automatically give you headshot kills with perfect hit precision.
5. If you want to take the feature even further, Lee has added a new FPI condition: **shotdamgetype=X** , where X is a number 1-6, corresponding to the body part being shot, as defined in the FPE file. The condition returns true if that particular body part is currently being shot.
6. Now you can do ANYTHING you want with this! For example, if an arm is shot, you can play a flinching animation then go into ragdoll. Or you can take away a certain amount of the characters health - with different body parts taking different amount of damage. You could make the character invincible until only an "achilles heel" limb is shot - great feature for BOSS fights! You could make the character limp after being shot in the leg. Since everything is user-defined, the sky is the limit!

---oOo---

LEVEL DESIGN TUT #1

By Bugsy

Designing a map for a game in FPSC is a rewarding experience, not only can you learn valuable architectural skills, (mostly from all the criticism I'll probably give you) but also, you'll get a feeling of accomplishment when you know you've improved. This tutorial is made to do just that. Read on and learn more!

When distilled to its purest form, the level is nothing but the walls and boundaries that the player walks around in. It is up to the level designer to casually lead the player by the hand in the boundaries, while letting them feel immersed, and if your game permits, confident on their own in your level. Basically, navigation should feel natural. Some games (mirror's

edge) take leading you by the hand to an extreme, and paint everything red to show where you go, whereas other games (half life 2) stealthily make the player WANT to go a certain way by making the level look more interesting in that direction. Walking into a boring part of a level that, while not bound from the player can break immersion and flow, which is why the player needs to stay constantly aware of where the right way to go is, it should feel second nature to them, letting them focus on the gameplay, that is called flow, that's what we're going to talk about next.

Flow is the levels continuity, keeping the player on the right track through the level, without even 1 second's worth of a "where do I go" feeling. if your level flows well, and is "smooth" than the player will know where to be at all times, and if they stray into a boring part of the level off the beaten path, then it is completely by choice. a well flowing, smooth level will keep the player immersed.

Immersion is a very important concept to gaming, mainly gameplay, but level design too. Immersion is basically where the player "forgets" that they're playing a video game. now while it's not that simple, it is basically a simpler term for when the player gets caught up in the experience, and essentially loses track of the world around them. It's tough to explain. good examples are:

- When you're so "in to" your game, that you don't notice (or don't care) that you accidentally knocked over a glass of Kool Aid on your new rug.
- When you finally "get" portals and realise that you've learned a new spatial awareness like none other before.
- When you pull off a triple back flip stalefish uber trick in SSX tricky and you want to "high five" your character.
- When you're so close to level 70 you don't give a crap about going to grandma's XD

Now you understand me when I say "forget you're playing a video game"

Immersion is something largely created by gameplay, but can also be created by level design. Can you imagine how much less immersed you would be if there was no level in:

Tony Hawk- have fun doing kickflips

Half life 2- It's like GMod! without spawning anything or building stuff ;(

Assassins' Creed- half the game is climbing. If there were nothing to climb, It wouldn't have made more than 10 sales.

See what I mean?

Immersion is made by having a level be as close to the real place as possible. Feel like you're on the mountain, BELIEVE that you're infiltrating a base. It's all because you're immersed. Now I'm not going to tell you how to make immersion, you can't "make" much immersion in level design, that's up to the gameplay, the level design's main goal is to not

"break" immersion. Keep your levels true to life, and you won't break any.

HOW NOT TO BREAK IMMERSION

Have you ever seen a stack of crates in a hotel room?

How about an oil barrel in a well off suburban neighbourhood?

If not, don't randomly place them in your map. If it doesn't belong, leave it out. "but I thought I was supposed to populate my levels!?" yes, populate them as best as you can, but that doesn't mean that you need to have leaking pipe in a hotel penthouse, or a table on the rough bank of a river, even if it is just for cover, or concealment, or even blocking the players view to the end of the world, there are always other options. Flow is one of the best ways level design can add to immersion, this is, if it's easy and FUN to navigate (mirrors edge, assassins' creed, infamous)

Alright, back to reality.

If you're reading this, you should understand it's still good old FPSC before dreaming up running across a rooftop in a massive city with a model pack 5 weapon in your hands, and FPSC can only handle so much. It'll break immersion when the player falls through the floor, or can't pass an object. When designing a level, keep in mind FPSC's awful collision system, sure it's nice to have planks laying around on the floor, or potted plants in a hallway, but if it means that the player has a bumpy jolty ride over the planks throughout the level, or falls through the floor because of the plant, than it's not worth it, the way I do it is: a room that doesn't look totally complete without a single floor entity (object that would rest on the floor in real life- table, crate, anything held down by gravity) then it's not complete.

Make sure to add lots of detail and "uumph" to your walls and ceilings as to not make them look so flat and "segmenty". Add wires, electrical box and outlets, lights, pipes, air ducts, chain link fences and other methods of sectioning off rooms (I count fences as non-floor centric). Play most online shooters and you'll notice that corridors are largely empty, this is to ease the players flow through the level, and the last thing you want is for the player to have a tedious, difficult task of walking around things.

Now, onto level gameplay/lag. You've got immersion, you've got flow, and you understand that you need to work against the "grain" of the editor to get good results now you need to learn how to maximize the playability of your level, this'll tell you how to reduce lag.

Lagging. It's something that every one of us, even some of the best computers will get from time to time, especially in an FPS game, that dreaded frame rate dip, that framy fight scene. It's something no developer should have to deal with, but is an issue that should be addressed when the pretty and fun part of level design is over. At this point, playtest your map, have fun in your extravagantly detailed albeit not-so optimised map. First things first, remember where you are looking at for the most part, and what is ABLE to be seen, no

lighting in that corner? no need for a breaker switch there. Never look behind you when passing that? better take that chair and barrel away. If you designed your level well, than removing these things, or "streamlining" as I call it is going to add at least 3-4 FPS to your map. Now that we've streamlined, try removing polies. if you have the resources, remove the faces from models that you'll never see the back of, take pictures of objects that you'll never get to and make "billboards" (2d images of the same objects for replacement) and replacement of certain objects for meshes less... well lower poly, don't need that fence for a cool shadow effect? replace it with the flat square with a texture, that plant making you fall through the floor anyway? just make it a billboard. Doing as much as you can to reduce lag in a map can result in at least 6-7 pickup in FPS rate. a rate which, even on the most laughable PC should never dip below 19.

That's all for this tut. If I decide to do another one, then I'll add more gameplay oriented stuff like:

- level design and enemy encounters
- reducing lag with characters in a map
- set pieces level design.
- Lighting.



