Middle East Technical University
Department of Computer Engineering

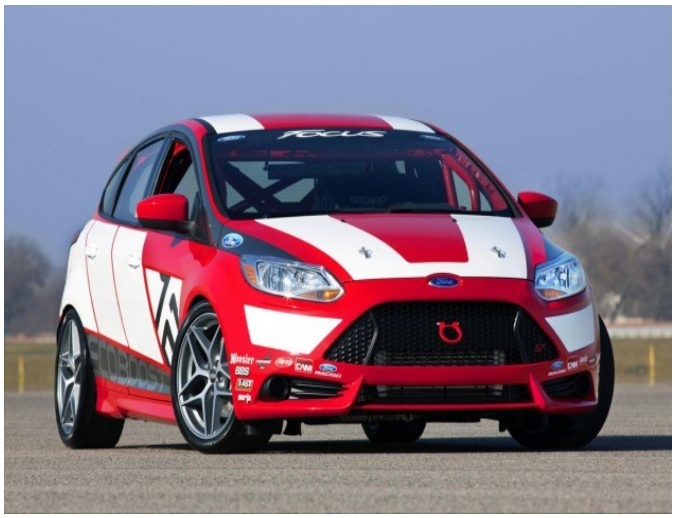
CENG 477

Introduction to Computer Graphics

Fall '2010-2011
Programming Assignment 2

CAR RACE IN 3D ENVIRONMENT

REGULATIONS



Due date: 19.12.2010 Sunday 23:59

Late Submission: Late submission is not allowed.

Submission: Use the COW system to submit your homework file. The homework should be done and submitted **individually**. You will submit a single tar.gz file named "hw2.tar.gz", which includes your source codes for the assignment and a makefile. Make sure that your code can be compiled by a single "make" command at the shell prompt. The makefile should produce an executable named "hw2" which can be executed directly by typing "./hw2" at the command line.

Evaluation: Your homework will be evaluated in a Linux environment. Before submission, you have to test your homework at one of the department's computers using Linux to have a common platform.

Cheating: In case of cheating, all parts involved (source(s) and receiver(s)) get zero.

Newsgroup: You must follow the newsgroup for discussions and possible updates on a daily basis.

MAIN THEME

In your second assignment, you are going to implement a simple race circuit with a race car that will be controlled by the user. There will be no other cars in the race, and no use of AI. The purpose of this assignment is to help you understand 3D graphics and geometric transformations in OpenGL.

SPECIFICATIONS

- In this assignment, you are going to implement a 3D environment which consists of a race car and some obstacles. You will use several geometric and Object3DS objects to implement the race car and obstacles.

- You are going to implement a number of camera modes that allow you to see the environment from different locations.

- You are expected to initialize OpenGL with your own implementation so no code for initialization is provided in this assignment except lighting initialization code.

- Lighting is not in the scope of this assignment. Therefore, the lighting initialization code is provided for you to use in the assignment.

Lighting initialization:

```
glEnable(GL_COLOR_MATERIAL); //enable colors
```

```
glEnable(GL_LIGHTING);        //enable lighting.
glEnable(GL_LIGHT0); //single point source of light.
GLfloat position[] = {-1000.0, 1000.0 , 500.0, 0.0}; //position of the light.
GLfloat blackColor[] = {0.1 , 0.1 , 0.1 , 1.0};          //ambient component.
GLfloat whiteColor[] = {1.0 , 1.0 , 1.0 , 1.0};          //diffuse&specular
component.
glLightfv(GL_LIGHT0, GL_POSITION , position);       //set position.
glLightfv(GL_LIGHT0 , GL_AMBIENT , blackColor);          //set ambient
light.
glLightfv(GL_LIGHT0 , GL_DIFFUSE , whiteColor);          //set diffuse
component.
 glLightfv(GL_LIGHT0 , GL_SPECULAR , whiteColor);         //set
specular component.
```

- There will be two screen modes. In the first screen mode, the screen displays a car object and there will be no environment box. The second screen displays the race environment and there will be an environment box.

- The environment box for the second screen mode limits your environment inside a box from x, y and z coordinate axes. You can simply think environment box as a rectangular box that you create your environment inside. You are free to define the size of your environment box depending on your needs but make sure that the environment box will be long enough to allow you to simulate car race atmosphere. The color of the polygons defining your environment box can be used as the sky color and floor color of the environment.

