**Appendix D – Feature Priority List**

Phase One - Models

1. Search for models
2. Download appropriate models
3. Load models into 3D Studio Max
4. Load models into XNA
5. Test models all can be complied and loaded
6. Render models to the XNA view window
7. Test and search for broken vertices

Phase Two - Graphics

1. Read and understand Riemer’s Tutorial’s
2. Use information from tutorial to create a Terrain
3. Use information from tutorial to create a Sky Dome
4. Set up the camera details needed, by using Riemer’s tutorials
5. Create the first person in car camera view
6. Create the third person behind the car view
7. Test the camera follows the car’s position in both views

Phase Three – Advance Graphics

1. Search and download a relevant height map
2. Alter sizes and details of Height map if needed
3. Use the tutorial by Riemer to load the height map
4. Use the tutorial by Riemer to render the height map
5. Test the different colours on the terrain
6. Create camera delays when turning left or right
7. Test the delays of the camera
8. Use the height information to detach collision
9. Test the collision at several points of the map

Phase Four – Refactoring

1. Locate all the functions that are related to the Terrain
2. Move the Terrain functions into a new class
3. Create the Terrain object in the Game class
4. Test that the Terrain can still be created and rendered
5. Locate all the functions that are related to the Sky Dome
6. Move the Sky Dome functions into a new class
7. Create the Sky Dome object in the Game class
8. Test that the Sky Dome can still be created and rendered
9. Locate all the functions that are related to the Car
10. Move the Car functions into a new class
11. Create the Car object in the Game class
12. Test that the Car can still be created and rendered
13. Locate all the functions that are related to the Camera
14. Move the Camera functions into a new class
15. Create the Camera object in the Game class
16. Test that the Camera can still be created and rendered
17. Use the Camera class as a base class
18. Derive different types of Cameras from the Camera class
19. Test the Chas Camera that it follows the car
20. Test the Stationary Camera

Phase Five – Exia Engine

1. Create the base Physics Manager class
2. Create an instance of it in the Car Object
3. Ensure the Car object is derived from a Physics Object
4. Store all the relevant properties in the Physics Object
5. Call the physics manager update function in the car objects update function
6. Pass itself into the physics manager update function
7. Create the interface of an IExiaCommand, which has an execute function
8. Create the acceleration command
9. Instantiate this acceleration command in the physics manager
10. Test the acceleration command
11. Create the velocity command
12. Instantiate this velocity command in the physics manager
13. Test the velocity command
14. Create the displacement command
15. Instantiate this displacement command in the physics manager
16. Test the displacement command
17. Create the RPM command
18. Instantiate this RPM command in the physics manager
19. Test the RPM command
20. Create the Torque command
21. Instantiate this Torque command in the physics manager
22. Test the Torque command
23. Create the Force command
24. Create the Traction, Rolling, Drag and Brake force commands
25. Use the results from those commands in the Force Command
26. Instantiate this Force command in the physics manager
27. Test the Force command
28. Create the Weight Transfer command
29. Instantiate this Weight Transfer command in the physics manager
30. Test the Weight Transfer command
31. Create the Angular Orientation, Velocity, & Acceleration command
32. Instantiate the commands in the physics manager
33. Test the commands
34. Create the Lateral Force command
35. Instantiate this Lateral Force command in the physics manager
36. Test the Lateral Force command
37. Test all the physics dynamics are executed in the physics managers update
38. Use the results in the Car Object

Phase Six – Interface Information

1. Ensure the relevant information can be accessed from the Game class
2. Create a new sprite batch
3. Retrieve the velocity information and convert it to MPH
4. Use the speed and render it to the screen using the sprite batch
5. Retrieve the gear information
6. Use the gear information to render a image of a gear stick
7. Retrieve the RPM information
8. Use the RPM information to render a dynamic image
9. Test the information updates

Phase Seven – Camera Effects

1. Use the information from the weight transfer
2. Add camera effect to tilt backwards during acceleration
3. Add camera effect to tilt forwards during deceleration
4. Use the information from the lateral force
5. Add camera effect to tilt left during a right turn
6. Add camera effect to tilt right during a left turn
7. Test all motions of camera

Phase Eight – Input

1. Learn Direct Input
2. Use the tutorial from WGP to read input devices with Direct Input
3. Ensure the device can be read in XNA
4. Test the device and configure dead zones
5. Create the SWInput class to manage the initialization of the device
6. Create the SWInputState class to hold information of input states
7. Use the device and configure the states with the game
8. Test the game is able to use the input from the steering wheel
9. Test that the input is configure correctly

Phase Nine – Sound Effects

1. Search and download relevant sounds effects
2. Load sound effects into the XNA framework
3. Create the sound manager to store the sound effects
4. Design a function that can play the relevant sound when needed
5. Test that the sounds can be played
6. Use the information from the car object to determine what sounds can be played
7. Configure the RPMs with the engine sound
8. Configure the Gear change sound
9. Test the sounds are playing when state changes
10. Test the effects of volume and pitch of the sound, as the RPM change.