A pic of a level using my classic style. (note that lighting will be covered in the next part of the tutorial)

"0 AMBIENCE" LIGHTING TUTORIAL

By Bugsy

Light is difficult to reproduce in a game, when it's really just a colour applied to surfaces. In reality, is light tangible, is there proof that light isn't just a construct of man to explain the absence of dark or vice versa? Is there even really dark? Either way, in a game, light must act the way it does in real life, like a wave, yes, light waves are transverse waves that have all of the colours of the spectrum in them, colourful lights have more or less of one spectrum colour, and no light in the universe, no matter how pure, is white, that's the first rule of FPSC lighting, unless under special circumstances, or direction of a professional, no white lights.

Transverse light waves bounce off of anything, walls included, there is no such thing as absolute darkness the same as there is no absolute light, this is an impossibility, a proper light illuminates the whole room, but can keep certain areas darker and others lighter. like so:



Notice that while the outer edges of the room are darker, and less inviting, the light remains bright in the center.

A popular style is very whitish, or vibrant lights covering a very small light zone, creating essentially the same effect I did there, however, completely unrealistic in that it does not refract off of the outer walls, this is logically impossible in nature. An example of unrealistic lighting is shown in the next image:



notice the vibrant light making an good colour, but leaving the evil darkness monster to engulf the rest of the room, this is impossible in reality.

The proper way to create excellent lighting refractions and reflections, and keep the number of lights down, is to use a very dark lighting colour, and give it a big range. I won't be naming colours here, but if done correctly, a very large dark light can illuminate an area unbelievably well, and also create a vibrant colour near its center.

PARTICLE DECAL SYSTEM

By Lee Bamber

Built on the foundation of the existing decal system, a new type of decal allows the spawning of particles and those particle properties can be changed to suit your game. Below is a step by step guide to creating your own particle effects:

1. Start the software with a blank level and create a simple floor
2. Go to Add New Entity and select GENERIC / DECALS / DUSTPUFF
3. Add this decal to the scene, then right click it to enter the properties dialog
4. Change the main AI script from DECAL.FPI to DECALPARTICLE.FPI
5. Leave CUSTOM SETTINGS to YES
6. Change the values in the Decal Particles group then run test game to see results

Decal Particle Properties

- **offsety** : Vertical adjustment of the start position
- **scale** : A value from 0 to 100, denoting size of particle
- **randomstartx** : Random start area
- **randomstarty** : Random start area
- **randomstartz** : Random start area
- **linearmotionx** : Constant motion direction
- **linearmotiony** : Constant motion direction
- **linearmotionz** : Constant motion direction
- **randommotionx** : Random motion direction
- **randommotiony** : Random motion direction
- **randommotionz** : Random motion direction
- **mirrormode** : Set to one to reverse the particle
- **camerazshift** : Shift particle towards camera
- **scaleonlyx** : Percentage X over Y scale
- **lifeincrement** : Control lifespan of particle
- **alphaintensity** : Control alpha percentage of particle

---oOo---

GETTING CHARACTERS TO CLIMB STAIRS

By Lee Bamber

Now your character is running around happily, you will soon realise they are not following you up and down stairs as the original characters once did. This default behaviour was intended, allowing you control over whether characters could follow you into new sections of your level. Granting the ability of characters to move up and down stairs is relatively simple by following the steps below:

1. Follow the steps outlined in 'Getting Characters To Think' to create a simple A vs E level

2. Click Add New Segment and select SCIFI / PLATFORMS / STAIRCASE METAL
3. Add a staircase in the centre of the level floor you created earlier
4. Move up one layer by pressing the PLUS [+] key and then select the GROUND segment
5. Paint a second floor that meets the top of the staircase with enough room to run around
6. Now click the 'Create New Waypoint' icon (third from the right in the toolbar)
7. This will create a waypoint node. Click and hold the node and drag it to the centre of the segment that meets the top of the staircase (do not place it right on the edge at the top of the stairs as you want the node to represent the floor segment, not the staircase)
8. Hold down SHIFT key and then left click on the node you just dragged
9. It will create a second node linked to the first. As before, click and hold this second node and then (without releasing the left mouse button), press the MINUS [-] key.
10. Whilst still dragging the second node, position it in the centre of the segment that represents the bottom of the stairs (not on the stairs, or below the stairs, but the floor segment that the player would stand on once they have fully descended the stairs)
11. Now click the test game, recruit the ally (by pressing G) and run up the stairs. Your ally should follow you, and if the enemy catches sight of you, so should he!

---oOo---

GETTING CHARACTERS TO GO THROUGH CERTAIN DOORS

By Lee Bamber

Now your new DarkAI character is following you almost everywhere, you may have noticed that they stop at certain door segments and proceed no further. This is the intended behaviour, and is the result of the way the stock doors have been designed. By default most stock doors will allow the character to pass though, but some doors will prevent entrance.

If you are a designer of your own door segments, you must bear in mind that only doors that use a CSG punch mesh that starts with the four letters "door" will instruct the DarkAI obstacle system to recognise the segment as a door and allow characters to pass though. In the segment FPS file, here is an example of a correctly named CSG punch mesh which allows DarkAI to recognise this door:

```
partmode2 = 1
meshname2 = meshbank\scifi\scenery\doors\door_b\door_b_punch.x
```


GETTING CHARACTERS TO TALK

By Lee Bamber

1. Locate this folder - files/audiobank/darkvoices. This has a converter app in it called **ConvertAllWAVFiles.exe**
2. Now place all of your wav files you wish to convert to lipsync files inside this folder
3. Execute the convert app
4. All of the wave files will now have an associated lipsync file for it
5. Place the wav files along with the lipsync files wherever you want to access them
6. Now open up your talking characters fpe file. You will see the following

```
;ai
Aiinit      = appear1.fpi
Aimain      = behaviours\greet player.fpi
Aidestroy   = destroy\fadecorpse.fpi
Soundset    = male
soundset1   = audiobank\darkvoices\canihelpu.wav
```

7. Point soundset1 to the talking wav file you wish to use
8. Use the fpi file that is in scriptbank/behaviours folder. This is just a generic (template) fpi for talking characters. You will probably want to edit it for your own use of the character
9. If you are having problems generating a LIPSYNC file, try downloading the SAPI SDK from Microsoft at: <http://download.microsoft.com/download/B/4/3/B4314928-7B71-4336-9DE7-6FA4CF00B7B3/SpeechSDK51.exe>
10. Be aware that only characters from the default stock character which are entitled (Talker) can use the lipsync data to animate the mouth, older stock characters cannot as they would require extra bone data in the head of the character model. Assume that all FPSC characters cannot use the lipsync data unless they have been expressly designed with the additional mouth bone system in mind. For reference, all characters in model packs 44 and 45 allow full lipsync control.

