CT249.II – Game Design & Programming

Report on Shiva Game, Asteroids

By Peter O'Toole
ID: 10391253

3rd March 2013

Table of Contents

Game Description

The game I choose to design was asteroids; the game is modeled in 3-D but viewed from top down which makes it 2-D from the gamer's point of view.

If asteroids hit your ship your ship will be destroyed. You have three lives, after dyeing three times you lose the game. Your ship can fly throughout the space and shoot bullets, the bullets can destroy the asteroids. For each asteroid you destroy, you get an extra fifty points of score. The goal of the game is to stay alive and to get as much score as possible.

You can use the 'up' arrow to forward trust and the 'right' and 'left' arrow to angle the ship. You can use the space bar to shoot; there is no limit to your ammo.

When you spawn and when a new level starts, your ship has a shield which protects it from asteroids; however that shield disappears after 3 seconds. Then you are on your own.

Screen shots

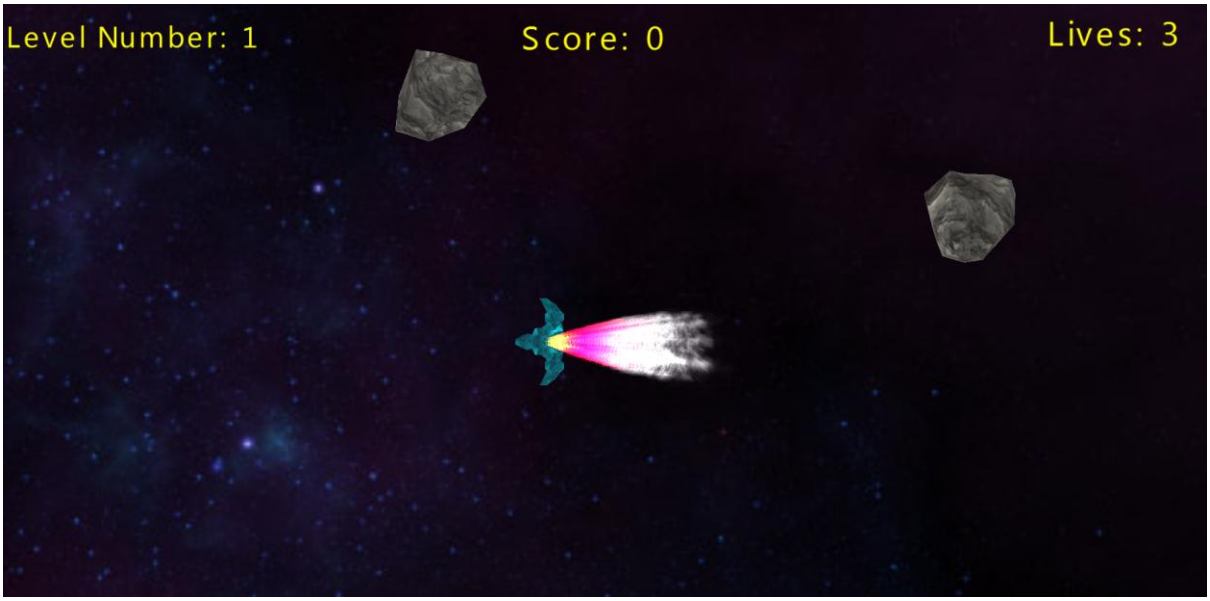
This is the opening hud, very simple, just a few labels and a play button, I would have liked to added a table of high scores using an XML file but I did not have time.

This is the opening screen when the game starts, there is a particle effect shown around the ship, this is intended as the shield in which protects the ship for the first few seconds. There are four labels on the playing hud. The level number, score and lives are constantly visible and update as each value changes. The center label starts with 'Begin' and moves to 'Next Level!', the label is displayed for the few seconds of a level and is hidden thereafter.

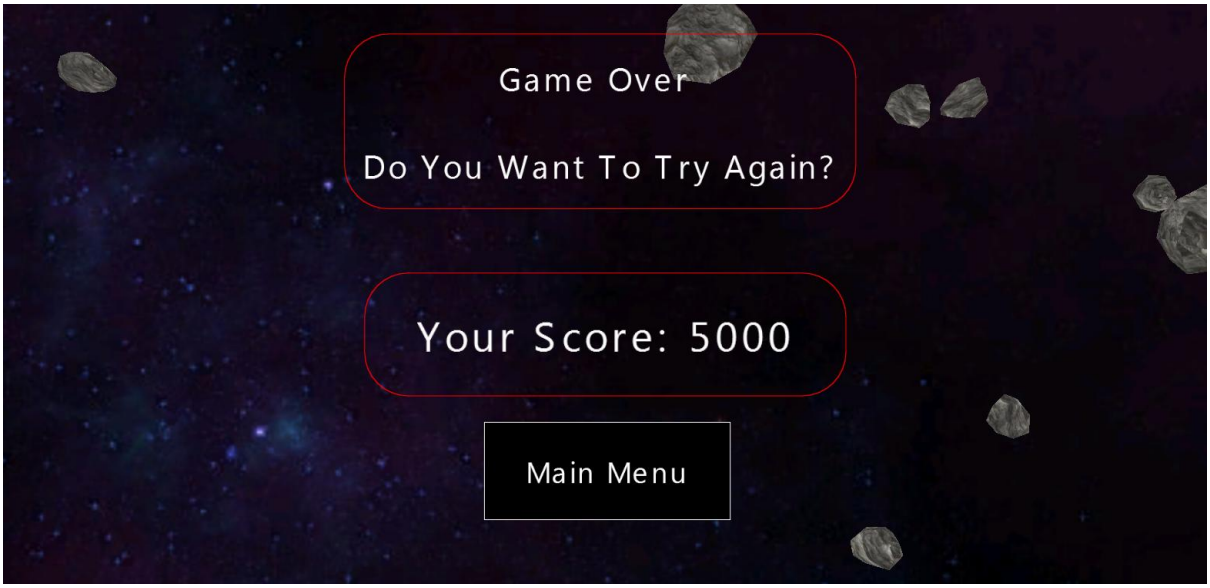
If an asteroid comes in contact with the player's ship, the ship is destroyed and the player's "lives" is deducted. There is an explosion particle emitter used to show the ships destruction. After a few second the ship respawns. If the player's lives goes to zero, the game is switched to an end game state (See below).

