

Development of Premier Race India (PRI) an Indian Themed Car Racing Game using Unreal Engine 5

Prof. Chandrapal Chauhan¹, Aniket Tandekar², Manav Patil³, Vijaya Adamane⁴, Aastha Petkar⁵, Anuja Panchariya⁶

¹Professor, Department of Computer Science and Engineering, Government College of Engineering, Chandrapur, Maharashtra, India.

²⁻⁶Student, Computer Science and Engineering, Government College of Engineering, Chandrapur, Maharashtra, India.

Abstract - This paper navigates the development odyssey of "Premier Race India" (PRI), a groundbreaking Indian-themed computer car racing game expertly crafted using blueprints and 3D assets the sophisticated features of Unreal Engine 5 (UE5). The primary objective was applying acquired concepts, with a specific focus on creating an immersive and culturally resonant car racing experience set of India. The methodology spans scripting, UI/UX design, level creation infused with Indian themes, creating 3D models reflecting Indian aesthetics, third-party asset integration inspired by Indian aesthetics, logic and functionality establishment tailored to Indian contexts, model integration reflecting Indian landscapes, and critical phases of testing, debugging, and deployment. Each stage is dissected, offering insights into decision-making processes, challenges faced, and innovative solutions employed throughout the development lifecycle, all driven by the overarching goal of capturing the essence of Indian culture in a digital gaming experience. The AI Cars are created which are able to perform path-finding and self-learning overtaking aggression. The discussion on the gameplay framework provides a nuanced understanding of core systems, emphasizing the utilization of cameras, components and the interplay between pivotal classes such as Actors, Controllers, and Pawns, all tailored to embody the spirit of Indian car racing. This elucidation serves as a structural foundation for PRI, showcasing the team's adept navigation of complexities in creating a seamless and dynamic gaming experience with distinctly Indian flavors. In conclusion, "Premier Race India" transcends traditional gaming norms, standing as an extraordinary endeavor that pushes the boundaries of gaming innovation, capturing the essence of Indian culture within the dynamic framework of UE5.

Key Words: Game Development, Car Racing Game, AI Car Model, Unreal Engine 5, Blueprints Visual Scripting, Assets system, 3D Modeling, Blender, UI/UX, Level Engineering, Indian Themed, Video Games.

1. INTRODUCTION

The term 'Game Development' is used to refer to a broad set of algorithms that include the process of creating video

games, encompassing everything from the initial conception of an idea to the final release and ongoing maintenance of the game. It includes implementing gameplay mechanics, designing algorithms for artificial intelligence, creating graphical user interfaces, handling physics and collision detection, managing game state, and optimizing performance for a seamless player experience. It is used to create interactive entertainment, educational simulations, and virtual training experiences. Its applications extend to diverse platforms, including mobile devices, consoles, PCs, and emerging technologies like virtual reality and augmented reality. The field combines artistic creativity with technical skills to produce engaging and immersive digital experiences. The project utilizes advanced game development techniques, prominently featuring Unreal Engine 5, blueprints, and third-party assets to craft an immersive racing game that serves as a testament to the team's technical prowess and creative acumen. Leveraging Unreal Engine 5's state-of-the-art features, such as the Nanite virtualized geometry system and Lumen Global Illumination, the game achieved stunning graphics and realistic lighting effects. The integration of Blueprints facilitated the rapid prototyping of intricate game mechanics, including vehicle controls, collision detection, and event-driven systems for power-ups and obstacles. Carefully selected third-party assets enhanced the visual appeal, incorporating detailed vehicle models, animations, and diverse environmental elements. Through rigorous playtesting, debugging, and optimization using Unreal Engine's profiling tools, the team delivered a showcase of technical innovation and creative excellence in the realm of video game development.

2. LITERATURE REVIEW

R. Kothari, S. Nawar, S. Kothari³, Asst. Prof. J. Jeswani (April 2021), "Game Development using Artificial Intelligence in Unreal Engine". IRJET

Excerpt: To develop a sandbox environment where the player character completes a specific set of objectives to proceed the main goal. The game will simulate combat from a thirdperson perspective. The player will encounter opponent waves they need to survive in order to progress.