---oOo---

GETTING CHARACTERS TO THINK

By Lee Bamber

Characters have new powers of intellect thanks to the introduction of DarkAI into FPSC, and with a few tweaks to their properties can become allies, enemies, neutral or grouped into factions. They also have enhanced path finding skills and an increased sense of survival and combat. Below is a step by step procedure to activating a character to use the new DarkAI system:

1. Start the software with a blank level with a simple floor
2. Click Add New Entity and select SCIFI / CHARACTERS / AI (Shotgun)
3. Add this character to your level twice, some distance from each other

4. Right click the first character and change the appear AI script from **appear1.fpi** to **Dark AI/ appear-ally-team1.fpi** and change the main AI script from **People/Pace10.fpi** to **Dark AI/ main-weapon.fpi**
5. Also delete the shoot AI script entry as this is not used by the above scripts
6. Finally click Apply Changes to finish configuring this new AI Ally character
7. Repeat steps 4-6 with the second character, but instead of choosing **appear-ally-team1.fpi** you chose **appear-enemy-team1.fpi**
8. You can now run the test game, run up to the Ally, press G to recruit and attack the enemy.



---oOo---

DARK AI

Dark AI FPI Reference

FPI COMMAND	TYPE	DESCRIPTION
addaiteam=x	Action	X = The team number (1 - 20 range cap). Team 1 is defaulted to the player. Can be called at any time to change teams.
aiaction=x	Condition	X = 0 if idle; X = 1 if attacking; X = 2 if in cover; X = 3 if retreating.
aiaddally=x y y y y y	Action (Faction Control)	Y= Dark AI team(s) to add as allies to the Dark AI team 'X' is equal to. Allied teams share AI calls.
aiaddenemy=x y y y y y	Action (Faction Control)	Y= Dark AI team(s) to add as enemies to the Dark AI team 'X' is equal to. Enemy teams are targetable.
aiaddneutral=x y y y y y	Action (Faction Control)	Y= Dark AI team(s) to add as neutral to the Dark AI team 'X' is equal to. Neutral teams are ignored.
aiaddpoint=x	Condition	X is the points ID number. This will create a point to be used with other AI point commands.
aiatcover=x	Condition	X = 1 if in cover, X = 0 if exposed.
aiautofactionoff=x	Action	If X = 1, then automatic faction disposition reciprocity is disabled for the 'Faction Control' actions, allowing asymmetric disposition.
aicalled=x	Condition	X = 1 or 0
aicalledbyplr=x	Condition	X = 1 or 0
aicallteam=x	Action	X = Range of call broadcast from entity.
aicanshoot=x	Condition	X = 1 is true if Dark AI can see an enemy, X = 0 is true if Dark AI cannot see an enemy.
aicleartarget	Action	Clears the AI's target list
aicondsoundfinished	Condition	Allows sequence of WAV sounds to be played inside FPI scripts.
aifollowplr=x	Action	X = 1 tells Dark AI ally entities to follow the player. X = 0 tells Dark AI ally entities to stop following the player.
aigotopoint=x	Condition	This will make the entity go from its current position to the point specified by 'X'. This path will avoid any obstacles.
aiheardsound=x	Condition	X = Unit range for Dark AI to listen for sounds (default 700 if no 'X' value is given).
ailookaround=x,y	Action	Tells the AI to turn to a random heading between x and y degrees offset from its current heading.
aimoverandom=x	Action	X = 1 allows Dark AI entities to move about randomly (must be looped or entity will stop moving after a while).
aimovetocover=x	Action	If X = 1 then Dark AI will seek the nearest trigger zone with the name "Safe Zone". If X = 0 Dark AI will seek the nearest cover it can find but if that cover is occupied it will seek the closest trigger

		zone with the name "Safe Zone".
aimovetosound	Action	Instructs Dark AI to pathfind to the location of the sound it heard.
aimovetotarget	Action	Instructs Dark AI to pathfind to its current target.
aionpoint=x	Action	X = The points ID number. This will check if the entity is on the point named by 'X'.
aiplrallteam=x	Action	X = Range of call broadcast from player.
airemove	Action	Completely removes the entity from the Dark AI system, allowing it to function as standard AI.
airespondtocal	Action	AI moves to location where 'aicallteam' was last heard.
airespondtoplrcall	Action	AI moves to location where 'aiplrallteam' last heard.
airotatetosound	Action	AI rotates to face sound.
airotatetotarget	Action	AI rotates to face target.
aisetmeleedamage=x	Action	Sets the amount of damage to be inflicted by 'aiusemelee'.
aisetspeed=x	Action	Sets the movement speed for the entity to the value of 'X'.
aistop	Action	Dark AI will halt and clear any current pathfinding instructions.
aitargetdistfurther=x	Condition	Dark AI target is further than a distance of 'X' units.
aitargetdistwithin=x	Condition	Dark AI target is within a distance of 'X' units.
aiteam=x	Action	X = Team number defined by 'addaiteam=#'
aiusemelee=x	Action	X = 1 then blood shows, X = 0 then blood does not show.
animationnormal	Action	Normal animation playback.
animationreverse	Action	Reversed animation playback.
ducking=x	Condition	X = 1 is true if Dark AI is ducking, X = 0 is true if Dark AI is not ducking.
healthgreater=x	Condition	X = Entity health.
idle=x	Condition	X = 1 is true if Dark AI is moving, X = 0 is true if Dark AI is not moving. (Used mainly for setting animations)
isdarkai=x	Condition	X = 1 if the entity has been loaded into the Dark AI system via 'addaiteam=X'.
movingbackwards=x	Condition	X = 1 is true if Dark AI is moving backwards, X = 0 is true if Dark AI is not moving backwards. (Used mainly for setting animations)
movingforwards=x	Condition	X = 1 is true if Dark AI is moving forwards, X = 0 is true if Dark AI is not moving forwards. (Used mainly for setting animations)
runningforwards=x	Condition	X = 1 if Dark AI is running, X = 0 if Dark AI is not running (used mainly for setting animations, Dark AI internally dictates when to run or not).
setaiactive=x	Action	X = 1 when Dark AI is activated, X = 0 when Dark AI is de-activated. Used mostly for Dark AI animation override when reloading, etc.
strafingleft=x	Condition	X = 1 is true if Dark AI is strafing left, X = 0 is true if Dark AI is not strafing left.

strafingright=x	Condition	X = 1 is true if Dark AI is strafing right, X = 0 is true if Dark AI is not strafing right.
-----------------	-----------	---

---oOo---

Model Animation list – Normal and TF341 (model pack 53)

- Animations in **ALL CAPS** are the new TF341 animations and are not found in the FPSC stock animation set.
- **WT** refers to the "weapontype" field assign in the weapon's gunspec.txt settings.
- Missing Anim numbers means there is no anim for that number.