- You are going to create a car object using OpenGL or GLU functions. This car object will have 4 wheels and 6 more 3D geometric shapes (GLU functions or user defined OpenGL calls). Apart from these limits, you are free to design the car of your dreams by means of shape and color.

- Unlike the first assignment, the circuit will be straight. It will consist of 3 lanes. In the right and the left side of the circuit there will also be obstacles. Considering the obstacles in the circuit and outside the road there will be 5 lanes in the car race environment in total.

- Like the first assignment, the circuit will be divided into sections. Since the environment has five lanes, a section that contains five regions, each of which, may or may not contain an obstacle. The inner three lane (2., 3. and 4. column of the sections) will represent the circuit lanes. Similar to the first assignment, you will divide it into a number of sections based on the input file named “METU_F1_2010.txt”. It will only be tested with a circuit of length 1000.

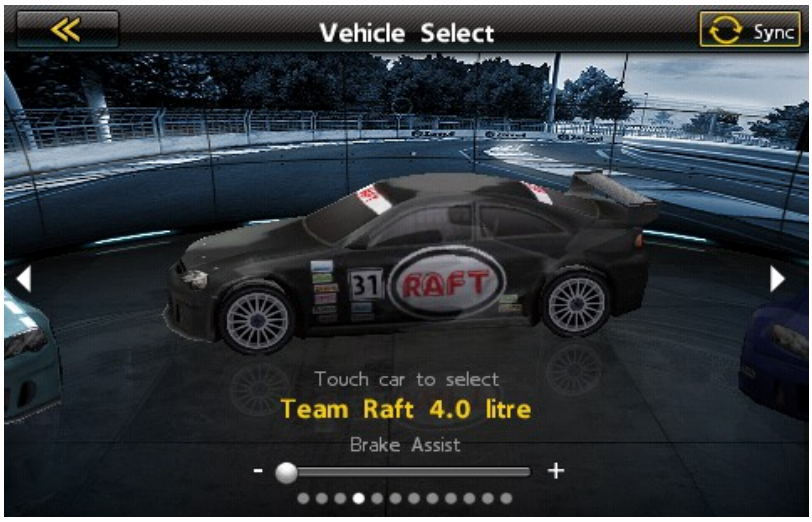
- There will be three types of obstacles: tree, wall and another car, which are represented in the input file as 'T', 'W' and 'C' characters respectively. These obstacles will be 3D objects. '_' (underscore) character will represent no obstacles in the input file.

GAME MODES

1- CAR SELECTION MODE

- When the program runs, it will first be in car selection mode. In this mode the program displays the car objects available under folder name 'cars' in the current directory.

- The Object3DS objects and the car you created by OpenGL functions will be displayed in the screen one by one. User will be able to change the car which is displayed on the screen by pressing left and right arrow keys. User can select displayed car by pressing 'Enter' button. After selection, the game will enter the race mode.



- Car selection mode will be similar to the above picture but it will be more simpler. There will be no background image or car properties. The car will be displayed in empty space. In order to visualize the car in different angles, it will be turning around itself in constant angular speed.

2- RACE MODE

- After selecting a car in selection mode, the game will go into the race mode. The race mode start with the car selected in the car selection mode.

- The car will start the race in the middle lane and parallel to road. It will be near to back surface of the environment box so that in front of the car, there will be more spaces compared to the back of the car. The car will move towards to the obstacles and it cannot go backward.

- The car will start the race with zero speed. The control buttons for the car is as follows:

Up arrow key: Increase speed

Down arrow key: Decrease speed

Left arrow key: Turn left

Right arrow key: Turn right

- Speed of the car will decrease slowly when the up arrow key is not pressed. To be more realistic, when the left or right button is pressed the car does not change lane immediately, it will simply turn left or right normally.

- Only the car will move in the car race environment. The other objects (obstacles, road, road lines etc...) will stay in their original positions. The car will travel with the key directives above (turn left, turn right, increase speed and decrease speed) in the environment.

- The car will stay in environment box borders. Speed of the car will decrease with a constant ratio when it goes out of the road.

- The road will be parallel to the opposite environment box edges and there will be discrete road lines on the road.

- All of the obstacles will be read from the input file and rendered when the race starts.

- Tree objects will be created by drawing a cylinder and a sphere over it. Walls will be created by drawing an rectangular prism. Car obstacles will be created by the one you will create for car selection mode or loading car object.