The opponents will be predictable AI which the player can predict in order to overcome and defeat them. The proposed work can be divided into ten modules.

Marvin T. Chan, Christine W. Chan, and Craig Gelowitz. (2015). Development of a Car Racing Simulator Game Using Artificial Intelligence Techniques. International Journal of Computer Games Technology

Excerpt: This paper presents a car racing simulator game called Racer, in which the human player races a car against three game-controlled cars in a three-dimensional environment. The objective of the game is not to defeat the human player, but to provide the player with a challenging and enjoyable experience. To ensure that this objective can be accomplished, the game incorporates artificial intelligence (AI) techniques, which enable the cars to be controlled in a manner that mimics natural driving. The paper provides a brief history of AI techniques in games, presents the use of AI techniques in contemporary video games, and discusses the AI techniques that were implemented in the development of Racer. A comparison of the AI techniques implemented in the Unity platform with traditional AI search techniques is also included in the discussion.

David Michalík *, Miroslav Jirgl, Jakub Arm and Petr Fiedler. (2021). Developing an Unreal Engine 4-Based Vehicle Driving Simulator Applicable in Driver Behavior Analysis—A Technical Perspective. Safety.

Excerpt: Vehicle safety remains a topic of major interest, and diverse assistance systems are implemented that focus primarily on analyzing the immediate vicinity of the car and the driver's control inputs. In this paper, by contrast, we emphasize understanding the driver's control performance via obtaining valuable data and relevant characteristics. To acquire the data, we employed an inhouse-designed, laboratory-built vehicle driving simulator. This simulator exploits the Unreal Engine 4 framework to deliver a high level of realism.

Michael, Jeanny Pragantha, Darius Andana Haris. (2020). Android Arcade-Style Racing Game "CarsTime" With Antigravitation Theme. IOP Conference Series: Materials Science and Engineering.

Excerpt: CarsTime is an arcade-style racing game with antigravitation theme. This game is made using Unity3D game engine with C# programming language. This game is an Android game. In this game, player will race against bots on track that is not bound by gravity so each car can move even on upside down track. There are three cars that can be used by players, namely C.car, Sonic and Magnum. There are three stages that player can choose to play, namely Map Aspal which doesn't have any additional difficulties, Map Pasir which makes player's car move slower and Map Salju which makes player's car slippery. There are three items that player can collect or interact while racing, namely coin which player can collect for buying car, boost can which player can collect to fill boost meter which can be used

when full to increase player's speed for a certain time, and boost pad which will increase player's and bots speed for certain time when touched. This game is tested by using three methods which are blackbox testing, alpha testing and beta testing. Based on the testing result, it can be concluded that "CarsTime" has an interesting racing gameplay with hard difficulty on Map Salju and easy difficulty on Map Aspal and Map Pasir.

3. PROPOSED WORK

To develop "PRI: Premier Race India" a low poly arcade Indian Themed Car Racing Game using Visual Scripting, Unreal Engine 5 where user where the player character completes a set of objectives by following a set of rules to achieve main goal in given amount of time. The player will compete with other AI opponents to win the race. The proposed work can be achieved using following methodology.

3.1 Scripting

A scripting language or script language is a programming language that is used to manipulate, customize, and automate the facilities of an existing system. It is usually interpreted at runtime rather than compiled. Scripting languages are also referred to as very high-level programming languages, as they operate at a high level of abstraction, or as control languages, particularly for job control languages on mainframes. In this context, the term script refers to a small program in such a language; typically, contained in a single file, and no larger than a few thousand lines of code.

3.1.1 Setting up a Controller and Pawn

The first step is setting up a controller and pawn. A pawn represents the player or AI-controlled character in the game. To set up a pawn we created a Blueprint Class with a base class Pawn. Inside its blueprint we added components to the pawn, such as a Skeletal Mesh for the visual representation of the character, collision components for interaction with the environment.