ANIM	NAME	WT0	WT1	WT2	WT3	WT4	WT5
Anim0	Spawn	3001,3002					
Anim1	Idle	712,736					
Anim2	Move Slow	738,760					
Anim3	Strafe left	762,781					
Anim4	Strafe Right	782,781					
Anim5	Move Fast	803,819					
Anim6	Throw	821,880					
Anim7	Climb	3211,3231					
Anim8	MELEE PUNCH	3063,3083					
Anim9	MELEE UPPERCUT	3084,3104					
Anim11	Impact Front	0,19					
Anim12	Bounce Front	20,,39					
Anim13	Get Up Front	3233,3272					
Anim14	Impact Back	40,59					
Anim15	Bounce Back	60,79					
Anim16	Get Up Back	3273,3302					
Anim17	Impact Left	120,139					
Anim18	Bounce left	140,159					
Anim20	Impact Right	80,99					
Anim21	Bounce Right	100,119					
Anim23	MELEE KICK	3105,3125					
Anim24	TRANSITION CROUCH	882,890					
Anim25	TRANSITION STAND	3324,3333					

ANIM	NAME	WT0	WT1	WT2	WT3	WT4	WT5
Anim26	MELEE STAND DEFENCE	3126,3142					
Anim27	MELEE STAND LEFT PUNCH	3185,3201					
Anim28	SHIEL D BACKHAND	349,365					
Anim29	MELE E STAND AIM DOWN	3202,3205					
Anim30	MELEE STAND AIM UP	3206,3209					
Anim31	Crouched Idle	892,915					
Anim32	Crouched Move	917,939					
Anim33	STACK IDLE	662,686					
Anim34	STACK MOVE SLOW	688,710					
Anim35	SHIELD CROUCH ADVANCE	3336,3357					
Anim36	Crouched Throw	941,1000					
Anim37	MELEE DEFEND IDLE	3129,3138					
Anim38	SHIELD CROUCH IDLE	3359,3360					
Anim39	SHIEL D CROUCH IDLE 2	3361,3362					
Anim40	GUAR D MOVE FAST	3304,3321					
Anim42	Recover Back	3233,3272					
Anim43	Recover Face	3273,3302					
Anim44	FLINCH GUT	3003,3022					
Anim45	FLINCH LEFT	3023,3042					
Anim46	FLINCH RIGHT	3043,3062					
Anim47	JUMP	3364,3388					
Anim48	GUARD IDLE	612,636					
Anim49	GUARD MOVE SLOW	638,660					

ANIM	NAME	WTO	WT1	WT2	WT3	WT4	WT5
Anim50	Weapon Spawn		3390, 3391	3779, 3780	4168, 4169	4557, 4558	4946, 4947
Anim51	Weapon Idle		1101, 1125	1490, 1514	1879, 1903	2268, 2292	2657, 2681
Anim52	Weapon Move Slow		1127, 1149	1516, 1538	1905, 1927	2294, 2316	2683, 2705
Anim53	Weapon Strafe Left		1151, 1170	1540, 1559	1929, 1948	2318, 2337	2707, 2726
Anim54	Weapon Strafe Right		1171, 1190	1560, 1579	1949, 1968	2338, 2357	2727, 2746
Anim55	Weapon Move Fast		1192, 1208	1581, 1597	1970, 1986	2359, 2375	2748, 2764
Anim56	Weapon Reload		1210, 1269	1599, 1658	1988, 2047	2377, 2436	2766, 2785
Anim57	Weapon Climb		3600, 3620	3989, 4009	4378, 4398	4767, 4787	5156, 5176
Anim58	Weapon MELEE PUNCH		3452, 3472	3841, 3861	4230, 4250	4619, 4639	5008, 5028
Anim59	Weapon MELEE UPPERCUT		3473, 3493	3862, 3882	4251, 4271	4640, 4660	5029, 5049
Anim61	Weapon Impact Front	1,19					
Anim62	Weapon Bounce Front	20,39					
Anim63	Weapon Get Up Front		3622, 3661	4011, 4050	4400, 4439	4789, 4828	5178, 5217
Anim64	Weapon impact Back	40,59					
Anim65	Weapon Bounce Back	60,79					
Anim66	Weapon Get Up Back		3662, 3691	4051, 4080	4440, 4469	4829, 4858	5218, 5247
Anim67	Weapon impact Left	120,139					
Anim68	Weapon Bounce Left	140,159					
Anim70	Weapon Impact Right	80,99					
Anim71	Weapon Bounce Right	100,119					
Anim73	Weapon MELEE KICK		3494, 3514	3883, 3903	4272, 4292	4661, 4681	5050, 5070
Anim74	Weapon TRANSITION CROUCH new		1271, 1279	1660, 1668	2049, 2057	2438, 2446	2827, 2835
Anim75	Weapon TRANSITION STAND new		3713, 3722	4102, 4111	4491, 4500	4880, 4889	5269, 5278

ANIM	NAME	WTO	WT1	WT2	WT3	WT4	WT5
Anim76	Weapon STAND COCK		3515, 3531	3904, 3920	4293, 4309	4682, 4698	5071, 5087
Anim77	Weapon STAND RECOIL SEMI		3570, 3577	3959, 3966	4348, 4355	4737, 4744	5126, 5133
Anim78	Weapon STAND RECOIL BURST		3578, 3590	3967, 3979	4356, 4368	4745, 4757	5134, 5146
Anim79	Weapon STAND AIM DOWN		3591, 3594	3980, 3983	4369, 4372	4758, 4761	5147, 5150
Anim80	Weapon STAND AIM UP		3595, 3598	3984, 3987	4373, 4376	4762, 4765	5151, 5154
Anim81	Weapon Crouched Idle		1281, 1304	1670, 1693	2059, 2082	2448, 2471	2837, 2860
Anim82	Weapon Crouched Move (same)		1306, 1328	1695, 1717	2084, 2106	2473, 2495	2862, 2884
Anim83	Weapon STACK IDLE		1051, 1075	1440, 1464	1829, 1853	2218, 2242	2607, 2631
Anim84	Weapon STACK MOVE SLOW		1077, 1099	1466, 1488	1855, 1877	2244, 2266	2633, 2655
Anim86	Weapon Crouched Reload		1330, 1389	1719, 1778	2108, 2167	2497, 2556	2886, 2945
Anim87	Weapon CROUCH COCK		3532, 3548	3921, 3937	4310, 4326	4699, 4715	5088, 5104
Anim88	Weapon CROUCH RECOIL SEMI		3549, 3557	3938, 3946	4327, 4335	4716, 4724	5105, 5113
Anim89	Weapon CROUCH RECOIL BURST		3558, 3568	3947, 3957	4336,4 346	4725,473 5	5114,51 24
Anim90	Weapon GUARD MOVE FAST		3696, 3710	4085, 4099	4474,4 488	4863,487 7	5252,52 66
Anim92	Weapon Recover Back		3622, 3661	4011, 4050	4400, 4439	4789, 4828	5178, 5217
Anim93	Weapon Recover Face		3662, 3691	4051, 4080	4440, 4469	4829, 4858	5218, 5247
Anim94	Weapon FLINCH GUT		3392, 3411	3781, 3800	4170, 4189	4559, 4578	4948, 4967
Anim95	Weapon FLINCH LEFT		3412, 3431	3801, 3820	4190, 4209	4579, 4598	4968, 4987

