Project 1

1 Overall Idea

To build a racing game with third person view from the back. The vehicle to be used will be a dirt-bike (default .obj version will be provided; you are free to use any other). The terrain will have to be developed indigenously using texture mapping. A fully functional GUI (Keyboard+ Mouse) will also have to be developed.

2 Phase I

To develop fully functional libraries for the following utilities

2.1 Terrain Renderer

A separate library which will generate terrain with a defined course/track for the race. The terrain must be uneven with small hills & valleys. The objective is to make the vehicle jump of hills and land properly on the terrain. You are free to introduce any obstacles if you like, such as trees, wooden-crates, water etc. as per difficulty level.

2.2 Texture Mapper

A separate library to map textures onto the terrain generated. Textures can vary from sand, grass, road, snow, etc. this library will handle the parsing, loading and mapping of all textures used in the game (not necessarily the terrain only).

2.3 Object Renderer

The 3D models will be provided in form of .obj files, which is a standard text based format of representing 3D objects. This library will have the necessary data structures for storage, manipulation and display of 3D models.

2.4 Physics Engine

This library will deal with all interactions between 3D models and the terrain. Also it will model the trajectory of the dirt bike, while it is in air. It will also handle collisions between world-objects and the bike. Such as bonus markers, obstacles etc.

2.5 Game HUD & Menu GUI

This library will handle all parts of the GUI and the in game HUD displaying time & score at least.

3 Phase II

Use the libraries developed above to build a fully featured application.

4 Phase III

Test the developed application and fix bugs which are discovered. Make a documentation of all the code using tools such as doxygen. Write up a manual for the game using latex.

5 Detail Specifications & suggestions

5.1 Single player

The developed game will be a single player game. Only one player will play the game at a particular time. An overall table will be maintained for every track keeping the best 5 times with the player′s name.

5.2 Track length

At least 10 different tracks will have to be created. the configuration of the track once developed must be stored in some format so that it can be replicated every time the track is used in the game. The tracks will have to be of fixed length. Practice tracks can be infinite (if you are able to generate terrain randomly at run-time). A description of the track will have to be given in the menu. In the description previous best-time, best-score and the length have to be provided. Added info can be provided as per your choice.

5.3 Bonus Markers & obstacles

Bonus markers & obstacles will be evenly distributed along the length of the track. Going through a bonus marker will give additional points to the player. Hitting an obstacle will slow down the pace and also result in loss of points.

5.4 Mandatory menu items (develop using Qt)

5.4.1 New Game

Selection of difficulty: Easy or hard (section of hard will result in increased no of obstacles)-> Selection of Track: At first only the first track can be selected. As the game progresses, more tracks will be available for selection.

5.4.2 Practice Mode

Same as above but there will be no timing or scoring.

5.4.3 High score viewer

This menu will display all the best-times & scores for all the tracks.

5.4.4 Exit button

To close the window of the game.
You can add more functionality as you want.

5.5 Scoring

Progression along the track will result in constant accumulation of points. Taking bonus markers, jumping higher & longer will add to the score. Hitting obstacles will result in loss of points.

6 Programming Environment and allowed libraries

1. For rendering interface (MENU) it is mandatory to use **Qt** (used to develop user interfaces for applications).
2. **Eclipse IDE** will be used in development of the game.
3. **OpenGL** libraries will be used in overall development.
4. Language: C /C++
5. **Valgrind** tool can be used for checking memory hazards etc.
6. Use **Github** for maintaining versions of the project while you are working over it.
7. **libpng** can be used for loading and parsing of textures.
8. Tools to read .obj files will be provided once you start this Project.

7 Suggestive video link

In the video you see there is no well-defined track as such, you′ll have to provide them. Instead of a cycle we will use a dirt bike. Animation of the player + bike is as of now optional.

Link to video is : /home/mtech/mcs122844/CSP301/

8 Models

The 3D models of the player and the bike will be provided.
Link : /home/mtech/mcs122844/CSP301/