Week4 Lecture: iPhone Games

Contents

Design Discussion.

We will add the aliens now and handle their wrapping behaviour before anything. Don't worry about their AI right now, we will handle that in due time. For now we will place them and write a load mission function, then set them up to be hit, add a particle effect for their demise, as well as add firing for the player and some scoring.

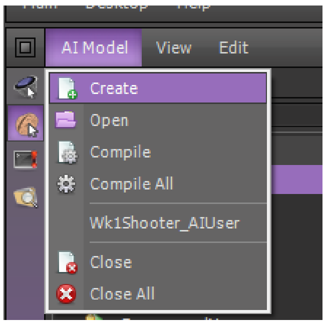
Creating variable for Aliens.

We will need a factory method to handle aliens. A factory class is first created, and then a single alien class. Then when told to, the factory spawns aliens and manages them. Because an alien cannot destroy itself, it sends a message to the factory to have it removed.

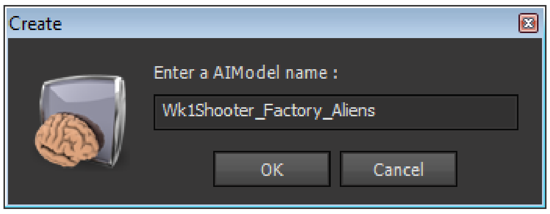
Let's start with the factory.

Alien Factory

Open the Code workspace and select AIModel->Create.



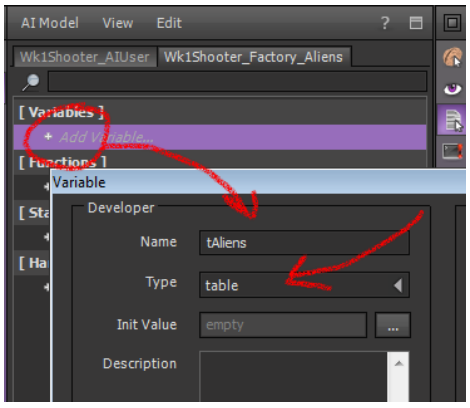
Name it Wk1Shooter_Factory_Aliens.



Enter a AIModel name :

Wk1Shooter_Factory_Aliens

Go to the top of Wk1Shooter_Factory_Aliens and add the following Variable...



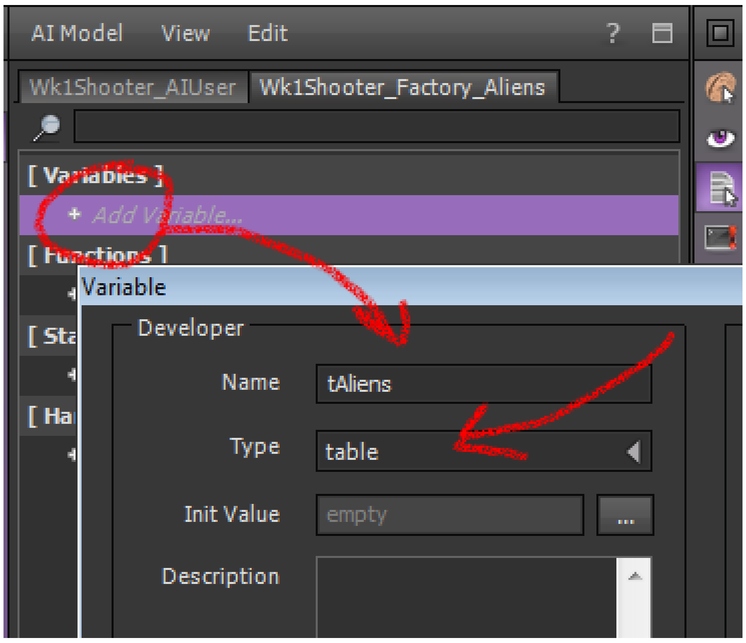
Name tAliens

Type table

Init Value empty

Description

It's a TABLE. Tables are lists of object of any type. We will just continually add objects of type Alien to tAliens...



Name: tAliens
Type: table
Init Value: empty

Wk1Shooter_Factory_Aliens will need a number of generic factory methods. We generalise the names so we can later copy the entire AI Model for handling other elements such as stars, explosions, lasers etc.

The functions...

destroyAllObjects

destroyObject

instantiateObject

The events...

onCreateObject

onInit

onEnterFrame

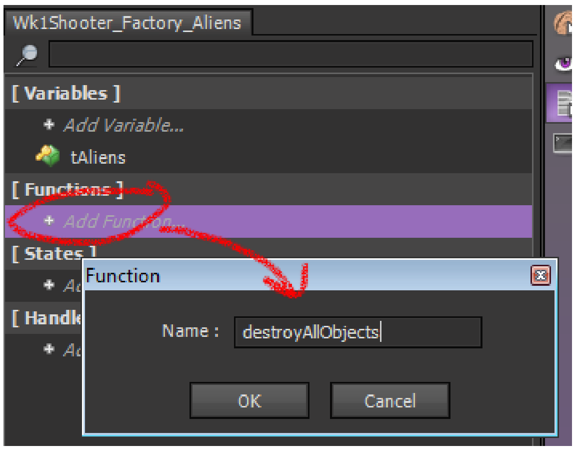
onDestroyObjects

onDestroyObject

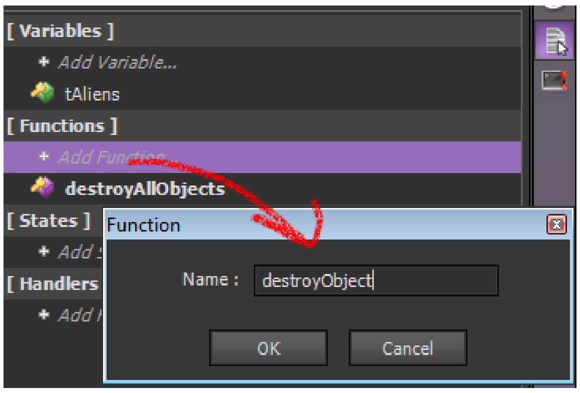
The variables could be generalised as well but it's easy enough to copy the AI Model and rename the specific variable by editing the variable name ShiVa will rename all the uses of it in the related code.

So go ahead and add them. I'll show you how to add each of them.

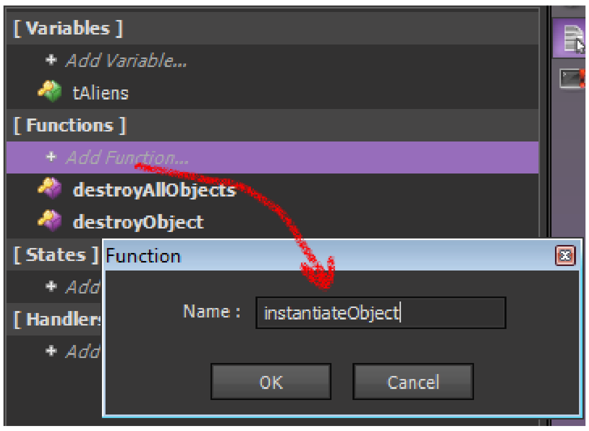
destroyAllObjects



destroyObject

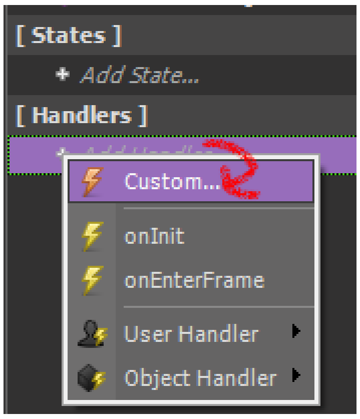


instantiateObject

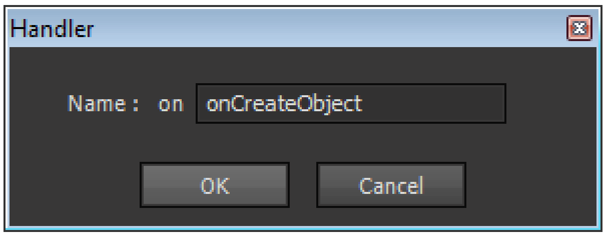


onCreateObject

This is a Custom Event so click on ...



Then name it...

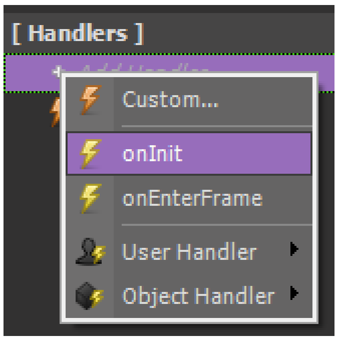

Handler

Name : on onCreateObject

OK Cancel

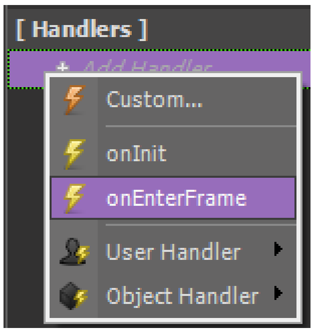
onInit

This is a built-in handler/event so you can just click on ...