ANIM	NAME	WTO	WT1	WT2	WT3	WT4	WT5
Anim96	Weapon FLINCH RIGHT		3432,3 451	3821,3 840	4210,4 229	4599,461 8	4988,50 07
Anim97	Weapon JUMP		N/A				
Anim98	Weapon GUARD IDLE		1001,1 025	1390,1 414	1779,1 803	2168,219 2	2557,25 81
Anim99	Weapon GUARD MOVE SLOW		1027,1 049	1416,1 438	1805,1 827	2194,221 6	2583,26 05
Anim100	TALK NEUTRAL 1	5368,5390					
Anim101	TALK NEUTRAL 2	5391,5413					
Anim102	TALK RADIO PTT START	5336,5342					
Anim103	TALK RADIO PTT IDLE	5342,5343					
Anim104	TALK RADIO PTT LOOP	5343,5360					
Anim105	TALK RADIO PTT END	5361,5367					
Anim106	TALK RADIO PTT FULL	5336,5367					
Anim108	GROUND SLUMP IDLE	38,39					
Anim109	GROUND SLUMP GET UP	3249,3271					
Anim110	MELEE SWORD SLASH	400,430					
Anim111	MELEE SWORD STAB	431,461					
Anim112	MELEE SWORD RIPOSTE	462,500					
Anim113	MELEE SWORD PARRY	501,519					
Anim114	MELEE AXE CHOP	520,549					
Anim116	SHIELD TRANSITION	341,348					
Anim118	SHIELD IDLE	365,366					
Anim119	SHIELD FLINCH	366,374					
Anim120	SHIELD ADVANCE	376,398					

ANIM	NAME	WTO	WT1	WT2	WT3	WT4	WT5
Anim121	SHIELD THROW NADE	581,609					
Anim123	SIT CHAIR DIE	268,280					
Anim124	SIT CHAIR FLINCH 1	321,329					
Anim125	SIT CHAIR FLINCH 2	330,339					
Anim131	COWER TO	2947,2957					
Anim132	COWER IDLE	2958,2988					
Anim133	COWER FLINCH	552,570					
Anim134	COWER FROM	2989,2999					
Anim135	SIT CHAIR STAND-UP	281,290					
Anim136	SIT CHAIR SIT-DOWN	291,320					
Anim137	SIT CHAIR	190,215					
Anim138	SIT CHAIR KBD	216,241					
Anim139	SIT CHAIR KBD TYPE	242,267					
Anim140	FEMALE WALK	5416,5438					
Anim141	FEMALE IDLE	5442,5448					
Anim142	FEMALE IDLE 2	5451,5457					
Anim148	D1	5460,5478					
Anim149	D2	5481,5499					
Anim150	Weapon TALK NEUTRAL1		3755,3 777	4144,4 166	4533,4 555	4922,494 4	5311,53 33
Anim151	Weapon TALK NEUTRAL2 (same)		3755,3 777	4144,4 166	4533,4 555	4922,494 4	5311,53 33
Anim152	Weapon TALK RADIO PTT START		3723,3 729	4112,4 118	4501,5 507	4890,489 6	5279,52 85
Anim153	Weapon TALK RADIO PTT IDLE		3729,3 730	4118,4 119	4507,4 508	4896,489 7	5285,52 86
Anim154	Weapon TALK RADIO PTT LOOP		3730,3 747	4119,4 136	4508,4 525	4897,491 4	5286,53 03

ANIM	NAME	WT0	WT1	WT2	WT3	WT4	WT5
Anim155	Weapon TALK RADIO PTT END		3748,3 754	4137,4 143	4526,4 532	4915,492 1	5304,53 10
Anim156	Weapon TALK RADIO PTT FULL	3723,3754	4112,4 143	4501,4 532	4890,4 921	5279,531 0	

Hints & Tips

FPSC GAME ENGINE OPTIMIZATION

By JHR & Red Eye

General Optimisation Tips

Tip by: Lee Bamber

By adding a few walls, and adjust the logic of the characters I smoothed out an entire game so it never dropped below 30fps. The performance meters (press TAB) will help understand where the drain is.

The general rule for great FPS is use the [and] square bracket keys to toggle wireframe on and off, see where lots of the scene is being rendered, and use walls and corners to block off all views to the portal that links that part of the map (the portal is the invisible hole such as a doorway, window or turning in a large room that separates the large areaboxes that make up the entire level.

Reduce the number of area boxes that must be rendered and your FPS shoots up because if the entity within the area box is thinking, and the area box is not entered, the entity does not have to do as much thinking. Another quick tip, go into your characters and look for the appear1.fpi script name. Click to change this FPI script name to a new script found in the PEOPLE folder called appearnofloorlogic.fpi. You can use this if you KNOW that your character is not going to fall down any holes or climb any stairs.

Level Design Tips

Tip by: Gorba Flamingo

These level design tips will help improve performance in your games.

- Use as few entities as possible. Will the player notice this thing if he is sprinting past it?
- Re-use areas, instead of building multiple huge rooms; simply have the player go through different parts of the one room. Keep the player in a room for a period of time.
- Make noise! Try to add sounds when the area just doesn't seem complete. FPS Creator comes with a great stock of ambient sounds which are ideal for labs, security stations, alleyways, and more!
- Use segments and entities in a way that nobody has done before. Does your game stand out or look like everyone else's?

Save Polies

Tip by: BIG Viking Games

If you are going to build an open area, like an outdoor scene, try to save polies by using for example billboards instead of "real trees" at places the player will never get access to.

