# Pseudocode

## Overarching Flow Chart

Call LoadSaveGame

Initialise Renderer

Initialise Direct3D

Initialise Game Manager

Initialise Physics

Initialise Sound Manager

Initialise Texture Loader

Initialise Text

Initialise Lighting

Load Direct3D 3D models and render to scene

Ship

Asteroid

SkyBox

Load SFX & Music files into memory

Set GameRunning bool to false

Render scene to screen

Draw buttons to screen

Play button

Settings button

Quit button

Window Input

Switch

Player pressed settings button?

Display settings

Player pressed quit button?

Clean-Up

Exit program

Player pressed high scores button?

Display high scores

Player pressed play button?

Set coordinates of objects to save game data

Set GameRunning bool to true

End switch

Loop

Input thread

Input loop

Did the player press the fire button?

Fire projectile

Did the player move the mouse/joystick?

Move the player

Did the player press escape?

Set GameRunning to false

Activate pause menu

Did the player defeat all enemies?

Display win screen

Did the player die?

Display loss screen

Loop

AI thread

Is there an enemy within X distance?

Is there an enemy within max shooting distance?

Move into range of enemy

Is an enemy in line of fire of the AI’s turrets?

Turn to face enemy

Are the weapons off cooldown?

Fire the weapons

Has the enemy died?

Activate entity died routine

Has the win condition been met?

Clean-up

Physics thread

For each object

Check for collisions

Perform calculations to check for collisions before moving

Move object if collisions aren’t going to happen

Apply angular velocity to all objects

loop

Sound thread

Sound loop

Are sounds currently playing?

Continue playing

Is a new sound required?

Play sound

Is GameRunning false?

Pause all sounds

Is the game win condition met?

Clean-up

loop

Main game loop

Is player dead?

Display loss screen

Close program

LoadSaveGame

Start

Open save file for reading

Do until end of file

Read Save #N (Coords: X, Y, Z, Rotation: X, Y, Z)

For each object

Read Save #N (Coords: X, Y, Z, Rotation: X, Y, Z)

loop

loop

Close save file

End

## Settings [A]

For each main menu button

Is the button visible?

Hide button

Loop

For each settings control

Is the button invisible?

Unhide button

Loop

Input player loop

Did the player press the increase SFX volume button?

Is the SFX volume not 100?

Sound -> increase SFX volume

Did the player press the decrease SFX volume button?

Is the SFX volume above 0?

Sound -> decrease SFX volume

Did the player press the increase music volume button?

Is the music volume not 100?

Sound -> increase music volume

Did the player press the decrease music volume button?

Is the music above 0?

Sound -> decrease music volume

Did the press the fullscreen/windowed toggle button?

Is the window windowed?

Call direct3d change screen setting function (fullscreen)

Is the window fullscreen?

Call direct3d change screen setting function (windowed)

Did the player press the escape key or back button?

For each settings button

Hide button

loop

for each main menu button

unhide button

loop

exit loop

loop

## Quit Game [B]

Is there a save file already saved in the default location?

Open the text file for editing

Save Player coordinates (Location: X, Y, Z, Rotation: X, Y, Z)

Save object coordinates (Location: X, Y, Z, Rotation: X, Y, Z)

Close text file

Is there no save file in the default location?

Open new text file for editing

Save Player coordinates (Location: X, Y, Z, Rotation: X, Y, Z)

Save object coordinates (Location: X, Y, Z, Rotation: X, Y, Z)

Close text file

Call clean-up function

## High Scores [C]

For each main menu button

Hide button

Loop

For each high scores button

Unhide button

Loop

Open hide scores text file for editing

Do until eof

Get Score & Player name

Loop

For i = lower bound of score & name array to upper bound of array

Sort from highest to lowest, giving a rank number

Next i

Display list of high scores onto screen

Close high scores text file

## Fire [D]

Is the cooldown bool false for the entities weapon?

Particle system -> Set projectile to visible

Particle system -> Move projectile to turret end

Particle system -> Set forward velocity and rotation of projectile

Set the cooldown bool to true

Set the cooldown time to current time plus X seconds

Each frame

Advance projectile forward by velocity amount

Has the object collided with something?

Is the collision against an enemy?

Apply damage to the enemy entity

Is the enemy dead?

Add to score

Play death animation

Play death sound

Is the enemy alive?

AI -> Received damage function

Is the collision not against an enemy?

Set projectile velocity to 0

Return projectile to starting location

Hide the projectile

Has the object not collided?

Has the projectile reached the distance limit?

Set projectile velocity to 0

Return projectile to starting location

Hide the projectile

Next frame

## Movement [E]

Input -> get new mouse coordinates

Camera.rotation.x, y & z = Input.Mouse.x, y & z

Is the player pressing the accelerate key?

Is the player at peak velocity?

Keep velocity as it is

Move entity (and check collision [G])

Is the player under peak velocity?

Velocity += speed

Move entity (and check collision [G])

Is the player pressing the brake key?

Is the player moving?

Velocity -= braking

Move entity (and check collision [G])

Is the player still?

Maintain 0 velocity

## Pause Menu [F]

Set GameRunning to false

for each pause menu graphic

unhide graphic

loop

get input loop

did the player click the quit button?

Quit Game [B]

Did the player click the resume button?

For each pause menu graphic

Hide graphic

loop

Set GameRunning to true

Exit loop

Loop

## Collision Detection [G]

Is there an object within a certain radius of our object (within collision distance)?

For each quarter of each model

Is the collide able object in the vicinity?

For each vertex in our model

Did the vertex collide with the object?

Apply collision damage and send message upwards to collided object

loop

loop

Awejndjad dawjdwajd