The player's ship can shoot, if one of those bullets collides with an asteroid, the asteroid is either broken into smaller pieces or destroyed. The player's score is also increased by fifty points.

When the player's ship is moving forward, a particle effect is shown to give the effect of an engine.

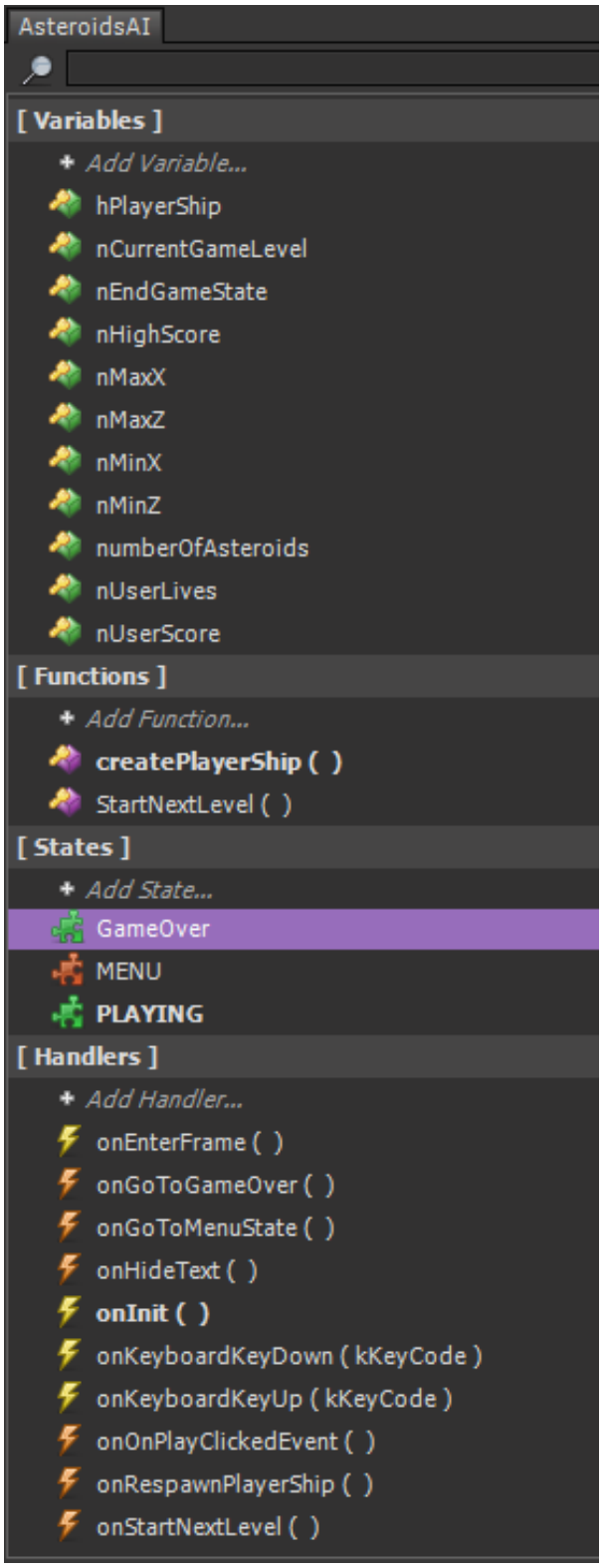


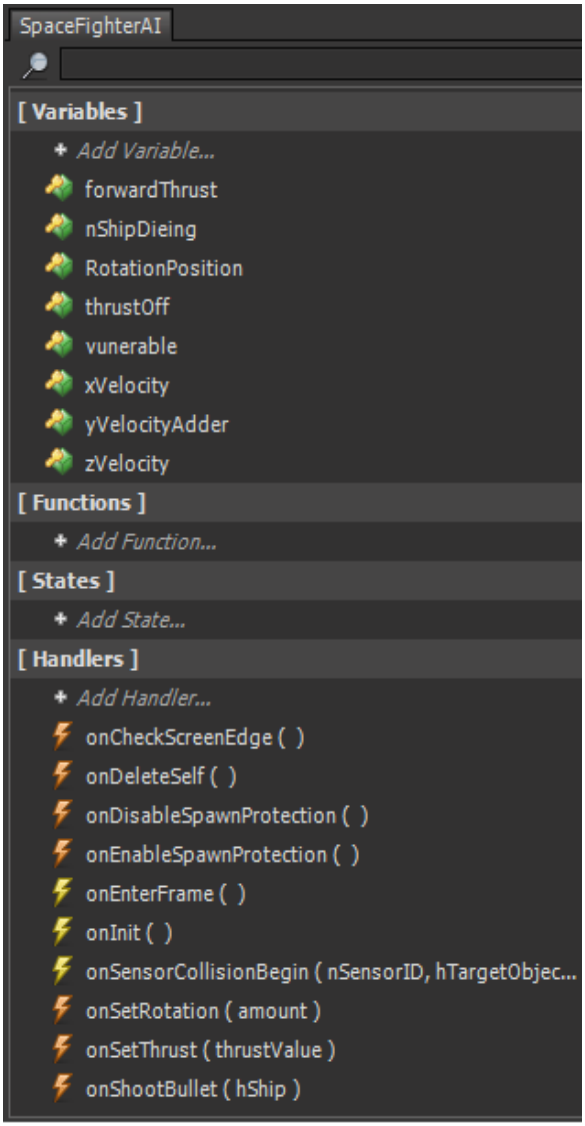