This can be done in quite many places, even in a factory. If there is a room with windows out to the factory floor (the boss's office) which the player will never get near except for seeing what's inside, don't put models there. Use billboards!

Take photos of the model from the angle which the player will see the model, and then make it a billboard. If you make one billboard for each model you won't get that plain feel.

If you make your own models you can delete faces on the model which never will be exposed (like under static box or barrel).

This is how the pros do it.

Screen Resolution

Tip by: Gorba Flamingo

- FPS Creators default resolution is 1024 by 768
- Lower the resolution for better performance, it can make a huge difference.
- Use real resolutions.
- How to change? FPSC Folder, open setup.ini. Change in there the width and the height.

Textures/ Pride

Tip by: Pride

1. A 256x256 requires relatively no engine effort to render.
2. A 512x512 takes 4 TIMES the effort to render, which is just a little bit
3. A 1024x1204 is 16 times more effort than the 256x256, which requires some actual effort
4. A 2048x2048 is 16 times more effort than a 512x512, meaning the engine is going to have to render quite a lot.

My suggestion is use texture size in relation to the importance of an item. Why stress the engine on a table or chair, rather than a soldier in front of you.

Lighting Techniques

Tip by: Dumpus

High texture lightmapping decreases performance, but high quality lightmapping doesn't! For example:

Lightmap texture size: 512

Lightmap quality: 50

This scenario will have long loading times, as well as decrease performance.

Lightmap texture size: 256

Lightmap quality: 100

This scenario will take the same amount of time to load, AND look the same, but without the performance decrease.

Use Few Characters

Tip by: Peter Gee

The use of characters in FPS Creator has a large impact on performance. Alongside graphical rendering, the engine must run the Artificial Intelligence scripts for each and every character.

Sometimes this involves running multiple infinite loops simultaneously, which usually has a large impact on the CPU and game speed.

This problem can be resolved by using fewer characters, or dynamically loading and unloading entities.

Remember there does not have to be huge hordes of enemy running at you every second of your game.

Enemies

Tip by: Nickydude

- Only spawn enemies when they are needed instead of having them all over the map at the start.
- Spawn them in places where the player will have to kill them otherwise they are left to wander and framerate will go down.

Save Framerate

Tip by: Bendak11

Using trigger zones to spawn enemies is an excellent way to increase performance. Also, instead of long open hallways, add a wall or a door so the system doesn't have to render the entire corridor at once.

Delete Dynamic Objects When Player Leaves Vicinity

Tip by: Hockeykid

When the player leaves room and goes through a door use a simple script to close the door and lock it then use a script like...

```
:state=0,always:globalvar=1  
:state=0,plrwithinzone=1,varequal=1:setvar=2
```

...for the **trigger zone**.

Then for the **dynamic object** attach a script like:

```
:state=0,always:globalvar=1  
:state=0,varequal=2:destroy
```

Entity Collision

Tip by: Red Eye

If you make small things , like a cup of tea, (just for eyecandy), then make it box collision and don't make it polygon collision. This will really increase your framerate.

Static Objects

Tip by: JRH

Use static objects wherever possible. To change a dynamic object to a static one, press the "Y" key while it is selected in FPS Creator. Apart from improving your games framerate, the objects will also be lightmapped, hence improving how well your games actually look.

Invisible Walls

Tip by: Puppy of Kosh

When using outdoor scenes, put invisible walls around the perimeter of the room.

Use of Skyboxes

Tip by: Dumpus

If your level is completely indoors, and there are no windows/places showing the outside, rendering a detailed skybox is pointless, and will unnecessarily hurt the framerate. Change it to a less detailed, smaller skybox to increase performance.

Indoor Level Skyboxes

Tip by: Red Eye

Make the texture file 1px-1px (you can do that even with paint) and colour it white or black. This is so the game engine does not need to render the whole skybank texture file, as you're indoors, so you will save framerate. Improves 3-5 FPS (Depends on how big your first Skybox was.).

Physics

Tip by: Red Eye

Imagine you want to create a box dynamically using a trigger zone, but you want to save framerate. By changing the properties of the entity (ex: box), setting the physics to off, you can save an extra few FPS.

Model Pack 3 (Boxes) Tip

Tip by: CoffeeGrunt

If you own Model Pack 3, you can save a good few polies by using bigger boxes in place of smallones, here's how:

- 1) Run FPSC, and add an entity from Model Pack 3 (Boxes_Wood), and select one of the crate_big entities.
- 2) Place it in the level, and make it static.
- 3) Right click on it, and change its texture by clicking on the small tab with 3 dots that appears when you put your mouse over the texture filepath. Navigate to entitybank/M3/Boxes_Wood, and select one of the Crate_Small_D textures near the bottom of the folder, double click to apply it to the entity.
- 4) There you are, you have just used one large crate, to fill the space of eight smaller ones! You can add a couple of the small crate entities around the large box to make it look less neat. This technique works best in well-ordered warehouse environments.

Lightintensity/Strength Command

Tip by: Apple Slicer

I have seen a lot of games that use dynamic lights that look "dull".

For an example, you may have noticed that if you place a dynamic light and a static light down, and test the game, you will notice that the dynamic light seems less bright compared to the static one.

To "fix" this, simply add something like this to the light's script:

"lightintensity=500" (This is an "action")

Here is an example:

Light1.fpi

```
;Artificial Intelligence Script  
;Header  
desc = Light Toggle (On By Default)  
;Triggers  
:state=0:lightintensity=500,state=1  
:state=1,activated=0:state=2,lighton  
:state=2,activated=1:state=1,lightoff  
;End of Script
```

Extra Notes: (by Flatlander)

I also tried 1000, however I couldn't tell much difference between 500 and 1000.

GamePlay Tips/Hints

Tip by: Pozag

Once you have your design layout on paper, colour the sectors that are crucial to the progress and the ones that are optional. When you build your level, only create the crucial sectors and put a place holder or just a fake door that would lead to the optional rooms/sectors. Test your level with your crucial sectors and monitor the performance/fun factor/Duration and etc.

Add one room at a time and see how it affects the overall user experience. Exploration greatly improves the immersion but over doing it doesn't really help either. It's important to balance this factor early in the creation then later as balancing the factor early will offer the possibility to iron out leaks and bugs quicker. When you're done with your complete design, you may or may not remove the place holders of "originally planned rooms". If you do leave the place holder, add a sound effect of locked doors. For some users, this will trigger a sense of mystery and might even enhance your puzzles.

IE: You found a key but you're not sure which door it will unlock.