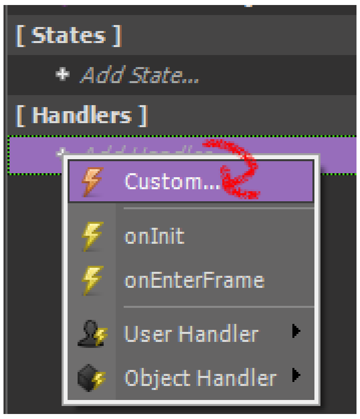


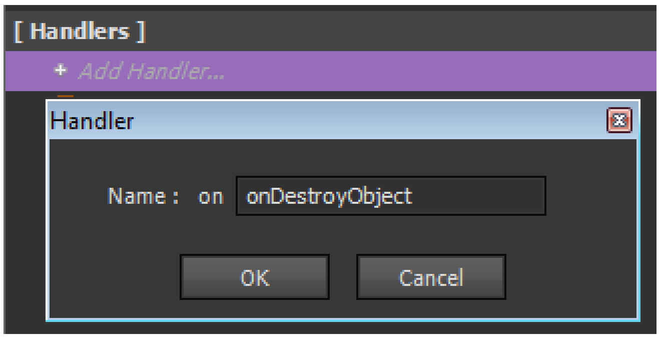
onEnterFrame

Another built-in handler/event...



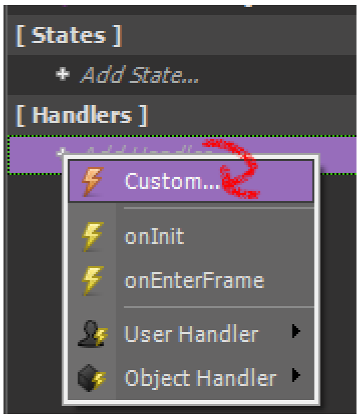
Add the onDestroyObject handler/event, it's a Custom handler...

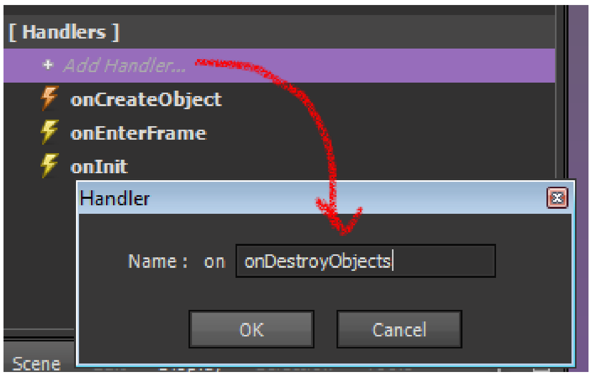




And finally, another custom handler/event...

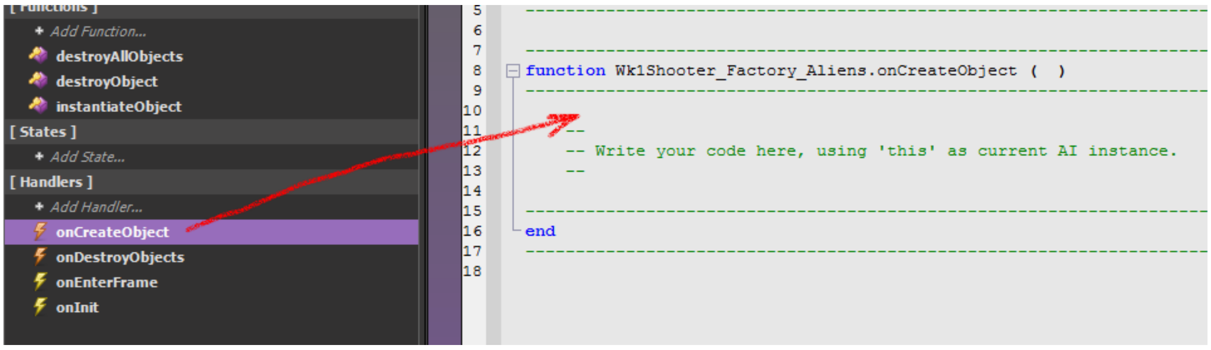
onDestroyObjects





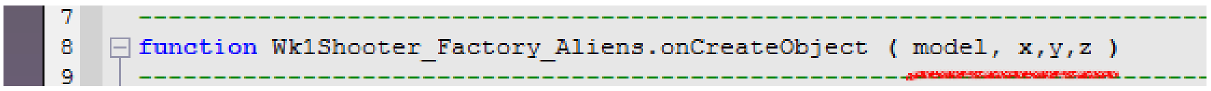
Okay all ready to code in.

Let’s start with onInstantiateObject. Double left click on it to open the script...



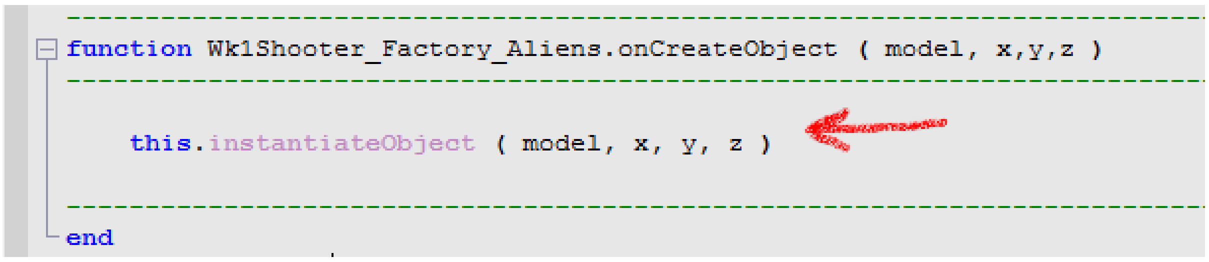
```
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
function Wk1Shooter_Factory_Aliens.onCreateObject (  )
--------------------------------------------------------------------------------

    --
    -- Write your code here, using 'this' as current AI instance.
    --

--------------------------------------------------------------------------------
end
--------------------------------------------------------------------------------
```

We’ll need to modify the parameter list so change this first...



```
--------------------------------------------------------------------------------
function Wk1Shooter_Factory_Aliens.onCreateObject ( model, x,y,z )
--------------------------------------------------------------------------------
```

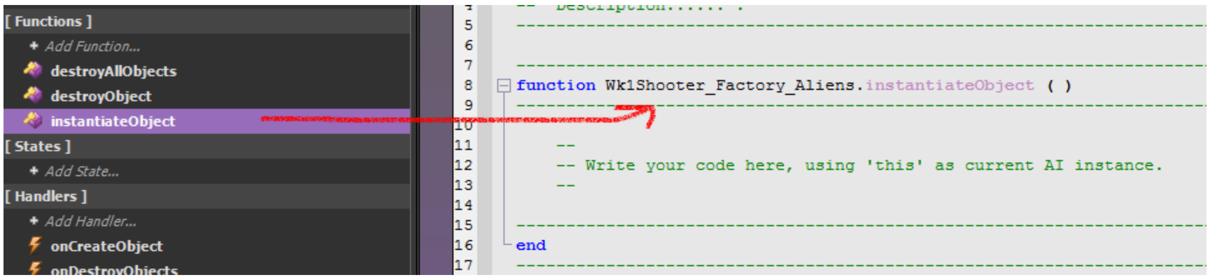
Now we call its own function...



```
--------------------------------------------------------------------------------
function Wk1Shooter_Factory_Aliens.onCreateObject ( model, x,y,z )
--------------------------------------------------------------------------------

    this.instantiateObject ( model, x, y, z )

--------------------------------------------------------------------------------
end
```

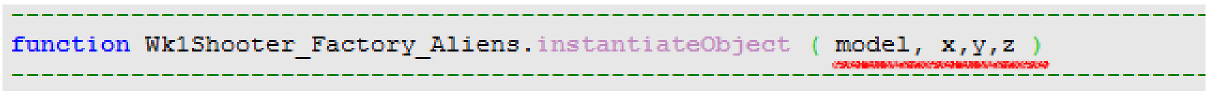
Let’s continue on and code the instantiateObject function...

Double left click on it in the function list to open the script.



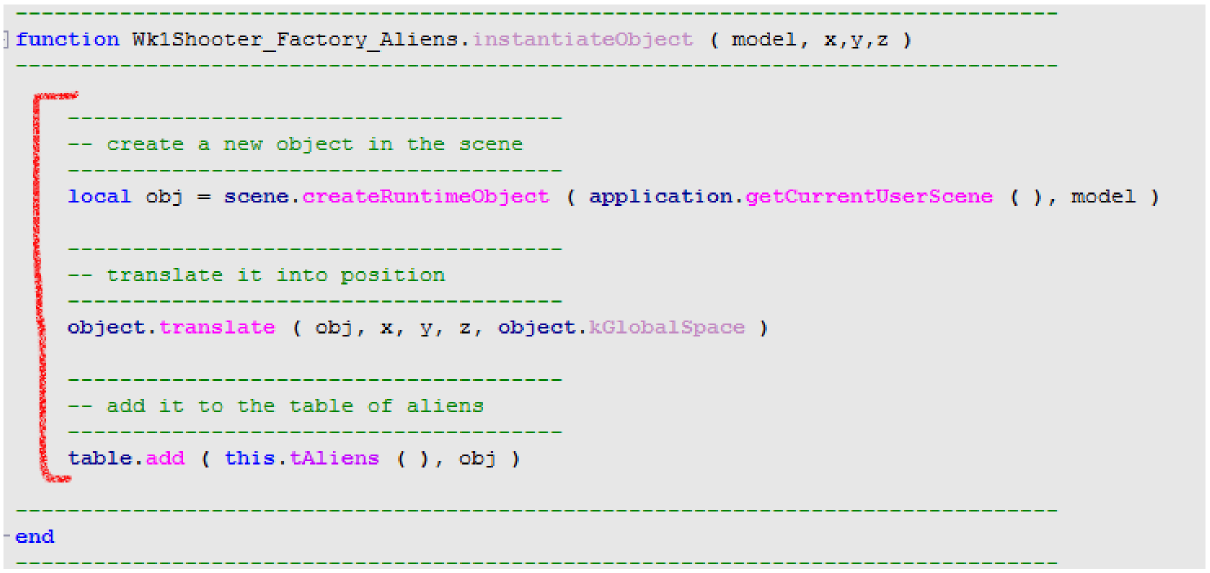
```
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
function Wk1Shooter_Factory_Aliens.instantiateObject ( )
--------------------------------------------------------------------------------

    --
    -- Write your code here, using 'this' as current AI instance.
    --

--------------------------------------------------------------------------------
end
--------------------------------------------------------------------------------
```