When the player's lives are depleted, the game switches to a game over state; the state displays a few labels and a button. The labels displays the high score and states that the game is over. The button brings the state back to the first hud.

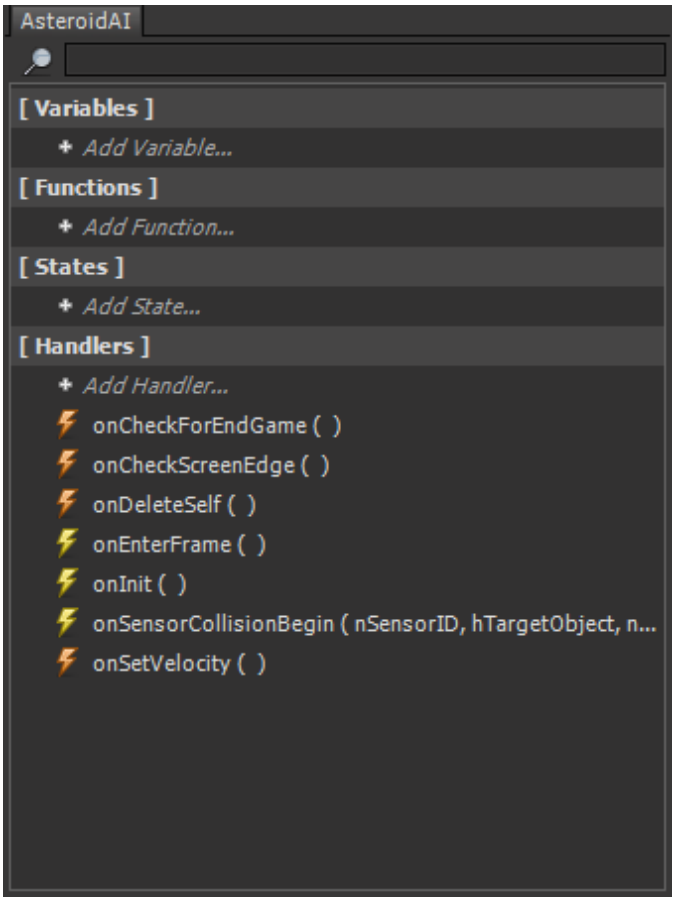


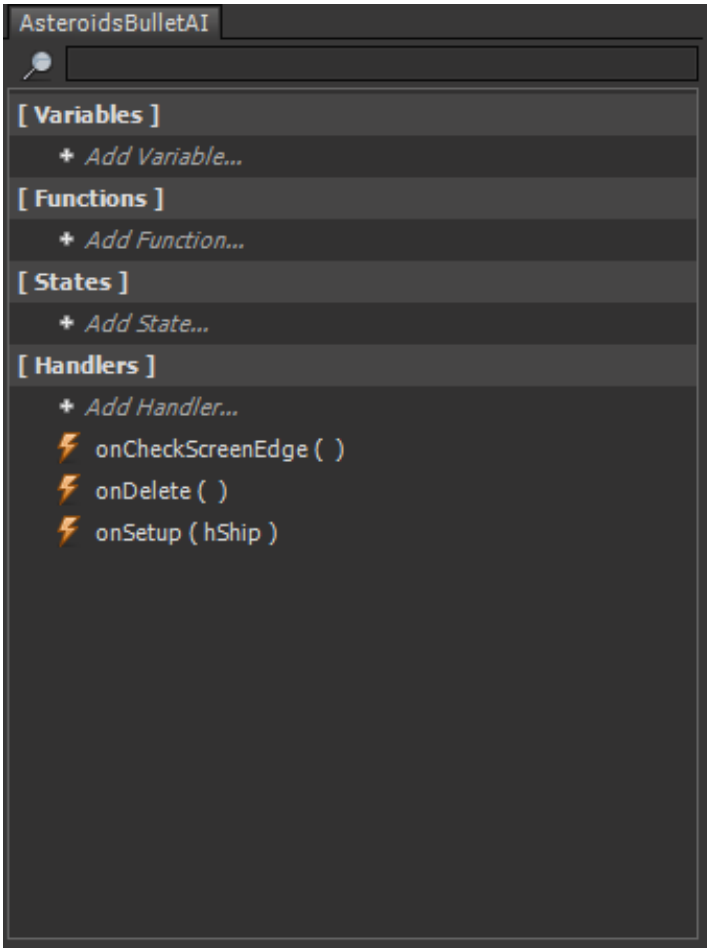
AIs

The variables, functions and handlers for each AI are shown below.






