

OBJECTIVES

The main objective of this game is to keep the player as entertained as much as possible and to make the game attractive by making its UI great. The following points can be summarized as the basic objectives of this project:

- To develop a racing game where the player reaches the finish line without colliding a single time wins.
- To continuously change ta position of the track and car as the race progresses.
- To stop the car if it collides with the obstacles.
- To make proper and efficient use of SFML, a graphics library so as to make the game enjoyable.
- To explore the features of C++ language and familiar with resource reusability by making user defined header files.

INTRODUCTION

Race car is a simple game in which there is a car which has to get to the finish line without colliding to any of the obstacles encountered while in the track. The game terminates as soon as the car reaches the finish point.

We will be doing this project by using Simple and Fast Multimedia Library(SFML), a C++ graphics library and OpenGL whenever required. SFML is a portable and easy-to-use API for multimedia programming. It can be used as a graphics resource and framework to support OpenGL. Though opengl is not used we have made a function for pseudo 3D effect.

APPLICATION

The application of this kind of project work is very huge. A significant part of the software industry is occupied by computer games in the world, and its share is not going to decrease. More and more people are having access to game playing devices day by day. They like to play games for entertainment when they are free from their work. Hence there is always an enough demand for good computer games.

From the students point view, the knowledge gained in software design techniques, computer graphics, sounds and all the multimedia stuffs are definitely going to be useful in their future career.

LITERATURE SURVEY

Most of our program development deals with the Object-Oriented Programming (OOP) concept; hence, different books were studied for the OOP concept. The book “Secrets of the object-oriented programming in C++” by Daya Sagar Baral and Diwakar Baral was used as reference for the development of program that assisted us to make our program development easier. For the SFML, different e-books were referred and different video tutorials from the website were viewed.

EXISTING SYSTEM

There are more than hundreds of clone games of this type and still more and more type of such games are being created using different programming languages. The first of this genre was released in the mid-late 1970s by Atari and many others launched the clone game of this after the development of this game in 1975 for the first time. Till now this type of arcade game is in further development phase. Some of the best and well known existing car racing game are Need For Speed, Asphalt, Dirt Rally etc.

METHODOLOGY

Since we are creating the game using SFML in c++ platform, there were several steps we took (and are taking) for the completion of this game.

First process consists of installation of SFML into our device according to the version. Then, we had to include three aspects of SFML into the linker:

- 1) **Sfml-windows:** This module has to be used simply for rendering our window. That is, it helps to create a new window for the purpose of displaying the output. Here we can manage it into several aspects like the size of our window, presence of toolbars, possibility of the close bar button and also to enable the users to resize according to their wish.
- 2) **Sfml-graphics:** Since our objective is game, it certainly requires graphics. Graphics is our most important tool to make this project. There are several aspects of using graphics and till this date we have used two elements: a) Shape (a rectangle one): Mainly to enter a texture onto a rectangle.
b) Texture: Because if there is a texture then we can modify it to make animations.
- 3) **Sfml-system:** This module is especially used in different events, like taking inputs and reacting likewise. This module comes real handy when we have to focus on the time aspect while we are making our animation. Similarly, it is used while opening, closing or pausing the game.

As in the end , we followed the same process of methodology

The programming methods we've used can as summarized below:

- Analyzing the concept that can be used to develop proper program.
- Discussion on the topic that might be faced onwards.
- Making the project schedule.
- Initial coding for creating logic.
- Coding the program.
- Execution and testing the program.
- Debugging.
- Program Documentation.

Then comes the coding part, compilation and modification which is discussed in following topics.

Description

The game can be started by selecting the new game option. After the race starts the car can be controlled by pressing different letters in the keyboard. The players can move the car on both sides and can even go backwards for few blocks. The track also keeps moving as the race progresses. If any obstacle come along the way of the car, the car cannot go from that way and has to go from the right or the left side. If car reaches the finish line without colliding player wins the race. After the car reaches the finish line, the game terminates and the time taken is recorded as the score. The high score is also recorded if it is so. This can also be played single player where the player tries to avoid the obstacles and reach to the finish line in minimum time.