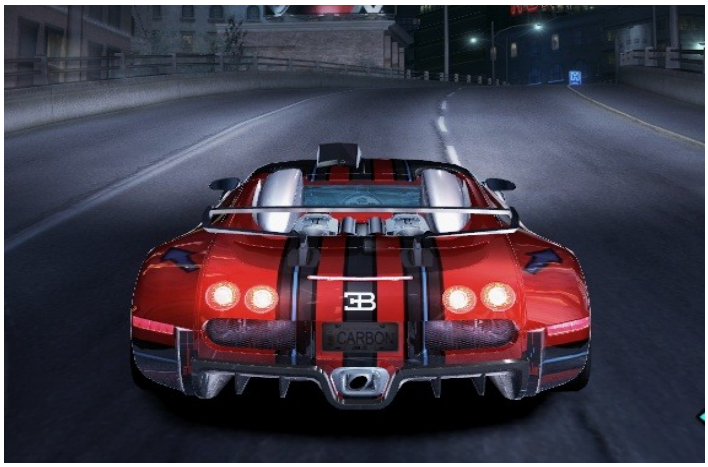
- There will be collision detection for the obstacles. When the car hits an obstacle, it cannot move to the axis which the obstacles stands. If the car has a speed on the other axis it will continue to move on that axis with the same speed.

- Between two obstacles, there will be at least one car distance.

Camera Modes

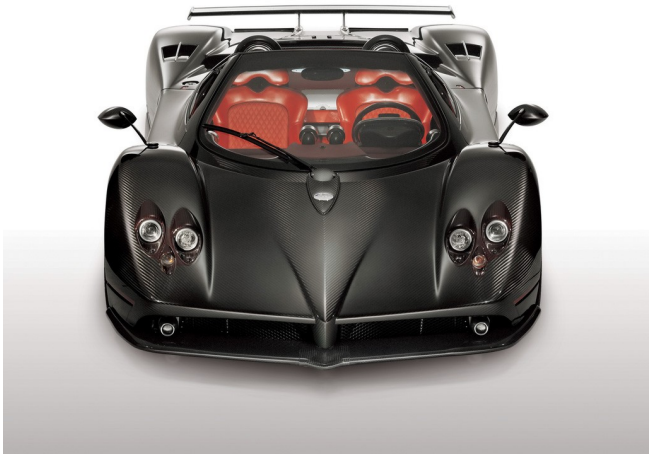
- For race mode there will be three camera modes for observing race environment.

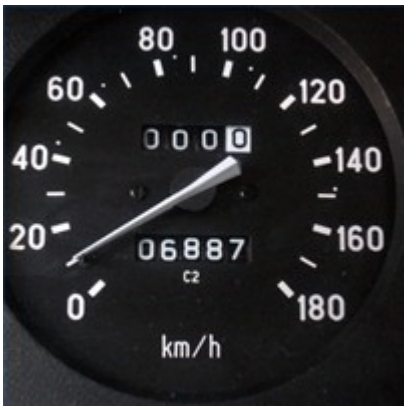
- In the first camera mode, the camera will be back of the car and follow it. The direction of the camera will be the same with the car but it will look to the ground with 30 – 45 degree angle. The distance between camera and the car gets larger when the speed increases and vice versa. When the program first enters car race mode it will be in this camera mode. The camera will go into this mode by pressing “1” key.



- In the second camera mode, the camera will be in front of the car windshield in the same height with height of the car. Its direction will be exactly the same with the car. The camera will go into this mode by pressing “2” key.

- In the third camera mode, the camera will be over the car. It will show the car from front side. The direction of the camera will be same with the obstacles move and look to the ground with 30 – 45 degree angle. The horizontal position of the camera will change by car move but its direction will not be changed. The height of the camera will remain unchanged. The camera will go into this mode by pressing “3” key.





- At the right bottom of the window, there will be a subwindow which contains a speedometer similar to the picture above. There will be a triangle and dots around this triangle. The triangle will turn according to the current speed. It is not necessary to write speed values on these dots, triangle and dots will be enough to draw speedometer. The speedometer will be in all three camera modes in car race mode.

OBJECT3DS OBJECTS

- In order to add some realism to your environment, you will put ".3DS" car objects into your environment as a part of your implementation. We will provide you some 3DS car objects which you can use.

- While locating objects, pay attention to the size and vertical position of your objects.

- For loading 3DS objects, you will use the code that we provide you with the assignment.