AsteroidAI
[Variables]
Add Variable...
[Functions]
Add Function...
[States]
Add State...
[Handlers]
Add Handler...
onCheckForEndGame ( )
onCheckScreenEdge ( )
onDeleteSelf ( )
onEnterFrame ( )
onInit ( )
onSensorCollisionBegin (nSensorID, hTargetObject, n...
onSetVelocity ( )


AsteroidsBulletAI
[Variables]
Add Variable...
[Functions]
Add Function...
[States]
Add State...
[Handlers]
Add Handler...
onCheckScreenEdge ( )
onDelete ( )
onSetup (hShip)

Code

All the code is below, commented as well as possible

```
-----------------------------------------------------------------
function AsteroidsAI.createPlayerShip ( )
-----------------------------------------------------------------


    local s = application.getCurrentUserScene()

    --Load players ship to game and attach the AI
    local ship = scene.createRuntimeObject (s, "spacefighter01")
    object.addAIModel (ship, "SpaceFighterAI")
    object.setTranslation (ship, 0, 0, 0, object.kGlobalSpace)
    object.setRotation (ship, 90, 0, 0, object.kGlobalSpace)
    object.setScale (ship, 0.3,0.3,0.3 )
    this.hPlayerShip(ship)



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

```
-----------------------------------------------
function AsteroidsAI.onGoToMenuState (  )
-----------------------------------------------

    --Change game state
    this.MENU ( )

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

```
-----------------------------------------------
function AsteroidsAI.onGoToGameOver (  )
-----------------------------------------------

    --Change game state
    this.GameOver ( )

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

```
--------------------------------------------------------------------------------
function AsteroidsAI.StartNextLevel ( )
--------------------------------------------------------------------------------

    --Incement level
    this.nCurrentGameLevel ( this.nCurrentGameLevel ( ) + 1 )

    --Get variables
    local s = application.getCurrentUserScene()
    local hUser = application.getCurrentUser ( )
    local levelnum = this.nCurrentGameLevel ( )

    --Update Level number on hud
    local hLevelNum = hud.getComponent ( hUser, "PayingHud.LevelLabel" )
    hud.setLabelText ( hLevelNum, string.format ( "Level Number: %d", levelnum ))

    --Update Lives on hud
    local initLives = this.nUserLives ( )
    local hLives = hud.getComponent ( hUser, "PayingHud.LivesLabel" )
    hud.setLabelText ( hLives, string.format ( "Lives: %d", initLives ) )

    --Update Score on hud
    local initScore = this.nUserScore ( )
    local hScore = hud.getComponent ( hUser,  "PayingHud.UserScore" )
    hud.setLabelText ( hScore, string.format ( "Score: %d", initScore) )

    --display center image, if the game is begining, display 'Begin' if else 'Next Level!'
    local hNextLevel = hud.getComponent ( hUser,  "PayingHud.nextLevel" )

    if(this.nCurrentGameLevel ( ) == 1 )then

        hud.setLabelText ( hNextLevel, "Begin!" )
    else
        hud.setLabelText ( hNextLevel, "Next Level!" )
    end

    --Hide centre text after one and a half seconds
    user.postEvent ( hUser, 1.5, "AsteroidsAI", "onHideText" )

    --Ensure the handle to the ship is not null, if it's not enable spawn protection, and then disable it after 1.5 seconds
    if(this.hPlayerShip ( ) ~= nil)then
        object.sendEvent ( this.hPlayerShip ( ), "SpaceFighterAI", "onEnableSpawnProtection" )
    end

    if(this.hPlayerShip ( ) ~= nil)then
        object.postEvent ( this.hPlayerShip ( ), 3, "SpaceFighterAI", "onDisableSpawnProtection" )
    end

    --Create a random number of asteroids, loads them and attach the asteroid AI
    local num = 1.5*(this.nCurrentGameLevel())
    for i = 0, num do

        this.numberOfAsteroids ( this.numberOfAsteroids ( ) + 1)

        local ast = scene.createRuntimeObject(s, "asteroid1")
        local x,z

        if(math.random(0,1)< 0.5)then
            x = math.random(this.nMinX() , this.nMinX()/2)
        else
            x = math.random(this.nMaxX() , this.nMaxX()/2)
        end

        if(math.random(0,1)< 0.5)then
            z = math.random(this.nMinZ() , this.nMinZ()/2)
        else
            z = math.random(this.nMaxZ() , this.nMaxZ()/2)
        end

        object.setTranslation(ast, x, 0, z, object.kGlobalSpace)
        object.setScale(ast, math.random(0.08, 0.12), math.random(0.08, 0.12), math.random(0.08, 0.12))
        object.addAIModel(ast, "AsteroidAI")
    end