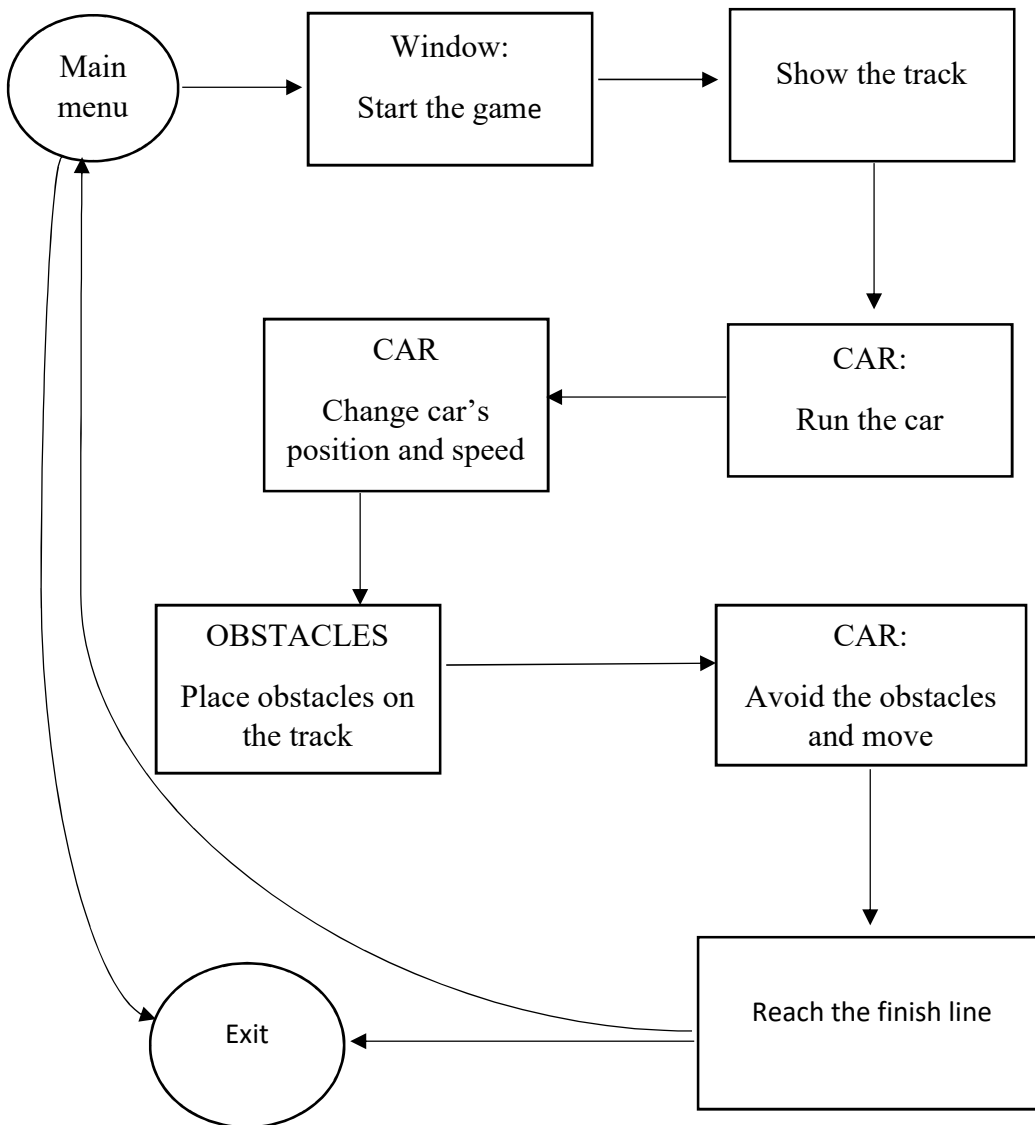
Implementation

There are different components which are to be created while programming in order to execute this game. The game will run as per the behavior of those components which are given below:

1. Track: Track is the place where the car will run and there will be different obstacles on different positions of the track.
2. Car: Car is the main component of this game which runs on the track and its position can be changed by pressing keys on the keyboard.
3. Obstacles: Obstacles come along the path of the car. The obstacles may be trees, blocks, walls, etc. which blocks the passage of the car.