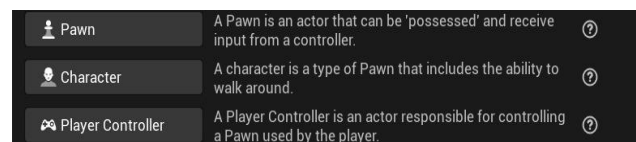


Figure 1: Setting up a Controller and Pawn

Then, added a CameraComponent to the Blueprint of actor that is playerPawn and attached it to the actor to define the camera's position and orientation. It is fixed so if player turns then camera will move accordingly.

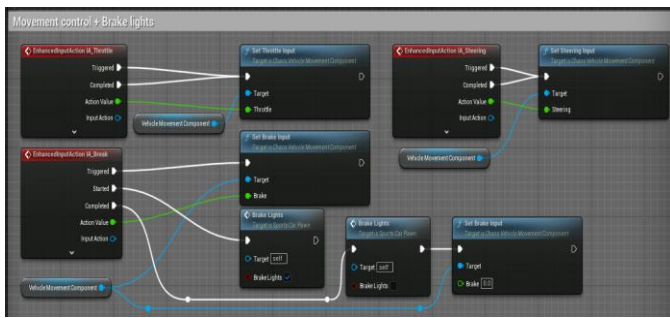


Figure 2: Player Car Movement Blueprint

A controller is responsible for controlling a pawn or group of pawns. To set up a controller we created a new Blueprint Class: PlayerController. Inside the Blueprint editor, we implemented input handling logic to control the pawn's movement, camera, and other actions. It is used for input events like "Axis Move" or "Axis Turn" to handle player inputs.

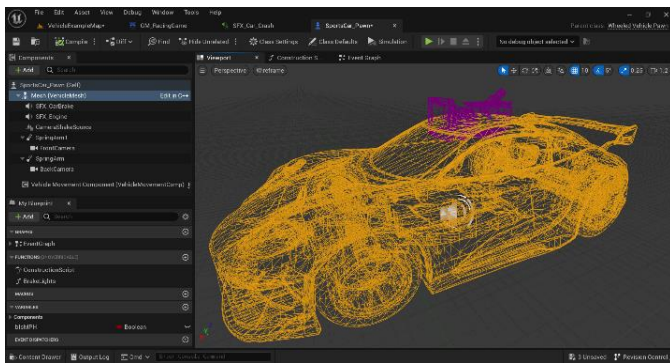


Figure 3: Player Car Skeleton

3.1.2 Creating AI Car Model

It starts with creating a new Blueprint Class for the AI car. It includes setting up acceleration, steering, braking, and handling for the AI cars, defining checkpoints and paths for it to follow, and implementing logic for decision-making.



Figure 4: AI Cars with path prediction and self-learning

AI cars to act as competitors for the player by adjusting parameters such as speed, aggression, and skill level to manage overtaking and precision. AI perception is used to

detect nearby obstacles, other vehicles, and track boundaries. These cars learn the path with each lap and gets better than previous.

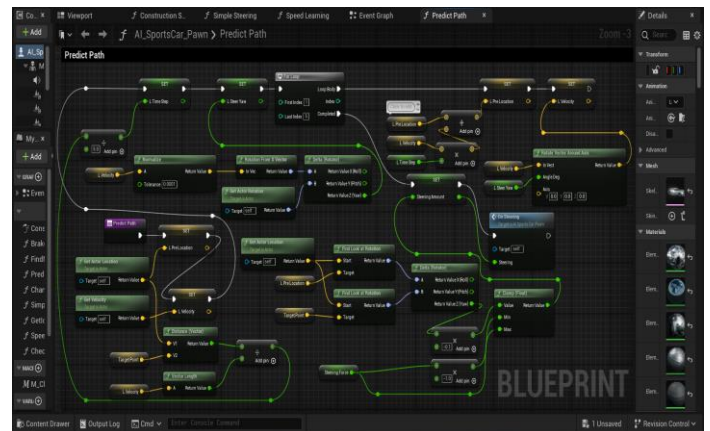


Figure 5: Path prediction script

Sample paragraph Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper.