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

```
-----------------------------------------------------------------------------
function AsteroidsAI.onHideText (  )
-----------------------------------------------------------------------------

    --Hide the centre text
    local hUser = application.getCurrentUser ( )
    local hNextLevel = hud.getComponent ( hUser,  "PayingHud.nextLevel" )
    hud.setLabelText ( hNextLevel, "" )

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

```
function AsteroidsAI.onInit (  )
-----------------------------------------------------------------------------

    --Set current user scene
    application.setCurrentUserScene("AsteroidsScene1", "")
    local s = application.getCurrentUserScene()
    local hcam = application.getCurrentUserActiveCamera( )

    --Set cameras position
    object.setTranslation (hcam, 0, 40, 0, object.kGlobalSpace)
    object.lookAtWithUp (hcam, 0, 0, 0, 0, 0, 1, object.kGlobalSpace, 1)
    local notUsed, notUsed2,targetPlane = camera.projectPoint(hcam, 0,0,0)
    local x1, y1, z1 =camera.unprojectPoint(hcam, -1, -1, targetPlane )
    local x2, y2, z2 =camera.unprojectPoint(hcam, 1, 1, targetPlane )

    if(x1>x2)then
        local temp=x1
        x1=x2
        x2 = temp
    end
    if(z1>z2) then
        local temp = z1
        z1 = z2
        z2 = temp
    end

     --Multiply all values by 1.08 to get a small offScreen value
    this.nMinX(x1*1.08)
    this.nMaxX(x2*1.08)
    this.nMinZ(z1*1.08)
    this.nMaxZ(z2*1.08)


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

```
--------------------------------------------------------------------------------
function AsteroidsAI.onKeyboardKeyDown ( kKeyCode )
--------------------------------------------------------------------------------

    --ensure the handle is not null, when keys are pressed, call approperate methods inside ship AI
    if(this.hPlayerShip ( ) ~= nil)then
        if(kKeyCode==input.kKeyLeft)then
            object.sendEvent ( this.hPlayerShip ( ), "SpaceFighterAI", "onSetRotation", 3 )
        elseif(kKeyCode==input.kKeyRight)then
            object.sendEvent ( this.hPlayerShip ( ), "SpaceFighterAI", "onSetRotation", -3 )
        elseif(kKeyCode==input.kKeyUp)then
            object.sendEvent ( this.hPlayerShip ( ), "SpaceFighterAI", "onSetThrust", true )
        elseif(kKeyCode==input.kKeySpace)then
            object.sendEvent ( this.hPlayerShip ( ), "SpaceFighterAI", "onShootBullet")
        end
    end
--------------------------------------------------------------------------------
end
```

```
--------------------------------------------------------------------------------
function AsteroidsAI.onKeyboardKeyUp ( kKeyCode )
--------------------------------------------------------------------------------

    --If the ship handle is not null, call apporiate methods
    if(this.hPlayerShip ( ) ~= nil)then
        if(kKeyCode==input.kKeyLeft)then
            object.sendEvent ( this.hPlayerShip ( ), "SpaceFighterAI", "onSetRotation", 0 )
        elseif(kKeyCode==input.kKeyRight)then
            object.sendEvent ( this.hPlayerShip ( ), "SpaceFighterAI", "onSetRotation", 0 )
        elseif(kKeyCode==input.kKeyUp)then
            object.sendEvent ( this.hPlayerShip ( ), "SpaceFighterAI", "onSetThrust", false )
        end
    end

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

```
------------------------------------------------
function AsteroidsAI.onOnPlayClickedEvent (  )
------------------------------------------------

    --Change state
    this.PLAYING ( )

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

```
function AsteroidsAI.onRespawnPlayerShip (  )
------------------------------------------------

    --Recreate splayer ship
    this.createPlayerShip()


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

```
function AsteroidsAI.onStartNextLevel (  )
-------------------------------------------------

    --Start the next level
    this.StartNextLevel ( )

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

```
function SpaceFighterAI.onCheckScreenEdge (  )
-----------------------------------------------------------------------------

    --Get object handle and possition
    local ship = this.getObject ( )
    local x,y,z = object.getTranslation ( ship, object.kGlobalSpace )

    --Get variables from user Main AI
    local minX = application.getCurrentUserAIVariable ( "AsteroidsAI", "nMinX" )
    local maxX = application.getCurrentUserAIVariable ( "AsteroidsAI", "nMaxX" )
    local minZ = application.getCurrentUserAIVariable ( "AsteroidsAI", "nMinZ" )
    local maxZ = application.getCurrentUserAIVariable ( "AsteroidsAI", "nMaxZ" )

    --If the player's ship left the screen, we want to bring it back to the other side
    if(x<minX)then
        object.setTranslation ( ship, x+(maxX-minX), 0,z, object.kGlobalSpace )
    elseif(x>maxX)then
        object.setTranslation ( ship, x-(maxX-minX), 0,z, object.kGlobalSpace )
    end
    if(z<minZ)then
        object.setTranslation ( ship, x, 0, z+(maxZ-minZ), object.kGlobalSpace )
    elseif(z>maxZ)then
        object.setTranslation ( ship, x, 0, z-(maxZ-minZ), object.kGlobalSpace )
    end

    --Check again in a fraction of a second
    this.postEvent (0.2, "onCheckScreenEdge" )

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