4. Window: This component creates the window and starts all the required components for the gameplay. It is also responsible for the closing of the game.

Block Diagram



RESULTS

With the completion of the project, the objective of the project was achieved and the game was designed and completed as per the requirement. The complete game was designed with the necessary features.

The main objective for the development of this game was fulfilled, but still some features in this game could not be completed which might be due to the time constrain, and also, this is the learning phase due to which much time was spent on learning the subject matter as the topic was completely unknown before the starting of project.

From this project, we can take positives as we achieved the co-operation between the team members, learnt to work in a team, backing up each other in need and also the challenges to develop the software by developing this small gaming program.

A glimpse of project:



Figure 1: main menu



Figure 2:track



Figure 3 a portion of track with object

PROBLEM FACED AND SOLUTIONS

This project was almost completed on time with lots of effort. Although the project was completed on time, many problems and errors were faced during the development of the game and the solutions to problems were obtained with the discussion with friends and teachers and the deep study of the codes finally made the error free game. The major problems that occurred during the development of program are listed as follows:

1. Adding graphic library SFML in code blocks.
2. We had great difficulties in developing proper logic due to lack of proper reference materials.
3. Problems in setting the physics of the game.
4. We faced a huge difficulty in putting the object (car) on the track and move it like in real life race.

These major problems were solved with the help of our friends, practice and observations made as per the change brought in the development phase of the program. We had to revise some of the physics formula for smooth movement of the car and further we consulted many books and tutorial to solve all minor and major problems faced during the development of program.

LIMITATION AND FUTURE ENHANCEMENT

As we've completed the game for entertainment, many more features can still be added to make the game more attractive and more popular among the user so as to make it popular among this competitive market. The following features can be added to our project:

1. Due to limited time frame we haven't added AI in our program so AI can be added and a new race mode can be introduced in this program.
2. It reads the selected inputs of keyboard only. Joysticks input is not read by this program which can be added for the enhancement of this program.
3. No traffic rules are taken under consideration. So, taking it into a factor we can add this feature and via this game children can learn the traffic rules.
4. We have just given pseudo 3D effect to our game but it can be further developed into a full-fledged 3D car race game with different modes.
5. The facility of saving the game is not possible, which could be added in future.
6. High score that is shortest time to reach the end point is also not shown which can be added.

DISCUSSION AND CONCLUSION

This project was the great way of learning the object-oriented programming. The object-oriented programming concept was implemented for the completion of this project. Besides, the graphics library SFML was used to make the project look attractive and efficient.

Before the starting of the coding of the project, different steps have been followed for the completion. First of all, the study, research and analysis of the topic of the project were done where different information about the topic was collected and then proposal for the topic was created documenting the information gathered. The designing of the layout of the project, which can be regarded as the framework for the project was thought and kept in mind before starting the coding part. The program was continuously tested throughout the development time and modified as per the need to make the project look attractive and user friendly.

Hence, from this project, we learnt many things required for the good programming practice and also, how to work in a team coordinating each other.

REFERENCES

Working with C++ alone without any extra references would have been difficult. We too have gone through various books and sites related to C++ programming and game programming. We are thankful to the author of those books and sites. The following books and sites have provided us the knowledge for the completion of our project,

- The Secrets of Object Oriented Programming in C++ - Daya Sagar Baral , Diwakar Baral
- Sfm1.dev.org
- <https://sonarsystems.co.uk/>
- https://en.wikipedia.org/wiki/Racing_video_game