3.2 Third Party Assets

Third-party assets, which are sourced from outside developers or companies and integrated into the game to enhance its visual appeal, include 3D models, textures, car models and race tracks, immersive sound effects and music tracks, or sophisticated code snippets for gameplay mechanics. We expedite the development process to provide players with a polished and captivating gaming experience by utilizing these third-party assets. We used 3D models of trees, rocks, cliffs, billboards, and street maintenance objects like traffic cones and street lights imported using Quixel-Bridge a Unreal Engine plugin.

From Unreal Engine Marketplace used following assets

1. Vehicle Variety Pack
2. City Sample Vehicle
3. Modular Building Set
4. City Sample Crowds
5. Quixel Bridge 3D Models

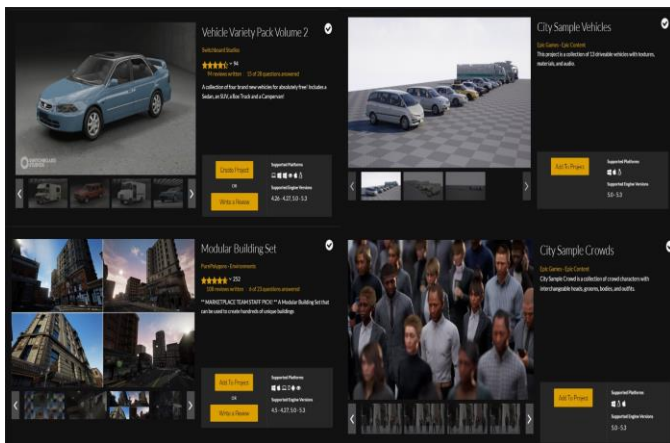


Figure 6: Assets from Unreal Marketplace

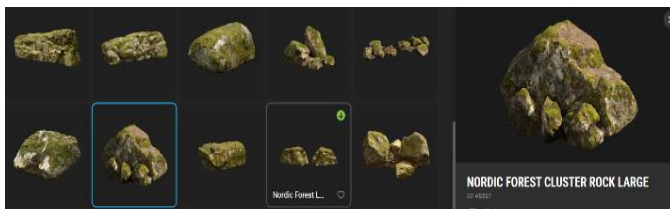


Figure 7: 3D Rocks from Quixel Bridge

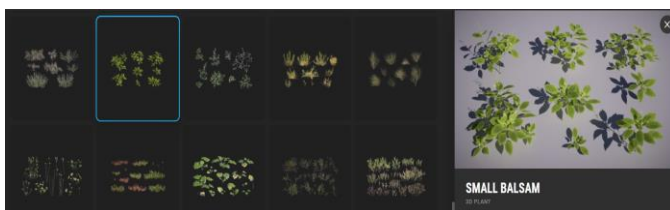


Figure 8: 3D Plants and Trees from Quixel-Bridge

3.3 UI/UX Engineering

User interface (UI) engineering is the design of user interfaces for the game, with the focus on maximizing usability and the user experience. It is a paramount in ensuring players are fully immersed in the thrilling and fast-paced gameplay. UX encompasses every aspect of the player's interaction with the game, from the moment they launch it to the exhilarating moments of crossing the finish line. The handling and responsiveness of the car controls significantly impact the UX. Precise and intuitive controls allow players to feel connected to their vehicles

3.3.1 In-Game Menus

We designed In-Game Menus to serve as accessible interfaces, enabling players to adjust settings and manage gameplay seamlessly without interrupting the race. These menus were crafted to include options such as pausing the game to fine-tune settings such as graphics and controls, facilitating actions like restarting or quitting races, and

seamlessly resuming gameplay. Additionally, players were provided to fine-tune race parameters, including adjusting difficulty levels and weather conditions, with the aim of enhancing their overall racing experience.



Figure 7: PRI main menu

3.3.2 HUD and Score Progression System

The HUD (Heads-Up Display) are featured to a comprehensive array of essential elements to provide players with real-time information during races. A speedometer was prominently displayed, indicating the vehicle's current speed and lap counter tracked progress throughout the race.