Sun in FPSC

Tip by: beachard5

You can add a sun in FPSC by just adding a Light (yellowish colour) two-three layers above your level. But if you have a skybox with a sun you can test your game, see where the sun is. And then place the light in the direction of the sun. Now you have to change the range of the light. So Right- Click on the light, change the colour. And set range to 10 000. That would do it.

There aren't Pure Colors in the World

Tip by: bkinsman and beachard5

There is nothing like: pure yellow or pure white or pure blue. No it is always in between.

You should have a play around with the colour selection i.e. make up a colour by moving the pointer around a bit so that it is more of a light peachy/creamy colour, but it should be light enough so it doesn't make it look like a strong peach colour. I have had a play with it and I think that "#e8f5ff" should be about the right light colouring, but hey people have a play around there are thousands/millions of colours to choose from.

The Use of Modifications

Tip by: JRH

There are several "mods" for FPS Creator, many of which are free. The use of mods can enhance gameplay and graphics, giving developers the ability to make unique games. Listed are the different modifications and a short review of their features.

Airmod:

For the typical developer Airmod allows the use of more complex scripts with a greater list of Actions and Commands than a default FPS Creator installation, as well as adding a number of features, removing bugs and other hard-coded limitations. Airmod is freely available, and is open source.

Ply's Mod:

Allows much more control over games through scripts. Includes the ability to use an unlimited number of variables.

S4Mod:

Faster and more stable game performance with several added features, alongside multiplayer enhancement.

Project Blue:

Integrated Ply's Mod V1.08, Airmod V0.6, Lemur V2 Multiplayer Features and S4Mod. The ultimate mod, merging all others into the one.

The more recent release of the FPS Creator V1.15 source code has enabled the development of new more compatible tools and mods, which may be expected to be released soon.

Texture Quality-Add-on to Prides

Tip by: Apple Slicer

In most cases, you would want to have a 512\512 texture size. However, this can kill performance in large rooms. But, using a 256\256 texture can look bad. Well, why not go in between? Use a 384\384 texture size for your walls and floors. There is truly not much of a notable difference in visuals, but fps speeds up dramatically.

Second, I have seen many games that have a room up to three or four segments high. Create a new "segment" and name this one "*your original segment name here*"-High ceiling. Now, you can cut the texture size down to even 256, and using clever lightmapping, keep it dark enough that the player will never know.

Keep player moving forward-add-on to Hockeykids

Tip by: Apple Slicer

If you use Hockeykids script, the player might be able to get back, and see that the stuff just disappeared. Not cool. To solve this, add a door, and apply this script (below) to it. Then, name the door, *door_shut_one*. Now, add a trigger zone just outside the door. Change its main fpi to *plrzoneactivateifused.fpi*.

Now, change its "*if used*" field to *door_shut_one*.

Now, what should happen is when your player leaves the room, and walks out the door, the door will shut behind him, in this same spot, you can add another trigger zone, and apply hockeykids script.

```
;Artificial Intelligence Script
;Header
desc = Remote Controlled Door (Open initially and Closed
by trigger)
;Triggers
:state=0:state=1,setframe=0,sound=$0
:state=1:incframe=0
:state=1,frameatend=0:state=2,coloff
:state=2,activated=1:state=3,sound=$1,colon
:state=3:decframe=0
:state=3,frameatstart=0:setframe=0javascript:insertSmiley
('[B]')
;End of Script
```

---oOo---

AN INTERVIEW WITH JOHN ZUUR PLATTEN, A TOP-NOTCH GAME WRITER.

By le Shorte

Q: How do you beat writer's block?

A: Okay, well, a couple of suggestions. I've found that most writer's block happens because you become overwhelmed by the amount of work that needs to be done. For instance, on a screenplay, you think "I can't write 120 pages", so that keeps you from writing one page. I suggest that the first step is to break the project down. In games, start by breaking down the levels (or missions), then tackling it as an individual module. You can break this down further by set-piece. Write a few and then start piecing it all back together.

Second, do writing exercises. So, for instance, you can start by writing about your day, or describing the space you're writing in, or an event that happened in your life. Something that you know and can easily describe. The creatively comes in how you communicate the details. Then, segue into creating the details and you are back into the world of fiction. And now you are rolling.

Third, writing is a discipline like many other creative endeavours, so the main issue is to do it. Write a little every day if you can, and it becomes part of the routine.

Last, don't be overly critical. I think what freezes a lot of writers is that they worry that what they are creating isn't any good, so they create nothing. Most writing in the creative space is really re-writing. Scripts, comic books, games, etc all go through many revisions, so don't invest everything you have in the 1st Draft. Hope that helps.

Q: What are some good, FPSC worthy ways, to create a climax to your story?

A: First point. All stories rely on conflict. Without conflict you don't have a narrative. There are all sorts of conflict, from violent to humorous, but all stories are based around the idea of a protagonist, and an antagonist (a hero and a villain). This can take many forms, obviously. What is great about games is that the conflict, for the most part, is the core element of gameplay. So what I try and do is find ways to support that with compelling characters and set-pieces that "pay-off" that gameplay. So, when we get to the climax of the story, it has to have been set-up along the way during the course of the game. A good climax is never a huge surprise (even if it is shocking). Instead, it is the logical conclusion of all that came before it. That is the biggest mistake that new writers make - they don't set-up the pay-offs to come. I'll give a couple of examples. Quick ones. In film, let's take a movie like *Back to the Future*. That entire movie is a series of set-ups in the first two acts that pay-off one after another in the third. Each beat of the story leads to a logical sequence that rewards the viewer. When Marty has to play his guitar at the dance, we know that he can do it because we've seen him in the first scene of the movie wanting to be a rock god. Now, at the end of the movie, the way he saves himself is by becoming a god (of rock and of his destiny). This is why the audience eats it up... because they see the character being rewarded (and by extension, they are as well). Now, from games.

I'll use an example from *Chronicles of Riddick*, since I wrote it. 😊

The pitch was that at the start of the game, Riddick is thrown in the slam, and at the end of the game, he escapes. Pretty simple. But how he escapes needed to be something more than simply killing the warden. He needed to get even with him in a unique way. Well, we had set up (for gameplay) that almost all the guards were trigger happy maniacs that would shoot first and ask questions second. So, to create a compelling climax, rather than Riddick killing the warden, instead, he used him as a distraction to escape. But he also set his

distinctive goggles over the warden's eyes and had him tied up, so that when the guards entered the darkened room, the first thing that their lights would reflect off of would be the goggles, and they would assume that the warden was Riddick. Of course, the result was they opened fire (as we had established they would), and in them killing the warden, we get an ironic and logical ending that shows how smart Riddick is at engineering his escape.

