

Highway Patrol

Kamal Bhatt, Lalit Pant, Akul Joshi

CSE Department of Galgotias University

Galgotias University

Greater Noida, Uttar Pradesh, India.

ABSTRACT

HIGHWAY PATROL CAR GAME

This research paper describes a case study, where we are focusing on developing a 3D racer game, using process based upon agile development: an evolutionary development method.

The project will cover implementation of real-time graphics, physics engine network support including sound effects and background music with catching sounds.

In the End our case study will show that this development process was an appropriate choice for our game development project. the objective of the is to provide the player (user) with a challenging and enjoyable experience.

Keywords- network support graphics

INTRODUCTION

Unity is a complete platform for 2D. It enables us to deploy to all or any the main and emerging mobile operating systems, speed up our development process, optimize our game, connect with an audience, and achieve

commercial success.

Unity may be a complete platform for building beautiful and engaging 3D, and 2D, games. In fact, more 2D games are made with Unity than with the

other game technology, and companies such as Disney, Electronic Arts, LEGO, Microsoft, NASA, Nickelodeon, Square Enix, Ubisoft, Obsidian, Insomniac, and Warner Bros rely on Unity tools and features for development purpose.

- This project is made using Unity game engine and C#

as scripting language.

- This project is game named “HIGHWAY PATROL”.

- This game is build keeping in mind the age group of

10-14-year-old children.

- It is simple UI is very easy to navigate through.

This game is based on obstacle dodging. A police car

while patrolling encounters an emergency situation in

wrong side of the highway and decides make a run in the

wrong lane of the highway. Now the user has to dodge

the upcoming cars and make the police car reach its

destination.

LITERATURE REVIEW

We started our research considering making a 2D game for children of age group 10-14. So, we started questioning and interviewing children of the same age group about which type of games they would like to play in their free time.

We have gone through all the points and planned to make a 2D offline obstacle dodging game with easy to understand UI and controls

BASIC IDEA OF GAME UI

Starting from the authentication scene, it will allow user to:

- I) Sign Up
- II) Sign In
- III) Reset Password
- IV) Exit

After successful sign in it will redirect to the main menu, it

will have 6 major buttons

- Play
- Instruction
- Controller settings
- Leaderboard
- Music mute
- Logout

Main menu will have background music to make it more attractive which can be pause and played by the user according to their wish. Play button will change the scene to

level 1 of the game and after achieving certain score it will be advanced to level 2 which will be quite difficult.

PROBLEM STATEMENT

The following problems may arise while developing or

using the existing car game:

To manage the control of the car

It simply refers to managing the controls, movement, and

other settings of the car game.

To manage the full UI of the gameplay.

The user interface must work in accordance with the user and the game for smooth functioning and experience and also to manage the updating of score

in gameplay and also in database

FUTURE SCOPE

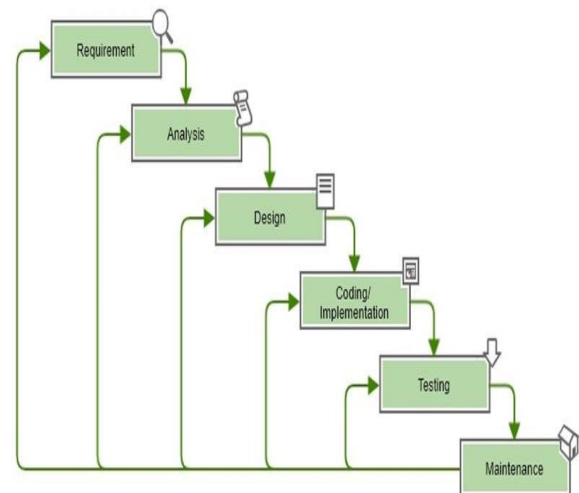
This game can be further developed by adding more levels, maps, options to select from multiple cars, adding more soundtracks and vehicle sound.

The game can be further optimized. Graphics can be increased by using high pixelated sprites.

Last but not the least it can be converted to a 3D Game.

METHODOLOGY

Software development life cycle (SDLC) This project follows the iterative waterfall model. The iterative waterfall model provides feedback paths from every phase to its preceding phases



When errors are detected at a later phase, these feedback paths allow correcting errors committed by programmers during some phase. The feedback paths allow the phase to be reworked during which errors are committed and these changes are reflected within the later phases.

REFERENCES

*Unity official documentation

<https://docs.unity3d.com/Manual/index.html>

*Firebase documentation for real-time database

<https://firebase.google.com/docs>

*Unity Rest API documentation

<https://docs.unity3d.com/Manual/UnityCloudBuildRestAPI.html>

*Firebase documentation for real-time

database<https://firebase.google.com/docs>