The Score and Progression system was implemented to track players' performance and advancement throughout the gameplay experience. A scoreboard was created to display and time remaining.



Figure 8: Time remaining counter



Figure 9: Speed and gear

3.4. Level Engineering

We created Indian-themed levels for our game, carefully designed a vibrant and culturally rich environment inspired by India's diverse landscapes and architecture. The track's individual sections were thoughtfully chosen to capture the distinct allure and splendor of India.

In our game we created following level maps:

1. Delhi Map
2. Rajasthan Map
3. Kashmir Map

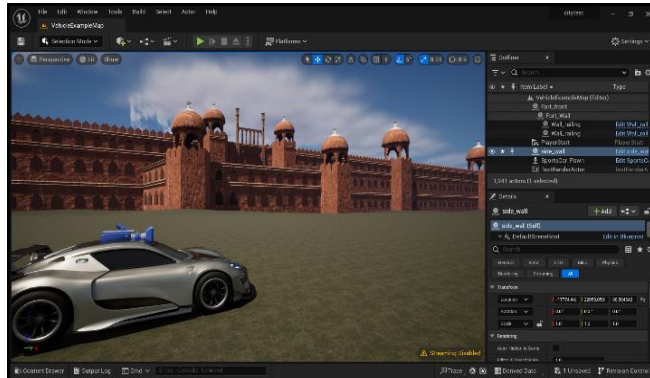


Figure 10: Delhi Map

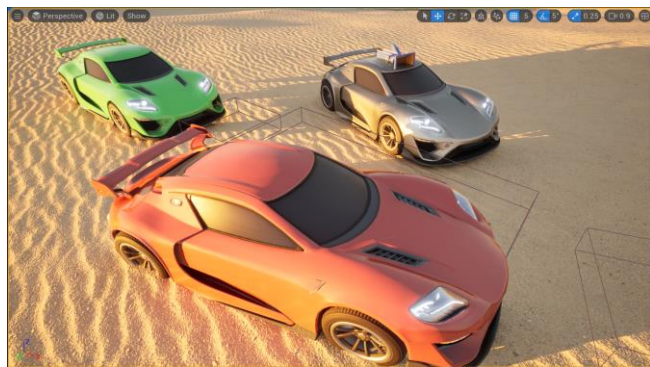


Figure 11: Rajasthan Map



Figure 12: Kashmir Map

3.4.1 3D Model Creating

Created 3D model of Indian architecture reminiscent of the iconic Red Fort (for Delhi Map) using Blender and Unreal Engine 5. Initially, we gathered reference images of the Red Fort from various angles to understand its intricate details using Google Earth. Then we set the scenes in Blender, we ensured proper scales and units.

Starting with a simple cube, we sculpted the basic shapes of the Red Fort, employing Blender's versatile modeling tools such as extrusion, scaling, and rotation. Adding architectural elements like arches, domes, and windows, we focused on maintaining accurate proportions and intricate details to enhance realism. Other 3D models were also created like Road, Traffic Maintenance Models, Barriers, etc.

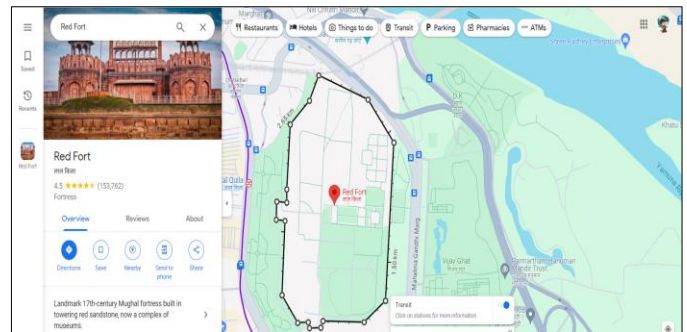


Figure 13: Original Red Fort -Source Google Maps



Figure 14: Our created 3D model of Red Fort

3.4.2 Creating the Level

By navigating through the menu, selecting "File," then "New Level," and then configured the various parameters of level like map size, type, spline type, etc. Utilizing the landscape tool, sculpted the terrain, ensuring it harmonized with our design concept by adjusting its size and shape accordingly. We also used Gaea for creating height-maps of the desired level.