Now change its parameter list...



```
--------------------------------------------------------------------------------
function Wk1Shooter_Factory_Aliens.instantiateObject ( model, x,y,z )
--------------------------------------------------------------------------------
```

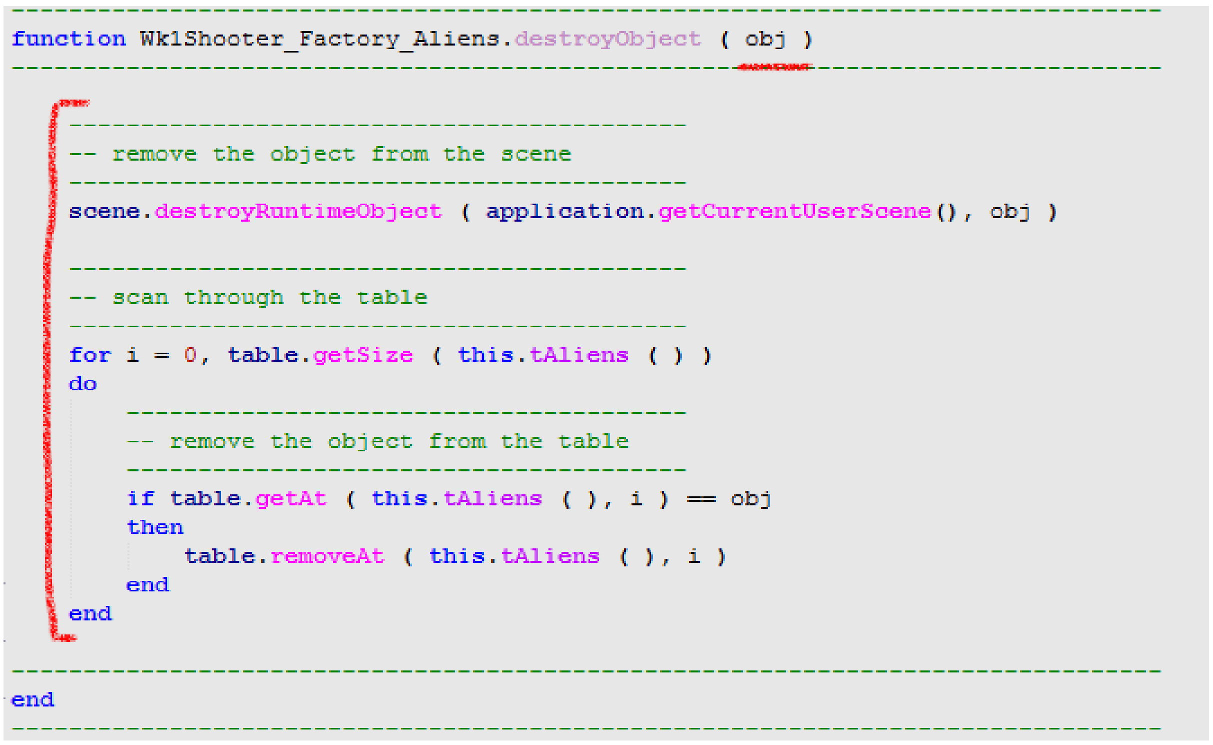
And then fill in the code... this is where we add it the scene and add it to the list of aliens we can manage using this factory object.



```
--------------------------------------------------------------------------------
function Wk1Shooter_Factory_Aliens.instantiateObject ( model, x,y,z )
--------------------------------------------------------------------------------

    --------------------------------------
    -- create a new object in the scene
    --------------------------------------
    local obj = scene.createRuntimeObject ( application.getCurrentUserScene ( ), model )

    --------------------------------------
    -- translate it into position
    --------------------------------------
    object.translate ( obj, x, y, z, object.kGlobalSpace )

    --------------------------------------
    -- add it to the table of aliens
    --------------------------------------
    table.add ( this.tAliens ( ), obj )

--------------------------------------------------------------------------------
end
--------------------------------------------------------------------------------
```

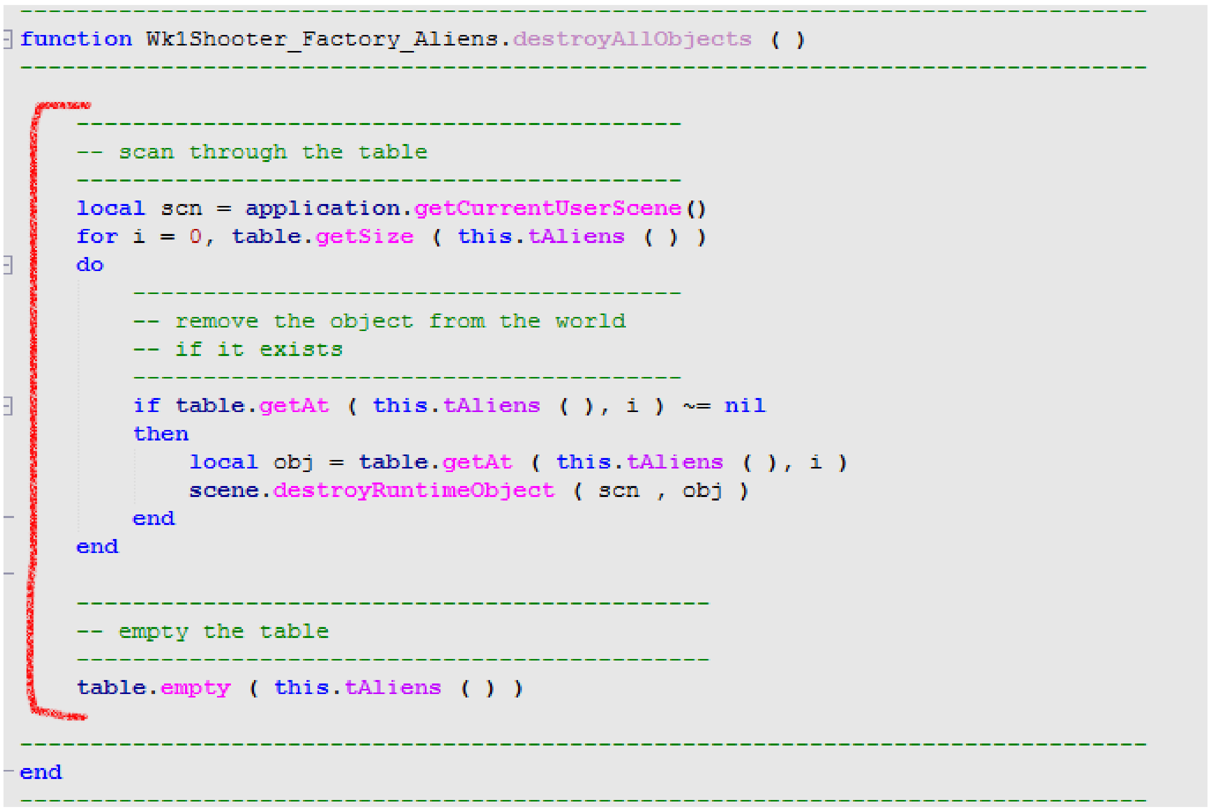
We store a local variable of the object when it’s created in our scene, using the passed in model name. Then we move it in global space to the passed in coordinates. Finally we add the object to the table of objects.

Double left click to open the destroyObject list and enter both the parameter and the code...



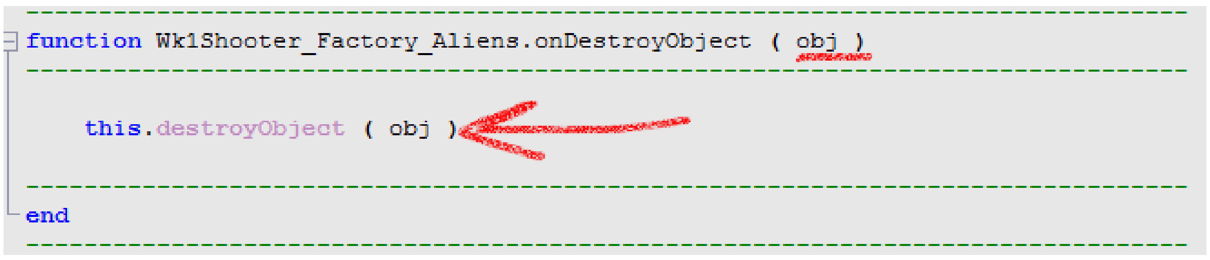
```
--------------------------------------------------------------------------------
function Wk1Shooter_Factory_Aliens.destroyObject ( obj )
--------------------------------------------------------------------------------

    -------------------------------------------
    -- remove the object from the scene
    -------------------------------------------
    scene.destroyRuntimeObject ( application.getCurrentUserScene(), obj )

    -------------------------------------------
    -- scan through the table
    -------------------------------------------
    for i = 0, table.getSize ( this.tAliens ( ) )
    do
        ---------------------------------------
        -- remove the object from the table
        ---------------------------------------
        if table.getAt ( this.tAliens ( ), i ) == obj
        then
            table.removeAt ( this.tAliens ( ), i )
        end
    end

--------------------------------------------------------------------------------
end
--------------------------------------------------------------------------------
```

Double click in the function list to select destroyAllObjects and enter the following code...