```
function SpaceFighterAI.onDeleteSelf (  )
-----------------------------------------------------------------------------

    --Set variable and delete the object
    this.nShipDieing ( false )
    scene.destroyRuntimeObject (application.getCurrentUserScene ( ) , this.getObject ( ) )

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

```
function SpaceFighterAI.onDisableSpawnProtection (  )
-------------------------------------------------------------

    --Disable spawn protection and stop particle effects
    this.vunerable ( true )
    sfx.stopParticleEmitterAt( this.getObject ( ), 2 )

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

```
function SpaceFighterAI.onEnableSpawnProtection (  )
-------------------------------------------------------------

    --Start the spawn protection and start particle emmiter.
    this.vunerable ( false )
    sfx.startParticleEmitterAt ( this.getObject ( ), 2 )

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

```
function SpaceFighterAI.onEnterFrame (  )
-----------------------------------------------------------------------------

    --Get Ship Handler
    local ship = this.getObject ( )
    local xVel = this.xVelocity ( )
    local zvel = this.zVelocity ( )

    --check if the ship is destroyed
    if(not this.nShipDieing ( ))then
        --If the forward thrust is off and its greater than zero, decrement the speed
        if( (this.thrustOff()) and (this.forwardThrust()>0) )then
            this.forwardThrust ( this.forwardThrust ( ) - 0.003)
        end

        if( this.forwardThrust()>0)then
            --update position of ship
            object.translate(ship, (-zvel*0.01*this.forwardThrust( )) , 0, (xVel*0.01*this.forwardThrust( )), object.kGlobalSpace)
        end

        --Get the rotational position of the ship
        local x, y, z = object.getRotation ( ship, object.kGlobalSpace )

        this.xVelocity( x )
        this.zVelocity( z )

        --update rotational position of the ship
        object.rotate ( ship, 0, this.RotationPosition ( ), 0, object.kGlobalSpace )
    end
-----------------------------------------------------------------------------
end
```

```
function SpaceFighterAI.onInit (  )
-------------------------------------------------------------------------

    --Get ship handle
    local ship = this.getObject ( )

    --stop all the particle emitters
    sfx.stopParticleEmitterAt( ship, 0 )
    sfx.stopParticleEmitterAt( ship, 1 )
    sfx.stopParticleEmitterAt( ship, 2 )
    --Start the shield particle emitters
    sfx.startParticleEmitterAt( ship, 2 )
    --Enable dynamics and ensure the ship stays in the right planes
    dynamics.createSphereBody(ship, 1)
    dynamics.enableGravity(ship, false)
    dynamics.enableDynamics(ship, true)
    dynamics.enableGravity(ship, false)
    dynamics.setGuardBox ( ship, -1000, 0, -1000, 1000, 0, 1000 )
    dynamics.enableGuardBox ( ship, true )

    --Check if the player's ship has left the screen
    this.postEvent ( math.random ( 0.02, 0.2 ), "onCheckScreenEdge" )
    --Send a delayed event to stop the spawn protection
    this.postEvent ( 4, "onDisableSpawnProtection")

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

```
function SpaceFighterAI.onSensorCollisionBegin ( nSensorID, hTargetObject, nTargetSensorID )
-------------------------------------------------------------------------------

    local hShip = this.getObject ( )

    if(this.vunerable ( ) and ( not this.nShipDieing ( )))then

        if(nTargetSensorID==0)then
            this.nShipDieing ( true )

            local lives = application.getCurrentUserAIVariable ( "AsteroidsAI", "nUserLives")
            lives = (lives - 1)

            local hUser = application.getCurrentUser ( )
            local hLives = hud.getComponent ( hUser, "PayingHud.LivesLabel" )
            hud.setLabelText ( hLives, string.format ( "Lives: %d", lives ))

            if(lives>0)then

                sfx.startParticleEmitterAt (hShip,  1 )
                object.setScale ( hShip, 0.00001, 0.00001, 0.00001)
                this.postEvent ( .7, "onDeleteSelf" )

                user.postEvent ( application.getCurrentUser ( ), 4, "AsteroidsAI", "onRespawnPlayerShip" )
                application.setCurrentUserAIVariable ( "AsteroidsAI", "nUserLives", lives )

            else

                sfx.startParticleEmitterAt (hShip,  1 )
                object.setScale ( hShip, 0.00001, 0.00001, 0.00001)
                this.postEvent ( .7, "onDeleteSelf" )

                user.postEvent ( application.getCurrentUser ( ), 3, "AsteroidsAI", "onGoToGameOver" )
            end
        end
    end
-------------------------------------------------------------------------------
end
```

```
function SpaceFighterAI.onSetRotation (  amount)
-----------------------------------------------------

    --Setting the ships rotation
    this.RotationPosition (amount)