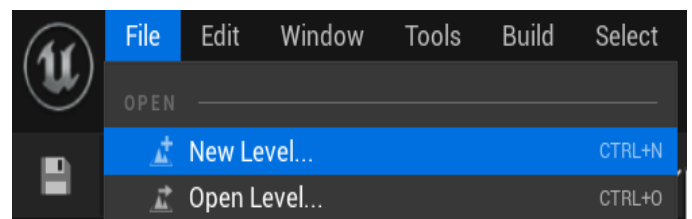


Figure 15: Creating new level



Figure 16: Level Landscaping

3.4.3 Creating Light and Atmosphere

Illuminated lighting and atmosphere manipulating light sources to evoke specific moods and atmospheres. Various lighting techniques such as directional lights, point lights, skyLight, atmospheric fog, clouds were created to balanced brightness and shadows to accentuate architectural details and enhance visual interest.

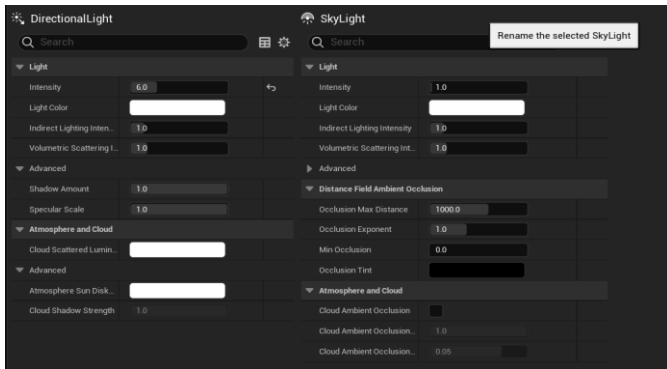


Figure 17: Atmospheric Lighting and Fog

3.4.4 Creating Game Environment

Through the integration of static meshes, including rocks, trees, buildings, and various props sourced from both the Unreal Engine Marketplace and custom-made assets from external modeling software like Blender and Maya, we enriched the environment, infusing it with vitality. Foliage and vegetation were placed using the foliage tool, embellishing the terrain with trees, bushes, grass, and other flora to enhance realism in respective of their atmosphere.

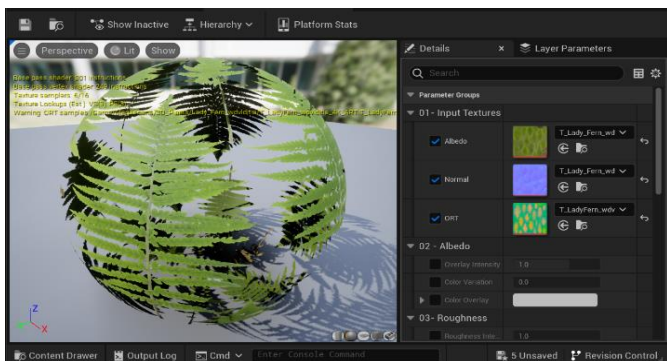


Figure 18: Plant foliage material

3.4.5 Creating Material Layers

Using Quixel Bridge material and Gaea layered the different materials for various objects and ground. The material is layered so that it will show sedimentation of various soil layers with height map. The materials of Trees and their barks were set to give the Indian look to environment. The changes in texture and smoothness of ground are tempered to give the soil realistic look so that it will have more quality in players sight and will blur as the distance increases.

