Graphics
Assignment 2

Deadline : 8:00 P.M. - 13-02-2019

Problem :

The previous assignment dealt with a 2D world. Now, we will move onto exploring 3D worlds. In this assignment, you will be trying to emulate a jet fighter plane game. The goal of your game is to obtain the maximum amount of points in the duration of your game. You are Link and you operate one of the finest fighter plane. The game can be seen from multiple views controlled by both mouse and keyboards. The requirements mentioned are minimal so please be creative and add interesting features.

3D World :

1. One of the important primary objects in the world will be the fighter airplane, which the player will be guiding through the land and sea obstacles located beneath to the final checkpost to mark the obstacle course “completed”. The plane will have the capability to hover(not move but stay in the air), rise upwards and go downwards, go forward, turn left and right, tilt left and right and then of course, shoot missiles and drop bombs.
2. Airplane dashboard. This will be displayed on the game screen in the bottom left corner. The dashboard must display the speed, current altitude and fuel remaining. This need not be textual but can be pictorial(like fuel bar or something). Zero fuel or zero altitude(hitting the ground) will result in crashing.
3. The obstacle course will be divided into multiple checkpoints. The plane will be guided towards a checkpoint with the help of a shiny red arrow, pointing to the checkpoint just beneath it. This arrow must be visible from far far away even if the actual checkpoint isn’t visible. Upon achieving that checkpoint the arrow will be

updated according to the succeeding checkpoint. This arrow will help the player navigate through the obstacle course.

4. Enemy cannons will be located on ground at each checkpoint attacking the player airplane. The fighter plane needs to destroy those enemy hideouts using missiles and bombs in order to mark that checkpoint completed and get the direction for next checkpoint.
5. Random rings of smoke in the air, passing the airplane through which will guarantee bonus points for the player.
6. Fuel ups on the way will guarantee that player has the opportunity to refill his tank for the long battle ahead.
7. Volcanoes upon the islands beneath, a complete NO-FLYING-ZONE for the player. The game will end as soon as the airplane comes too close to the proximity of a volcano.
8. Shooting enemy parachutes flying in the air is another way to earn some more points apart from the ones generated by the regular flying time. Missiles installed in the airplane could be used to shoot the flying targets ahead.

Refer : https://youtu.be/Am5t_dHqC9I

Controls :

1. Airplane will rise upwards(in hovering manner) with SPACEBAR. Releasing SPACEBAR will make gravity come into play pulling airplane downwards. With this one can control the altitude of the plane.
2. Use W to increase the speed of the airplane in forward direction. Buttons A and D will be used to tilt the airplane in left and right directions respectively.
3. Buttons Q and E will be used to rotate the plane counterclockwise and clockwise respectively.
4. MOUSE BUTTON 1(left click) will be used to release missile whereas MOUSE BUTTON 2(right click) will be used to drop the bombs.
5. Use other keys (mentioning them into README) to achieve Bonus 1.

Camera views:

Since now your objects have a third dimension to them as well, the following camera views have to be incorporated to give a more complete feel to the game:

• Plane view: This is a view from the plane’s position where only a part of the world in front is visible. In other words, in this view, we see what the plane sees, as if we were the plane.

• Top View: This is a top-down view, as if we were looking vertically downwards from a position
in the sky. This gives a clear picture of the path.

• Tower view: Here, the camera is sitting on a tower, to the side of the plane of playing, observing it at an angle.

• Follow-cam view: This is a view of the plane and the region in front of it from a location behind and above it, as if the camera is following the plane.

• Helicopter-cam view: Here, the camera is movable with the mouse in an intuitive manner. Clicking and dragging should change the look angle, the up vector should remain up always,and the scroll wheel will move the camera closer or farther away from the scene.up vector should remain up always, and the scroll wheel will move the camera closer or farther away from the scene.

Bonus:

1. Airplane controls combined to perform special manoeuvres like looping-the-loop or a barrel roll.
2. Dashboard of the plane has a functioning compass.
3. More interesting obstacles. And no! Just keeping coins hanging in midair without any purpose will not cut it :P

Grading :

You will be graded based on the correctness and efficiency (speed) of the implementation of the elements described above. Tentative grading would take place in several stages.
Version 1.0: A static world (ground, space, plane, dashboard) - 30 marks
Version 2.0: Movement of the plane, collisions, checkpoints, key controls, etc: - 30 marks
Version 3.0: Views, enemy, missiles, bombs, etc: - 30 marks
Version 3.5: Fuel. health, volcanoes, etc: - 10 marks
Version 4.0 (BONUS): Be creative. - 20 marks

Submission :

You submissions should include your source code and a makefile. Do not use any non-standard libraries. In addition to these, include a file named README.md in the submission that gives a one page description of the game and how to play it.
Details of how to submit and any modification to the above submission details will be posted by
the TAs towards the submission deadline. This assignment will take time to complete. Start early. All error scenarios must be gracefully handled (Games crashing during testing will be penalised).

Plagiarism in any form shall not be tolerated (MOSS will be used) and a straight F grade for the course will be given. Not protecting your code and collaborative efforts will also be counted as Plagiarism.