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

```
function SpaceFighterAI.onSetThrust ( thrustValue )
-------------------------------------------------------------------------

    local ship = this.getObject ( )

    --Setting the ships thrust, starting and stopping the particle emitter

    if(not this.nShipDieing ( ))then
        --change ship velocity
        if(thrustValue)then
            this.forwardThrust(0.2)
            this.thrustOff ( false )
            sfx.startParticleEmitterAt (ship, 0 )
        else
            this.thrustOff ( true )
            sfx.stopParticleEmitterAt ( ship, 0 )
        end
    end
-------------------------------------------------------------------------
end
```

```
function SpaceFighterAI.onShootBullet ( hShip)
-------------------------------------------------------------------------

    --create bullet
    local s = application.getCurrentUserScene ( )
    local b = scene.createRuntimeObject ( s, "Bullet" )

    --Add the AI, set up the bullet
    object.addAIModel ( b, "AsteroidsBulletAI" )
    object.sendEvent ( b, "AsteroidsBulletAI", "onSetup", this.getObject ( ))


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

```
function AsteroidAI.onCheckForEndGame (  )
-------------------------------------------------------------------------------

    --Get value from Main AI
    local isEndOfGame = application.getCurrentUserAIVariable ( "AsteroidsAI", "nEndGameState" )

    --If it is the end of the game, all asteroids should delete themselfs
    if(isEndOfGame)then
        scene.destroyRuntimeObject (application.getCurrentUserScene ( ) , this.getObject ( ) )
    end

    --Check again in one second
    this.postEvent (1, "onCheckForEndGame" )
-------------------------------------------------------------------------------
end
```

```
function AsteroidAI.onCheckScreenEdge (  )
-------------------------------------------------------------------------------------

    --Get the objects handle and get its translation
    local ast = this.getObject ( )
    local x,y,z = object.getTranslation ( ast, object.kGlobalSpace )
    --Get the edge of the screen varibles
    local minX = application.getCurrentUserAIVariable ( "AsteroidsAI", "nMinX" )
    local maxX = application.getCurrentUserAIVariable ( "AsteroidsAI", "nMaxX" )
    local minZ = application.getCurrentUserAIVariable ( "AsteroidsAI", "nMinZ" )
    local maxZ = application.getCurrentUserAIVariable ( "AsteroidsAI", "nMaxZ" )
    --If the object leaves the screen, bring it to the other side of the screen
    if(x<minX)then
        object.setTranslation ( ast, x+(maxX-minX), 0,z, object.kGlobalSpace )
    elseif(x>maxX)then
        object.setTranslation ( ast, x-(maxX-minX), 0,z, object.kGlobalSpace )
    end
    if(z<minZ)then
        object.setTranslation ( ast, x, 0, z+(maxZ-minZ), object.kGlobalSpace )
    elseif(z>maxZ)then
        object.setTranslation ( ast, x, 0, z-(maxZ-minZ), object.kGlobalSpace )
    end
    --Check again in a fraction of a second
    this.postEvent (0.2, "onCheckScreenEdge" )
-------------------------------------------------------------------------------------
end
```

```
function AsteroidAI.onDeleteSelf (  )
-------------------------------------------------------------------------

    --A handle to destroy the object
    scene.destroyRuntimeObject (application.getCurrentUserScene ( ) , this.getObject ( ) )
-------------------------------------------------------------------------
end
```

```
function AsteroidAI.onInit (  )
-------------------------------------------------------------------------------

    --Get object handle and the current level
    local ast = this.getObject ( )
    local nGameLevel = application.getCurrentUserAIVariable ( "AsteroidsAI" , "nCurrentGameLevel")
    --Enable collisions and set bounding box to keep the asteroid on the right plane
    local r = object.getBoundingSphereRadius ( ast )
    dynamics.createSphereBody(ast, r*0.5)
    dynamics.enableCollisions ( ast, true )
    dynamics.setMass ( ast, r*r*r )
    dynamics.setGuardBox ( ast, -1000, 0, -1000, 1000, 0, 1000 )
    dynamics.enableGuardBox ( ast, true )
    dynamics.enableGravity(ast, false)
    dynamics.enableDynamics(ast, true)

    --Set the asteroid to a random velocity based on the game level
    local vx = 5*math.random(-nGameLevel, nGameLevel)
    local vz = 5*math.random(-nGameLevel, nGameLevel)
    dynamics.setLinearVelocity(ast, vx, 0, vz, object.kGlobalSpace)
    --Set speed limit
    dynamics.setLinearSpeedLimit (ast, 10*nGameLevel)
    dynamics.setAngularSpeedLimit ( ast, 10 )

    --Setting angular velocity
    local avx = math.random ( -2, 2 )
    local avy = math.random ( -2, 2 )
    local avz = math.random ( -2, 2 )
    dynamics.setAngularVelocity ( ast, avx, avy, avz, object.kGlobalSpace)
    dynamics.setLinearDamping(ast, 0)
    dynamics.setAngularDamping(ast, 0)


    --Check if the ateroid has left the screen and check if the game is over.
    this.postEvent ( math.random ( 0.02, 0.2 ), "onCheckScreenEdge" )
    this.postEvent (0.05, "onCheckForEndGame" )

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

```
function AsteroidsBulletAI.onDelete (  )
-------------------------------------------------------------------------------
    --Delete this object
    scene.destroyRuntimeObject ( application.getCurrentUserScene(), this.getObject ( ) )

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