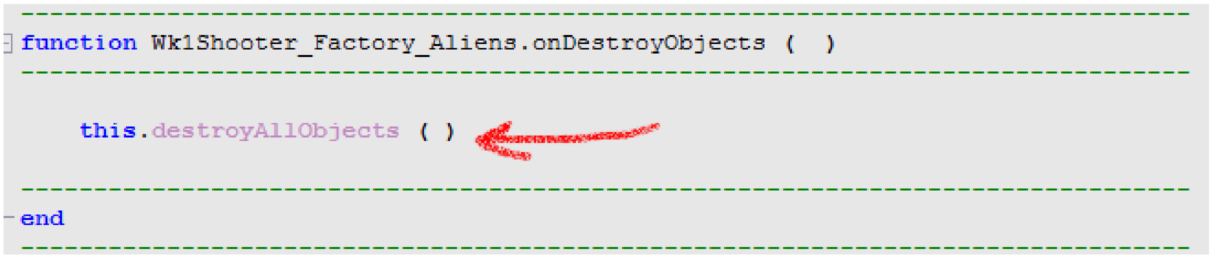
```
--------------------------------------------------------------------------------
function Wk1Shooter_Factory_Aliens.destroyAllObjects ( )
--------------------------------------------------------------------------------

    -------------------------------------------
    -- scan through the table
    -------------------------------------------
    local scn = application.getCurrentUserScene()
    for i = 0, table.getSize ( this.tAliens ( ) )
    do
        ---------------------------------------
        -- remove the object from the world
        -- if it exists
        ---------------------------------------
        if table.getAt ( this.tAliens ( ), i ) ~= nil
        then
            local obj = table.getAt ( this.tAliens ( ), i )
            scene.destroyRuntimeObject ( scn , obj )
        end
    end

    ---------------------------------------------
    -- empty the table
    ---------------------------------------------
    table.empty ( this.tAliens ( ) )

--------------------------------------------------------------------------------
end
--------------------------------------------------------------------------------
```

There’s a bit more to finish this off. Double click on the onDestroyObject handler and add this code...



```
--------------------------------------------------------------------------------
function Wk1Shooter_Factory_Aliens.onDestroyObject ( obj )
--------------------------------------------------------------------------------

    this.destroyObject ( obj )

--------------------------------------------------------------------------------
end
--------------------------------------------------------------------------------
```

Finally double left click on the handler onDestroyAllObject and enter the following code...



```
--------------------------------------------------------------------------------
function Wk1Shooter_Factory_Aliens.onDestroyObjects (  )
--------------------------------------------------------------------------------

    this.destroyAllObjects ( )

--------------------------------------------------------------------------------
end
--------------------------------------------------------------------------------
```

Cool. Now we have a generic factory object, which is ad hoc enough to handle our alien, but could easily be copied and have some variable renamed to use it for any object that needs multiple copies.

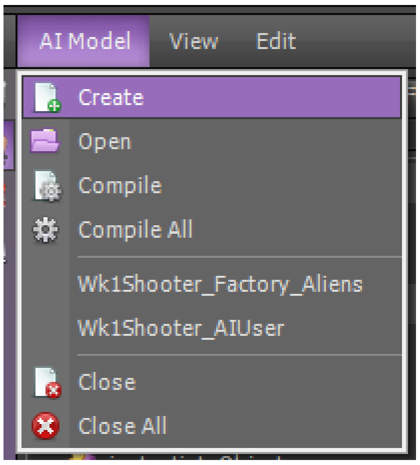
Alien Class

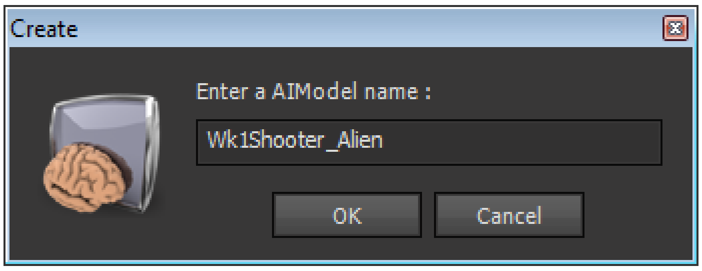
Now that we have the factory object to spawn aliens and manage them it’s time to create an alien class.

We need to think about what we want our aliens to do. Some things come to mind.

- Shoot
- Move
- Hunt
- Avoid
- Die

So let’s make an AI Model called Alien...

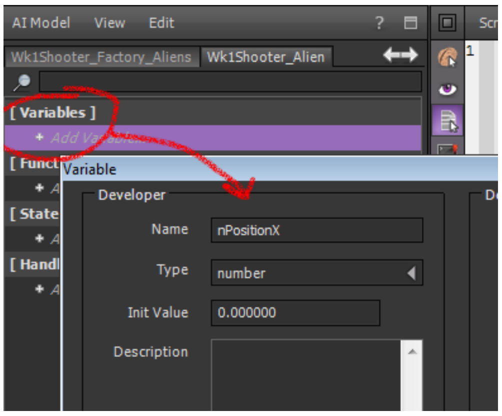




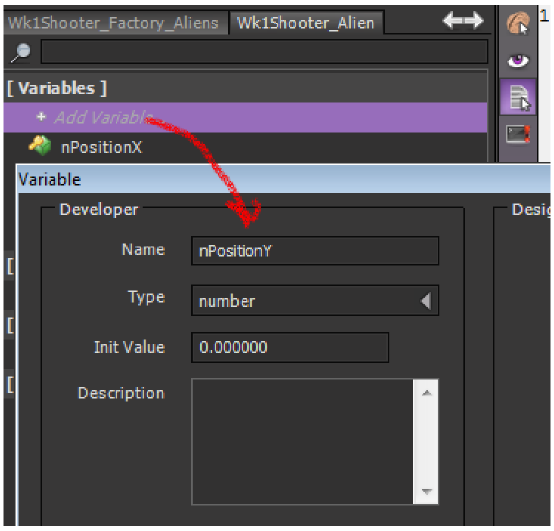
Let’s add all the functionality listed above. We’ll need some variables first...

- Position
- Is Firing?
- Damage
- State – more on this in a minute
- Score
- Speed

Add nPositionX and nPositionY ...

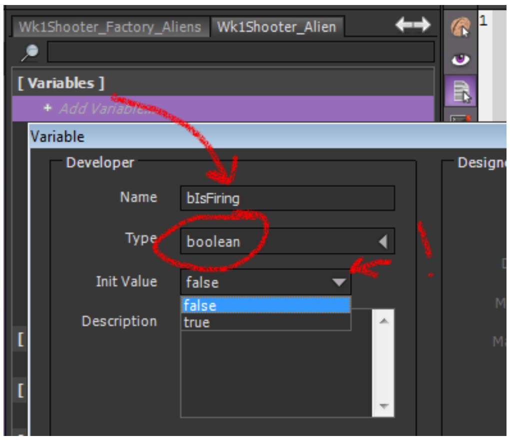


Name nPositionX
Type number
Init Value 0.000000
Description



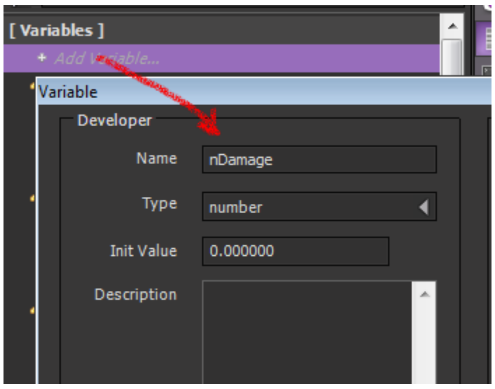
Name nPositionY
Type number
Init Value 0.000000
Description

Next we'll add bIsFiring... (NOTE! Change it's default to False...



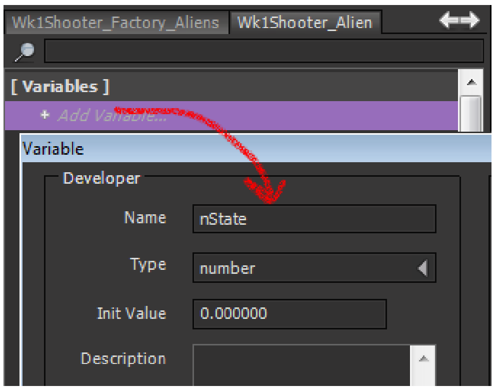
Name bIsFiring
Type boolean
Init Value false
Description

nDamage is another number...



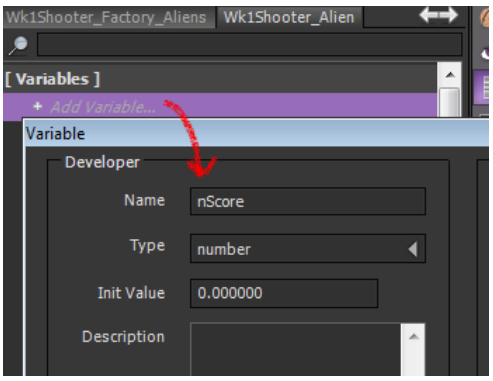
Name nDamage
Type number
Init Value 0.000000
Description

State is a number...