Q: How should writers warm up before they sit down to start typing or writing away at a story?

A: I like listening to music. I have a large collection of soundtracks and also of ambient music, so I'll sometimes start with those. Also, I need to clear out any distractions, so I answer emails first and then put the phone on vibrate. You have to live inside your own head while you are writing, so you need to have that space to yourself. I also never assume that when I sit down, it is going to happen right away. So, if I'm working on a script, I'll usually start by reading what I've recently written, and then do any corrections that I see need to be done. This sort of gets me walking before I start running on all new material.

Q: Help! I'm a game designer who just found out a feature isn't capable in the engine's limitations! What do I do?!

A: Yeah, that is a tough one. A big problem in all creative endeavours is when expectations exceed reality. Often, you have a great idea, but the limitations of time, budget and technology mean that you can't deliver it. I also hate the "perfect idea" that happens three months too late. I can't tell you the number of times that someone has come up with an amazing idea on a project I'm working on when we are just about to go into alpha. You hate it because you know it could make the game so much better, but it is not going to happen. Big disappointment. So, my suggestion first and foremost is you have to design within the box that you have before you. I've seen a lot of amazing games on the platforms like the iPad that have been hugely successful by focusing on what makes the game fun and entertaining, not on how many pixels can be pushed around the screen at what frame rate. The trick with independent game development is that since you can't compete with the budgets of EA and Activision, don't try. Do what they aren't doing. Make it quirky. Make it cool in ways their marketing departments won't allow. That is not anything that is limited by technology. That is making smart creative and design choices given what you can deliver.

Q: Are there any screenwriting programs you recommend for a small budget? We're talking about, say, \$50 of a budget.

A: That is a good question. I use Final Draft, which is the industry standard. It runs about 200 dollars, but can create scripts, comic books, manuscripts, etc. Very intuitive and easy to use. There are some others that are free or nearly free. I believe there is a program called Celtx that is based on open source. And Movie Outline (sorry, not sure of the price). There is

also Movie Magic, which is about the same price as Final Draft. I do know that there are templates for Word which are available for free that have screenplay formatting. Most of the scripts I write for games are either in standard screenplay format, or in [Microsoft] Excel.

Q: If you could give one message to every aspiring game designer out there, what would you say?

A: Do what you believe in. This is the same message I would give to any creative. So many folks try and create something that they believe will sell, or what they see is "hot" right now, and so rather than creating something interesting, they instead create a lesser, "me too" version of what is already being done (better) by others. Do what makes you happy, and what you believe in.

Q: Do you need to come out with the next Call of Duty or Half-Life to be a successful game developer?

A: No. In fact, I don't think you can. The market to create the monster franchise is falling apart in my opinion. So, instead, the market is going to be looking for interesting new expressions of tried and true gameplay. In the FPS space, I think that means moving away from space marines, commandoes, WW2 and special forces types, and looking at new content that doesn't have me facing waves of zombies or aliens or Nazis or terrorists. Once again, I think that a game like Portal is a very good example of a smart way to approach independent development.

Q: What is the best way to get noticed as a game developer?

A: Very good question. I would say that you need to be unique, and have a very distinctive "voice". So, what is the overall "vibe" of both your game and the people behind it. Rock Star is a great company because they make very good games, but you also have the sense that the people that run the company see themselves as Rock Stars. They are projecting an attitude that is part of their "brand". So, if you are doing a Steampunk game for instance, for instance, what is the name of the company that makes me feel this vibe? What is the overall tone of the piece? How are the players within that company projecting that image? And so on. It is the idea of branding not just the game, but the developer, in a way that projects an image that is appealing.

Q: If all someone can come up with is a story made clichés, what is, in your opinion, the best way to get out of the cliché state of mind?

A: So, let's start with a cliché we all know, and then work out way out from it.

Cliché - Lone hero with a gun and dark past that includes some personal tragedy.

Now he wants revenge.

Okay, if we look at what Mark Millar did in Kick-Ass, he trains his 10 year old daughter to be a murderous assassin. So now he has projected that need for revenge, and passed it on to his daughter. And of course, because it is such an outrageous idea, it naturally lends itself to black comedy. And now that lone hero becomes Big Daddy, which is the obvious name he would choose.

So something new and original springs from a guy acting like a second-rate Batman.

Usually, in my opinion, it doesn't have to be much to a cliché and tweak in some new direction that can be a lot of fun.

Q: What's an effective way to convey story through level design without overpowering FPSC?

A: Well, I think the best way to tell story in games is:

1 - Play it.

2 - Show it.

3 - Say it.

4 - Cut Scene it.

So, if the narrative beat can be playable - your ally turns on you in the game by actually shooting at you - then that is the best way.

Then, if you can't do that, show it by seeing him retreating toward an enemy and not getting shot, then joining up with them and now you know he's gone to the other side.

Then, have some VO that explains what is happening if need be.

Finally, have the Cut Scene if you feel it is necessary to advance the story and build the relationship between the characters.

I would say that the cut-scene is your final choice. Not you're first. To the extent the game can tell your story, the better.

Bioshock is very good at giving you a tremendous amount of story information that comes across through gameplay. It is in the world, in the enemies, in the sounds, etc. The world you are exploring and fighting in is the story.

Side Question: So having the player actually experience the event is more compelling than a cinematic cutscene?

A: Yes. If you can play trying to run your character across the collapsing bridge, rather than simply showing it, I believe that is better. That is what games can bring to the party that other media cannot.

Questions & Answers

Q: HOW DO I CHANGE THE DEATH ANIMATIONS OF A CHARACTER?

A: These are the stock death animations:

- anim11 = 0,19
- anim14 = 40,59
- anim17 = 120,139
- anim20 = 80,99
- anim61 = 0,19
- anim64 = 40,59
- anim67 = 120,139
- anim70 = 80,99

This is how they should look for better ragdoll:

- anim11 = 0,1
- anim14 = 40,41
- anim17 = 120,121
- anim20 = 80,81
- anim61 = 0,1
- anim64 = 40,41
- anim67 = 120,121
- anim70 = 80,81

Open up the characters .fpe file, go down and you'll see "*animation info*", along with all the animations. Then just change the animations listed above.

For example you would change

(anim11 = 0,19) to (anim11 = 0,1)
(anim64 = 40,59) to (anim64 = 40,41)

---oOo---

Outro

Well, we've come to the end of Part 2 of this guide and I'd just like to thank all those who've provided something for the guide, without you guys FPSC would be sitting in obscurity, relegated to the dusty shelf of time. I'd also like to thank the FPSC community for their help and encouragement to newcomers, showing them just how fantastic **FPS Creator** really is!