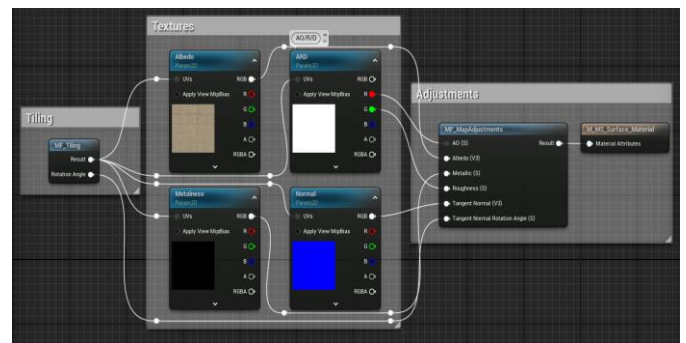


Figure 19: Material Instance of a simple brick

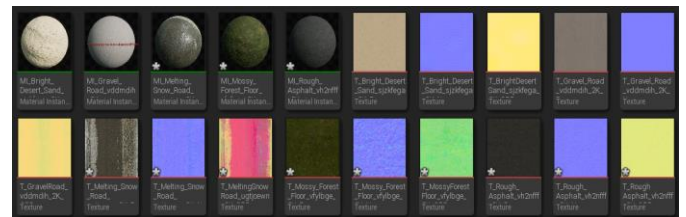


Figure 20: Level materials and textures

3.5 Logic and Functionality

In game development, logic and functionality are the backbone of creating immersive and engaging interactive experiences. Logic encompasses the intricate set of rules and algorithms governing the behavior of the game, dictating everything from player input responses to win/loss conditions. Developers designed gameplay logic to ensure a seamless and enjoyable experience, scripting events, defining mechanics, and implementing artificial intelligence for lifelike character behaviors. On the other hand, functionality refers to the interactive features that bring the game to life, encompassing user interfaces, physics engines for realistic movement, sound and music integration, and networking capabilities for multiplayer experiences. Achieving a delicate balance between sophisticated logic and smooth functionality is essential to crafting a successful game, where players are not only challenged by the intricacies of the game world but also seamlessly immersed in its interactive elements. Whether handling collision detection, character animations, or

multiplayer networking, the synergy between logic and functionality defines the essence of a well-crafted and enjoyable gaming experience.



Figure 21: Checkpoint logic on car movement

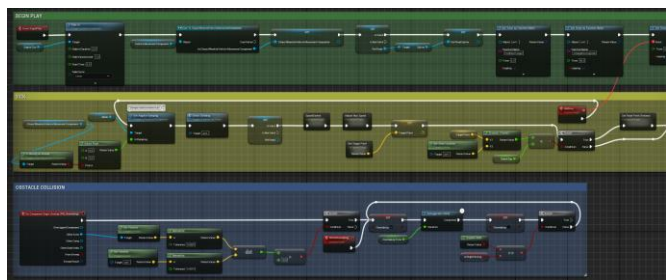


Figure 22: Time-race level logic

3.6 Integrating Model

Integrating models in game development involves the seamless incorporation of 3D or 2D assets, such as characters, environments, and objects, into the game engine. Models serve as the visual representation of the game's elements, contributing to the overall aesthetic and immersive quality of the virtual environment. Game developers utilize modeling software, like Blender or Maya, to create these assets, considering factors such as polygon count, texture quality, and animation rigging. Once the models are crafted, they are integrated into the game engine, where their functionalities and interactions are defined through scripts and logic. The integration process also involves optimizing models for performance, balancing visual fidelity with the game's technical requirements. Successful model integration is essential for delivering a visually captivating and coherent gaming experience, where the artistry and functionality of these digital assets converge to create a compelling and interactive virtual world for players to explore.

3.7 Testing and Debugging

In game development, the phases of testing, debugging, and deployment are integral to ensuring a polished and

stable gaming experience. Testing involves systematically evaluating various aspects of the game, including gameplay mechanics, graphics, and user interfaces, to identify and rectify potential issues. This process encompasses unit testing for individual components, integration testing to assess the collaboration of different elements, and user acceptance testing to gauge player experience. Debugging follows testing, addressing identified issues through the identification and correction of errors in the source code. Game developers employ debugging tools and techniques to enhance performance, fix glitches, and refine the overall functionality of the game. Once testing and debugging are satisfactorily completed, the game is ready for deployment. Deployment involves packaging the finalized game for distribution across platforms, ensuring compatibility and adherence to platform-specific guidelines. This phase also includes considerations for marketing, release strategies, and ongoing support, marking the culmination of the development cycle and the transition to delivering the gaming experience to a wider audience. A thorough and well-executed testing, debugging, and deployment process is crucial for releasing a high-quality and enjoyable game to players worldwide. We used the V-model of software engineering to develop the PRI.