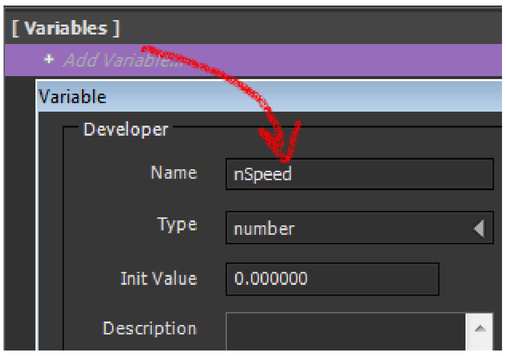
Name nState
Type number
Init Value 0.000000
Description

Score is a number...



Name nScore
Type number
Init Value 0.000000
Description

Speed is a number also...



Name nSpeed
Type number
Init Value 0.000000
Description

State

There is a section in the AI Model called State... which allows us to change the functionality of the object based on state changes. Our Alien as discussed has several modes of operation (or states) and we can get the AI Model to switch between them as required.

What states do we want? I can think of a few.

- Hunting
- Avoiding
- Attacking

The aliens when hunting are looking for humans and so their game loop will call these functions.

- handleHunting
- handleMoving
- handleCapture
- handleDamage

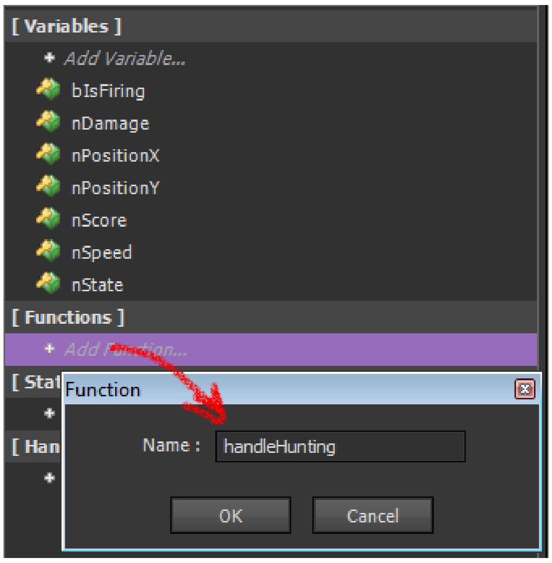
The aliens when avoiding will perform these function. Notice what they don't do.

- handleAvoid
- handleMoving
- handleDamage

When the aliens are in attack mode they perform different functions again.

- handleTrackPlayer
- handleFiring
- handleMoving
- handleDamage

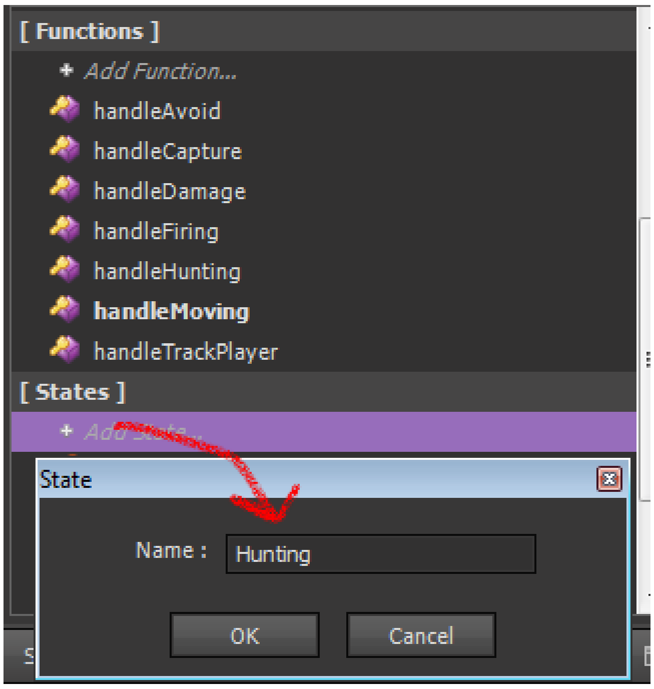
Let's make all the functions. I'll start you with one and you can do the rest. (take note of the variables you should have in this pic)


[Variables]
+ Add Variable...
bIsFiring
nDamage
nPositionX
nPositionY
nScore
nSpeed
nState
[Functions]
+ Add Function...
[Stat
+
[Han
+
Function
Name : handleHunting
OK
Cancel

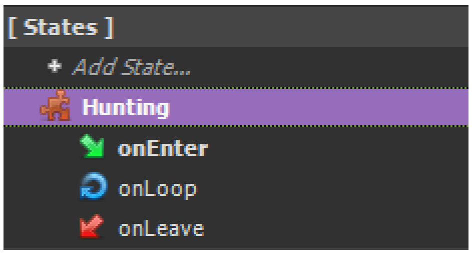
Add all these functions...

- handleMoving
- handleCapture
- handleDamage
- handleAvoid
- handleTrackPlayer
- handleFiring

Now we add the states. Here's how you add the first one. Notice also in the screen grab the list of functions you should by now.

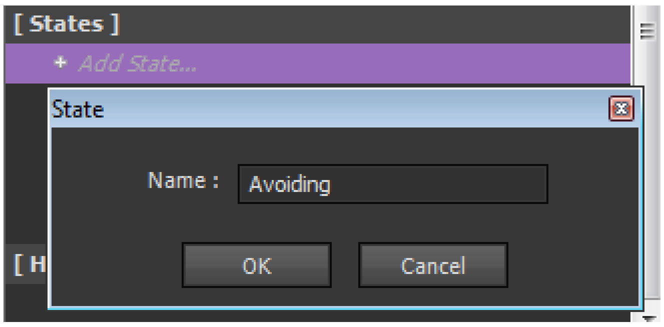


Let's have a look at this first state. It has three actions, onEnter, onLoop and onLeave...

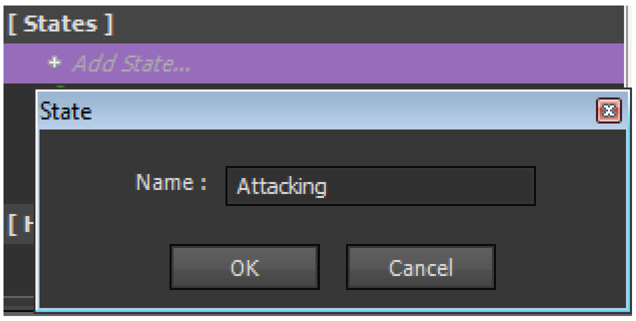


- **onEnter** happens the first time and only once. You should initialise behaviour in here.
- **onLoop** happens every game cycle. This is where you would call the functions required based on the states (ie handleHunting, handingMoving etc...)
- **onLeave** happens when you change state but before the next state changes (which would start with onEnter)

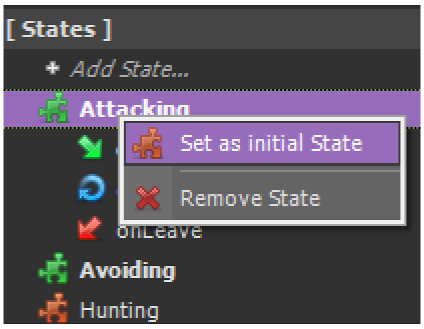
Let's add the other two states... Avoiding



And Attacking...



Finally let’s set the initial state by right clicking on the Attacking state and selecting Set Initial State – the icon will turn orange...



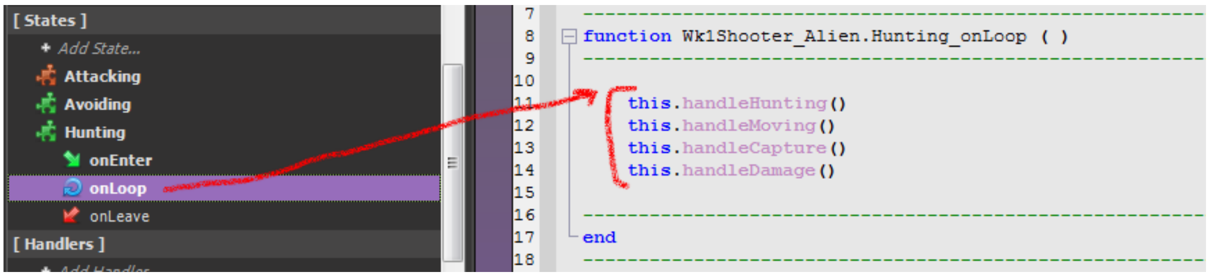
Behaviour in states.

I'll reiterate the functionality of the Hunting state...

The aliens when hunting are looking for humans and so their game loop will call these functions.

- handleHunting
- handleMoving
- handleCapture
- handleDamage

So open Hunting's onLoop function, and add a call to all those functions...

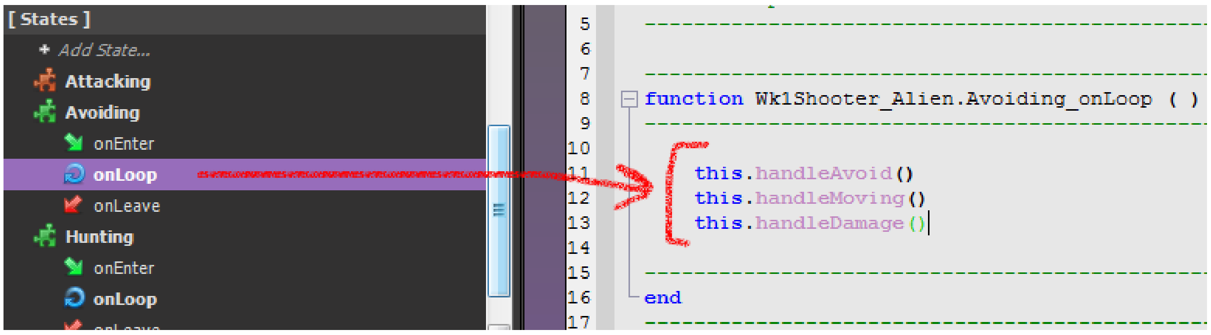


```
------------------------------------------------------------
function Wk1Shooter_Alien.Hunting_onLoop ( )
------------------------------------------------------------

    this.handleHunting()
    this.handleMoving()
    this.handleCapture()
    this.handleDamage()

------------------------------------------------------------
end
------------------------------------------------------------
```

Next we'll define the Avoiding state...