```
function AsteroidAI.onSensorCollisionBegin ( nSensorID, hTargetObject, nTargetSensorID )
--------------------------------------------------------------------------------

    --Get object handle, current user, possition, linear velocity, user scene, calculate size and get varible from User Ma
    local o = this.getObject ( )
    local hUser = application.getCurrentUser ( )
    local x, y, z = object.getTranslation ( o, object.kGlobalSpace )
    local vx,vy,vz = dynamics.getLinearVelocity ( o, object.kGlobalSpace )
    local hScene = application.getCurrentUserScene()
    local sx, sy, sz = object.getScale ( o )
    local size = (sx+sy+sz )/3
    local numberAsteroids = application.getCurrentUserAIVariable ( "AsteroidsAI", "numberOfAsteroids")


    --If the object that comes in contact with the asteriod is a bullet, do the following.
    if( nTargetSensorID == 10 ) then

        --Destroy the bullet
        object.sendEvent ( hTargetObject, "AsteroidsBulletAI", "onDelete" )

        --Increase the score
        local score = application.getCurrentUserAIVariable ( "AsteroidsAI", "nUserScore")
        score = (score + 50)
        application.setCurrentUserAIVariable ("AsteroidsAI", "nUserScore", score )

        --update the current hud
        local hUser = application.getCurrentUser ( )
        local hScore = hud.getComponent ( hUser, "PayingHud.UserScore" )
        hud.setLabelText ( hScore, string.format ( "Score: %d", score ))

        --If the object is big enough split it into pieces, otherwise destory the entire thing.
        if(size>0.07)then
            local numPieces

            if(size>1.1)then
                numPieces = 3
            else
                numPieces = 2
            end

            --Calculate Piece size
            local pieceSize = size/numPieces

            --Create that number of pieces
            for i=1, numPieces do

                --Update the number of asteroids
                numberAsteroids = (numberAsteroids + 1)
                --Create the asteroid
                local a = scene.createRuntimeObject ( hScene, "asteroid1" )
                local localVelX = math.random(-1, 1)
                local localVelZ = math.random(-1, 1)
                --set the possition, scale, AI model
                object.setTranslation ( a, x+localVelX, 0, z+localVelZ, object.kGlobalSpace )
                object.setScale (a, math.random(pieceSize*0.8, pieceSize*1.2), math.random(pieceSize*0.8, pieceSize*1.2),
                math.random(pieceSize*0.8, pieceSize*1.2) )

                object.addAIModel (a, "AsteroidAI")
                object.postEvent (a, 0.05, "AsteroidAI", "onSetVelocity", vx+localVelX, vz+localVelZ)
            end
            --destroy the current asteroid
            scene.destroyRuntimeObject (hScene, o )
        else
            --tell the asteroid to delete itself in half a second
            this.postEvent (.5, "onDeleteSelf" )
            --set it's speed and size to more or less zero, start particale emmiter.
            --the particles start straight away, after half a second, it is deleted and the emitter stops
            dynamics.setLinearVelocity(o, 0, 0, 0, object.kGlobalSpace)
            dynamics.setAngularVelocity ( o, 0, 0, 0, object.kGlobalSpace)
            object.setScale ( o, 0.00001, 0.00001, 0.00001)
            sfx.startParticleEmitterAt (o, 0 )
        end


        --Decrease the number of asteroids and send the update to the user Main AI
        numberAsteroids = (numberAsteroids - 1)
        application.setCurrentUserAIVariable ( "AsteroidsAI", "numberOfAsteroids", numberAsteroids)
```

```
function AsteroidsBulletAI.onSetup ( hShip )
-------------------------------------------------------------------------------

    --Get the object handle
    local b = this.getObject ( )

    --Match the translation and rotation of the ship
    object.matchTranslation ( b, hShip, object.kGlobalSpace )
    object.matchRotation ( b, hShip, object.kGlobalSpace )
    --Set the size of the bullet
    object.setScale ( b, 0.3,0.3,0.3 )

    --Create a bounding shpere to ensure the bullet doesn't move out of plane
    local r = object.getBoundingSphereRadius ( b )
    dynamics.createSphereBody(b, r*0.5)
    dynamics.setGuardBox(b, -1000, 0, -1000, 1000, 0, 1000)
    dynamics.enableGuardBox ( b, true )
    dynamics.enableGravity ( b, false )
    dynamics.enableDynamics ( b, true )

    --Set the foward velocity of the bullet
    dynamics.setLinearVelocity ( b, 0, 20, 0, object.kLocalSpace )
    --Stop linear and angular damping
    dynamics.setAngularDamping ( b,0 )
    dynamics.setLinearDamping ( b,0 )

    --Start the bullet checking if it left the screnen or not
    this.postEvent ( 0.2, "onCheckScreenEdge" )


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

Creation of Models, Graphics & Code

The skybox image and all models used came from course material. All particle effects were created by me. The graphics for huds were created by me using Shiva. A small portion of the code was taken from sample solutions given in labs, the rest was written by me.

Conclusion

In short, I am glad I choose to take Game Design & Programming, I have learned a lot about 3-D Games over the past eight weeks. I wish the semester was longer so that I could have created a more complex game, for example a first person shooter. However now that I know the basics I can start to work on more complex things in my free time. I will definitely be taking a follow on to this course next year.