Figure 23: Game logic testing

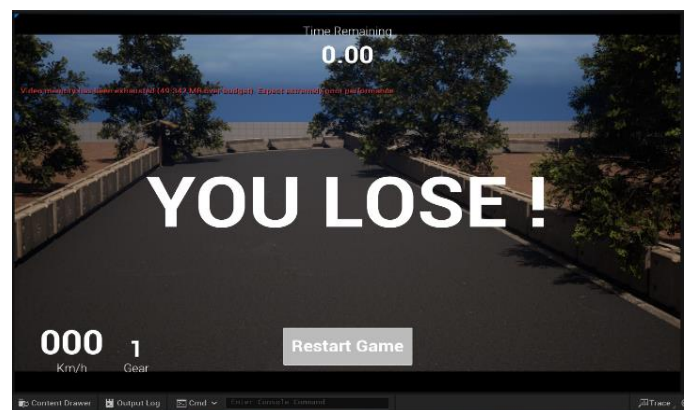


Figure 24: Game lose window

4. CONCLUSIONS

In game development of PRI: Premier race India, the methodology encompasses setting up controllers, pawns, and AI behaviors, integrating third-party assets for enhanced visuals and sound effects, designing intuitive UI/UX elements including in-game menus and HUDs, crafting Indian-themed levels with meticulous attention to detail, implementing logic for gameplay mechanics and functionality, and conducting rigorous testing and debugging phases. Through this approach, the development of PRI is achieved delivering a polished and immersive gaming experience that not only celebrates Indian culture but also captivates players with engaging gameplay, stunning visuals, and seamless functionality across platforms.

REFERENCES

- [1] R. Kothari, S. Nawar, S. Kothari³, Asst. Prof. J. Jeswani (April 2021), "Game Development using Artificial Intelligence in Unreal Engine", IRJET.
- [2] Marvin T. Chan, Christine W. Chan, and Craig Gelowitz. (2015). Development of a Car Racing Simulator Game Using Artificial Intelligence Techniques, International Journal of Computer Games Technology.
- [3] David Michalik *, Miroslav Jirgl, Jakub Arm and Petr Fiedler. (2021). Developing an Unreal Engine 4-Based Vehicle Driving Simulator Applicable in Driver Behavior Analysis—A Technical Perspective. Safety.
- [4] Michael, Jeanny Pragantha, Darius Andana Haris. (2020). Android Arcade-Style Racing Game "CarsTime" With Antigravitation Theme, IOP Conference Series: Materials Science and Engineering.
- [5] <https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-engine-5-3-documentation>
- [6] <https://docs.quadspinner.com/>
- [7] <https://docs.blender.org/manual/en/latest/>

BIOGRAPHIES



Prof. Chandrapal U. Chauhan,
Professor,
Department of Computer Science and
Engineering,
Government College Of Engineering,
Chandrapur 442401.



Mr. Aniket Ashok Tandekar,
Student (Final Year),
Department of Computer Science and
Engineering,
Government College Of Engineering,
Chandrapur 442401.



Mr. Manav Rajkumar Patil,
Student (Final Year),
Department of Computer Science and
Engineering,
Government College Of Engineering,
Chandrapur 442401.



Miss Vijaya Vithobaji Adamane,
Student (Final Year),
Department of Computer Science and
Engineering,
Government College Of Engineering,
Chandrapur 442401.



Miss Aastha Suresh Petkar,
Student (Final Year),
Department of Computer Science and
Engineering,
Government College Of Engineering,
Chandrapur 442401.



Miss Anuja Rajesh Panchariya,
Student (Final Year),
Department of Computer Science and
Engineering,
Government College Of Engineering,
Chandrapur 442401.