The aliens when avoiding will perform these function.

- handleAvoid
- handleMoving
- handleDamage

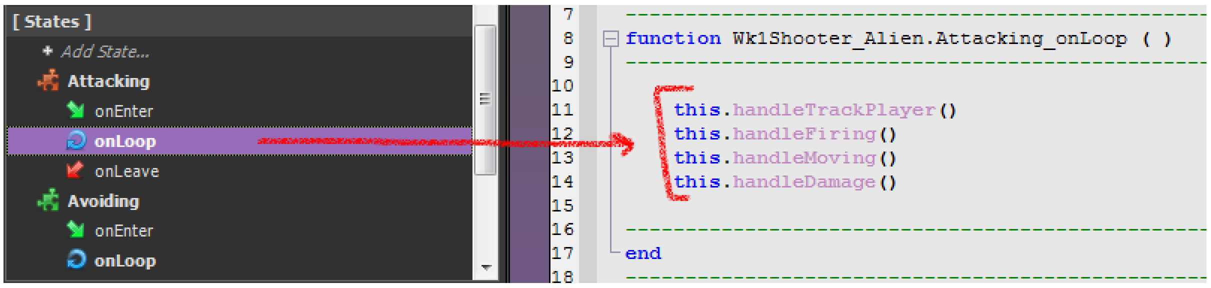


```
------------------------------------------------------------

------------------------------------------------------------
function Wk1Shooter_Alien.Avoiding_onLoop ( )
------------------------------------------------------------

    this.handleAvoid()
    this.handleMoving()
    this.handleDamage()

------------------------------------------------------------
end
------------------------------------------------------------
```

Finally open the onLoop function for Attacking and fill in these function calls...

When the aliens are in attack mode they perform different functions again.

- handleTrackPlayer
- handleFiring
- handleMoving
- handleDamage



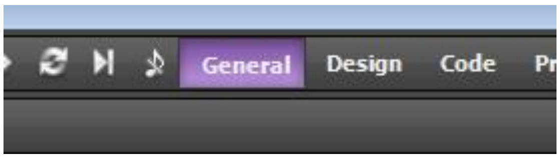
```
function Wk1Shooter_Alien.Attacking_onLoop ( )

    this.handleTrackPlayer()
    this.handleFiring()
    this.handleMoving()
    this.handleDamage()

end
```

Okay all the hard bits are done... now for the fun bit. Not really, it's all hard work from here ☺

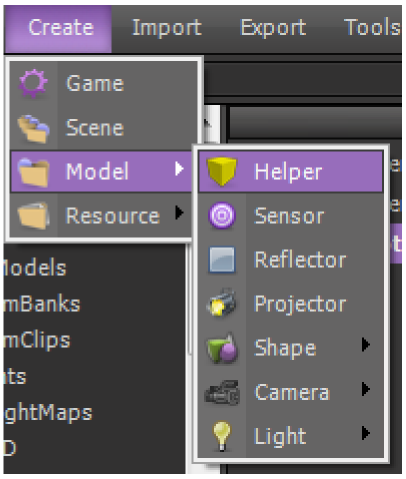
Helper Nodes.

In order for the factory to work it has to be in the scene attached to some object. We could attach it to the player because we know they will persist throughout the game, but better to use helper object and attach the factory to that. It's like a dummy node. Here's how to set it up.

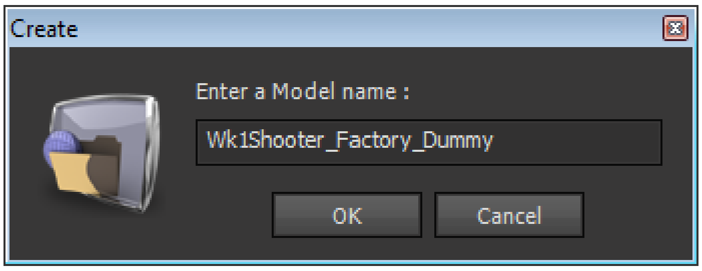
Go to the General workspace...



Then click on Create->Model->Helper...

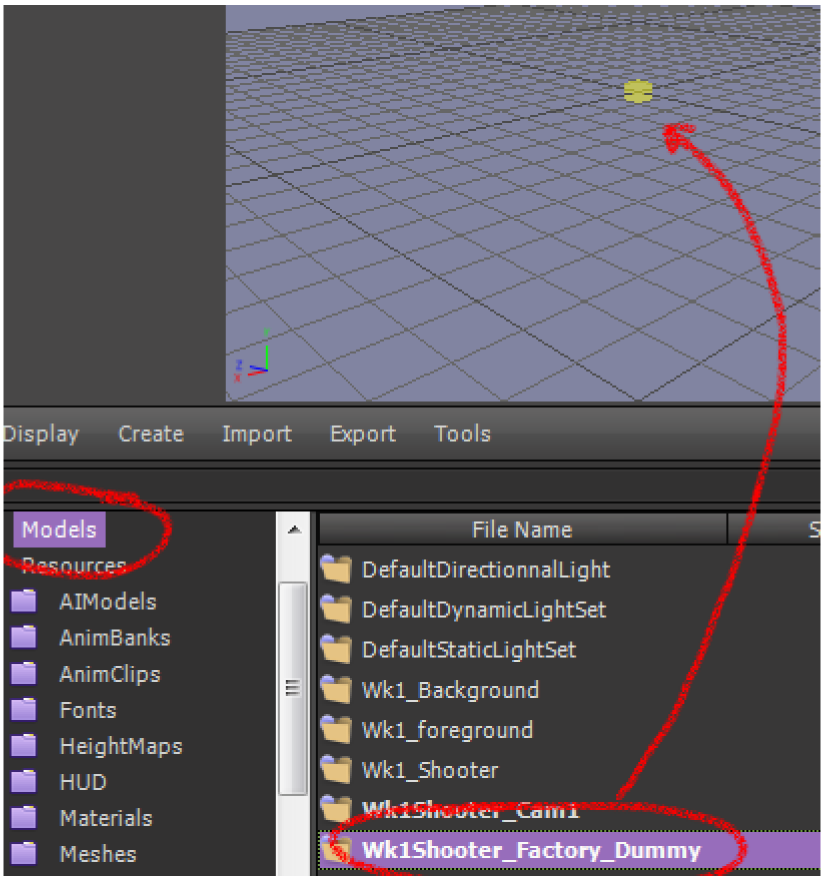


Call it Wk1Shooter_Factory_Dummy.

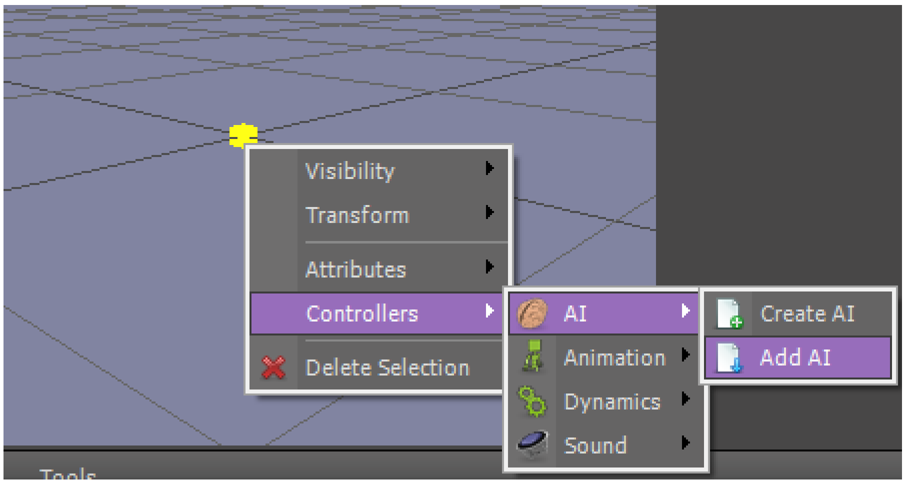


Enter a Model name : Wk1Shooter_Factory_Dummy

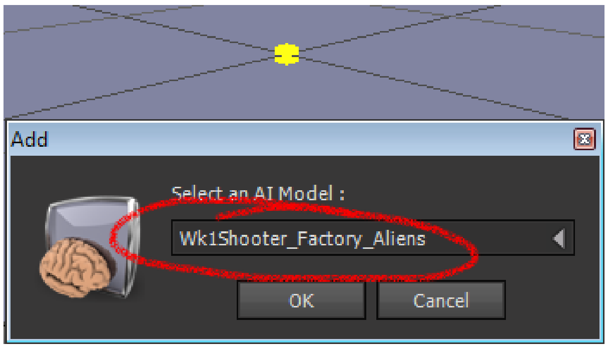
Double left click on it to open it in the Model tree to see it appear in the Scene Viewer. NOTE it is not in the scene right now.



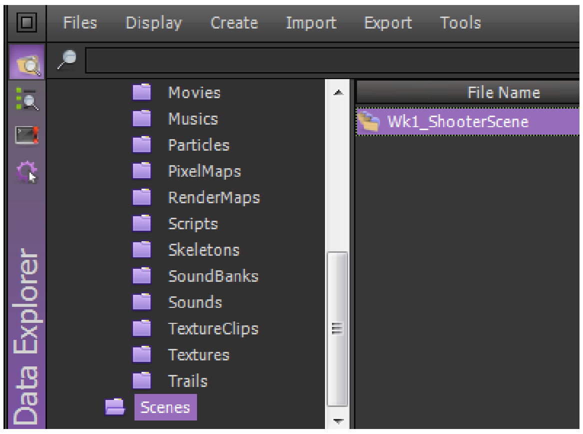
Double left click on the yellow square to zoom in. Then right click, select Control->AI->Add AI



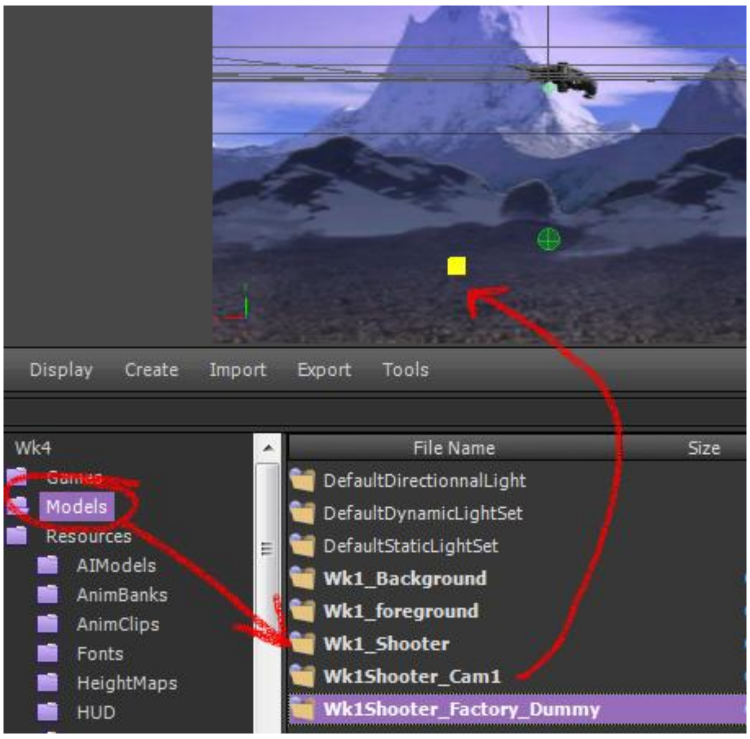
Now select the Wk1Shooter_Factory_Aliens



Now this node can be dropped into the Scene. So double left click on the your Wk1Shooter_Scene in the Scene tree...(save anything required if asked to do so)



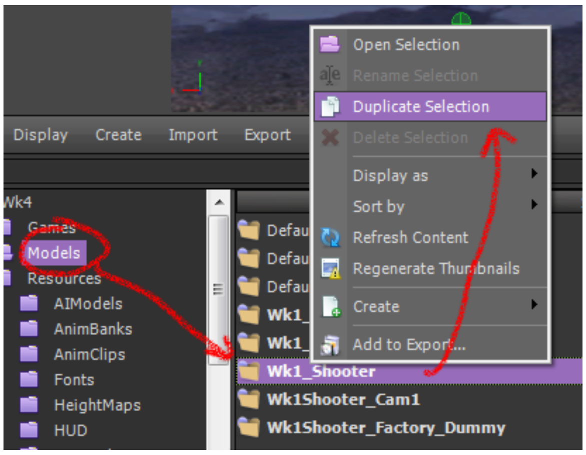
Click on the Model tree in the Data Explorer and drag N drop the Wk1Shooter_Factory_Dummy into the scene...



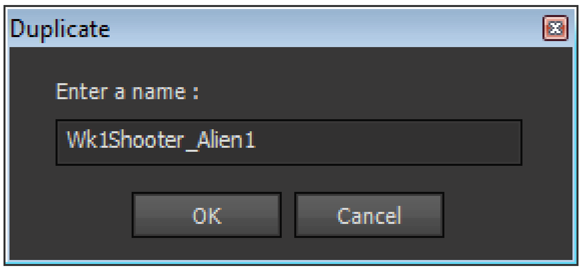
Almost – done. We can now call the factory but we need to add the Alien model first.

Create an alien.

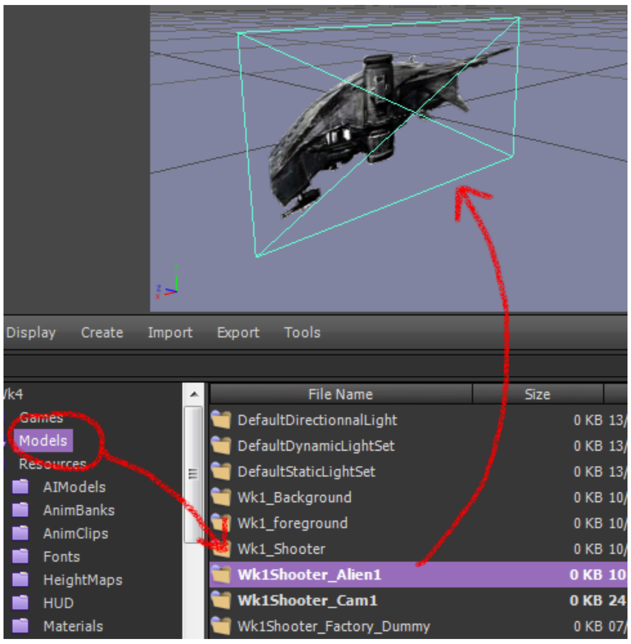
Our shooter ship is the same sort of the thing we want for the alien so let's start by duplicating that from the Model tree in Data Explorer...



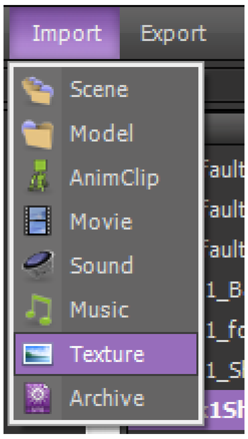
Give it the name of, you guessed it Wk1Shooter_Alien1...



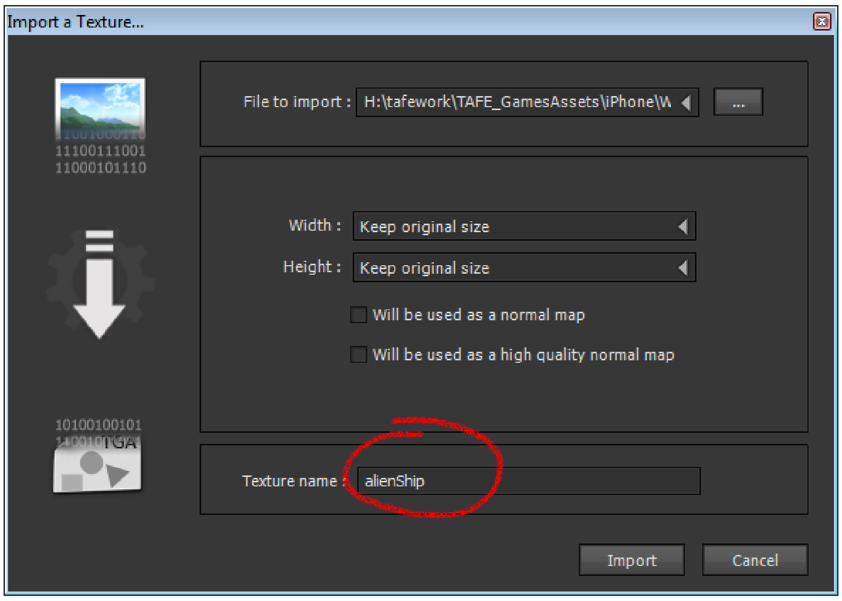
Double left click on it in the Data Explorer to open it in the Scene Viewer, double left click on it in the scene to zoom it – Yes it's the drop ship right now. NOTE it is not in the scene yet...Save anything you are asked to save.



What do we need to do? Change the material right? So click on Import ->Texture in the Data Explorer...



Load the alienShip.png texture...


Import a Texture...

File to import : H:\tafework\TAFE_GamesAssets\iPhone\W

Width : Keep original size

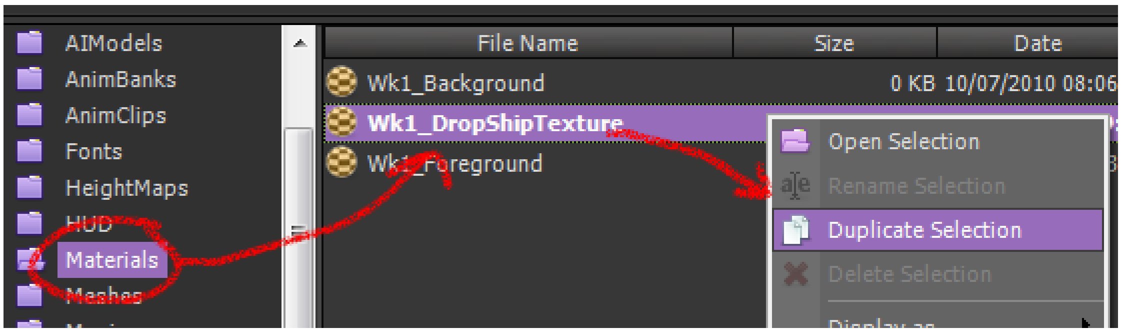
Height : Keep original size

Will be used as a normal map

Will be used as a high quality normal map

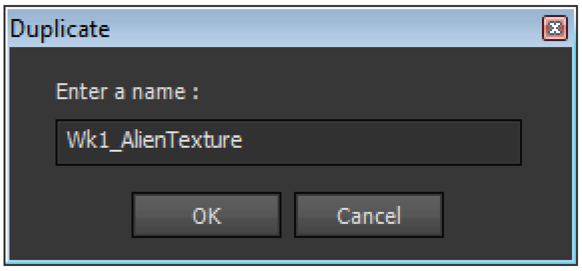
Texture name : alienShip

Now open the Material tree in the Data Explorer and duplicate the Wk1_DropShipTexture...



File Name	Size	Date
Wk1_Background	0 KB	10/07/2010 08:06
Wk1_DropShipTexture		
Wk1_Foreground		

Give it the name Wk1_AlienTexture.

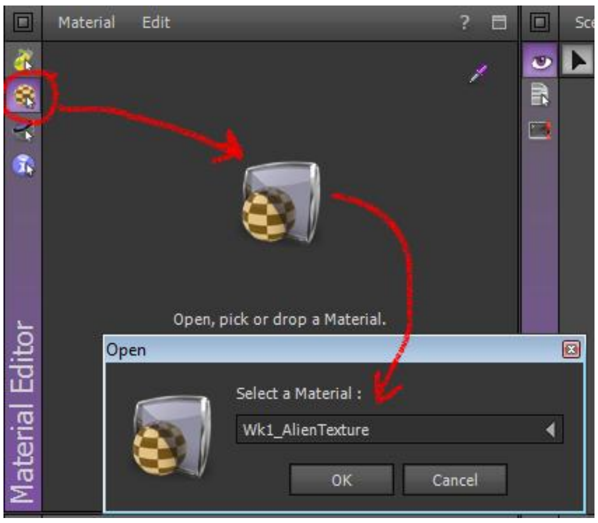

Duplicate

Enter a name :

Wk1_AlienTexture

Great! Having fun yet? We are just at the beginning of this sucker. What have we gotta do next? We need to drop the texture on the new sprite, add the AI for the alien to that model and then add it to our game as a resource. So here's how we do it.

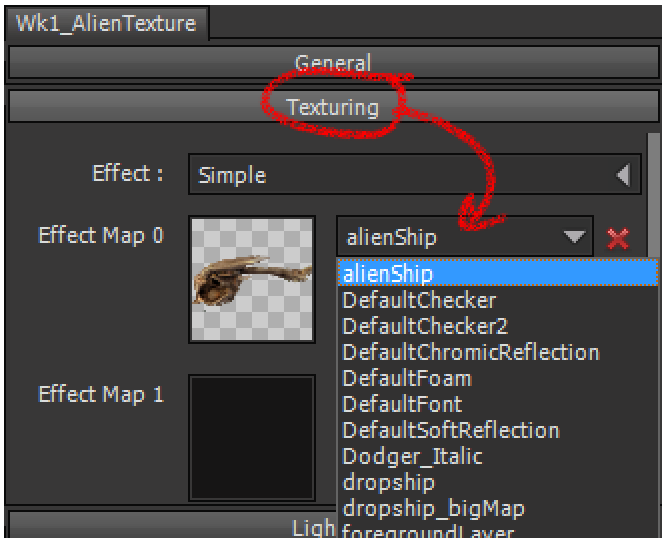
Open the Material Editor, open the Wk1_AlienTexture.



Select a Material :

Wk1_AlienTexture

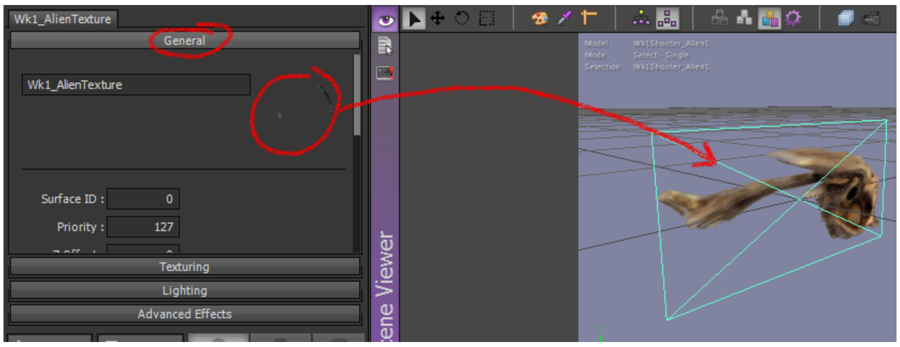
Click on the Texture tab and change it's Effect Map to alienShip...


Wk1_AlienTexture

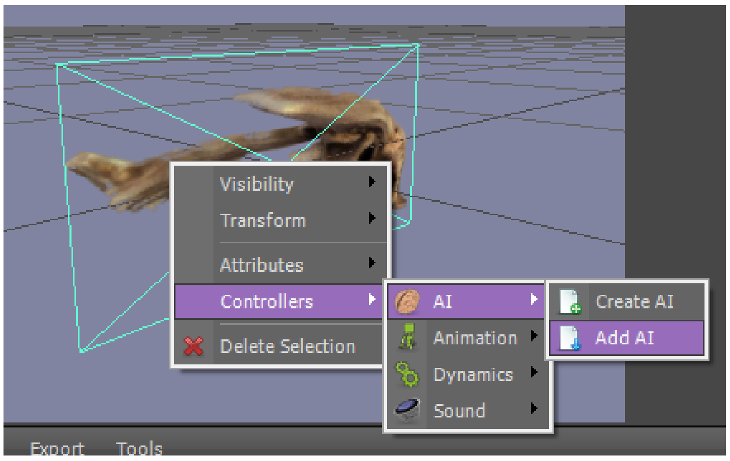
Effect : Simple

Effect Map 0 alienShip

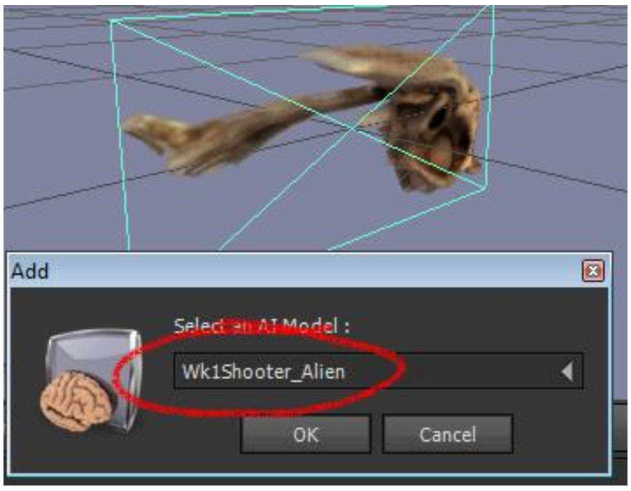
Click on the General tab, click on the swatch and drag the texture onto the mesh..



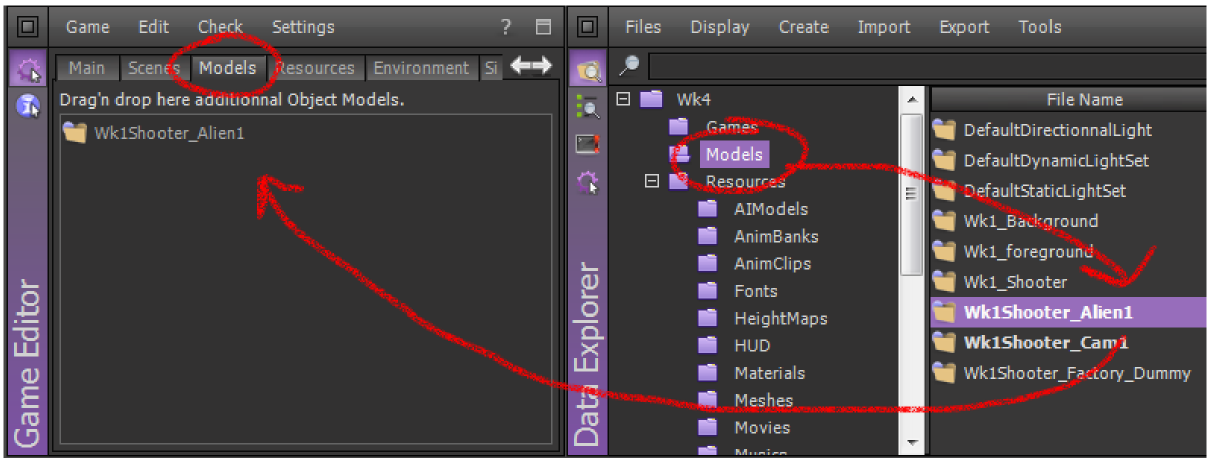
Cool – now we need to add the AI Model for it. Right click on the mesh and select Controllers->AI->Add AI...



Select the Wk1Shooter_Alien AI Model...



Now it's all attached we just need to drop it on the game. Select the Models tab in the Game Editor, then select the other Models tree in the Data Explorer. Drag and drop the Wk1Shooter_Alien into the Models tab for the Game.



Phew! Hard work hey? But now we have an alien with it's own code, and factory which can spawn aliens and some functions to control it all. Well done. What? Nothing to see yet? Sorry, have faith. Stay tuned for the next exciting instalment...(SAVE IT ALL BY CLOSING SHIVA)