**A**

**Project Report On**

**"Cube Runner"**

(CSE204 - Software Group Project - I)

**Prepared by**

Htet Lwin Kyaw (22DCS025 / B-batch)

Mithil Mistry (22DCS049 / C-batch)

Kevin Patel (22DCS063 / C-batch)

**Under the Supervision of**

Prof. Naina Parmar

Prof. Vaishali Vadhavana

**Submitted to**

Charotar University of Science & Technology (CHARUSAT)

for the Partial Fulfilment of the Requirements for the

Degree of Bachelor of Technology (B.Tech.)

in Computer Science and Engineering (CSE)

for 3rd semester B. Tech

**Submitted at**

A blue text on a white background

Description automatically generated

A blue and white logo

Description automatically generated

**Devang Patel Institute of Advance Technology and Research (DEPSTAR)**

At: Changa, Dist.: Anand, Pin: 388421

October 2023

**DECLARATION BY THE CANDIDATE**

I hereby declare that the project report entitled “**Cube Runner**” submitted by our team to Devang Patel Institute of Advance Technology & Research, CHARUSAT in partial fulfilment of the requirement for the award of the degree of **B. Tech** in Computer Engineering, from Devang Patel Institute Of Advance Technology & Research, is a record of Bonafede **CSE204** Software Group Project – I carried out by our team under the guidance of **Prof. Naina Parmar**, **Prof. Vaishali Vadhavana**. I further declare that the work carried out and documented in this project report has not been submitted anywhere else either in part or in full and it is the original work, for the award of any other degree or diploma in this institute or any other institute or university.

Htet Lwin Kyaw (22DCS025)

Mithil Mistry (22DCS049)

Kevin Patel (22DCS063)

Prof. Naina Parmar, Prof. Vaishali Vadhvana,

Assistant Professor, Assistant Professor,

Computer Science & Engineering, Computer Science & Engineering,

DEPSTAR, Changa, Gujarat. DEPSTAR, Changa, Gujarat.

**ACKNOWLEDGEMENT**

We, the developers of a Society Security System “Cube Runner”, with immense pleasure and commitment would like to present the project assignment. The development of this project has given me wide opportunity to think, implement and interact with various aspects of management skills as well as the new emerging technologies.

Every work that one completes successfully stands on the constant encouragement, goodwill, and support of the people around. We hereby avail this opportunity to express my gratitude to the number of people who extended their valuable time, full support, and cooperation in developing the project.

We express deep sense of gratitude towards our Head of the CSE Department, Prof. Chirag Patel, and project guide Prof. Naina Parmar and Prof. Vaishali Vadhavana for the support during the whole session of study and development. It is because of them that I was prompted to do hard work, adopting new technologies.

We are sincerely thankful to all the people who helped us to complete the project in one way or the other. They all together provide us with a favourable environment, and without them it would not have been possible to achieve our goal.

**Thank you,**

Kevin Patel

Mithil Mistry

Htet Lwin Kyaw

**ABSTRACT**

This project presents the development of a captivating and dynamic Cube Runner game for PC platforms. Cube Runner is a popular gaming concept where players navigate a cube-shaped runner through a challenging three-dimensional environment while avoiding obstacles and collecting objects.

Our project aims to create an immersive gaming experience by dynamic level design, and a variety of obstacles and different levels.

Through rigorous development and testing, we have crafted a polished and enjoyable Cube Runner game that caters to players of all ages. This project contributes to the world of PC, offering a fresh and exciting take on a classic gaming concept.

The report details the development process, challenges faced, and the innovative solutions implemented to create this engaging Cube Runner game. It also presents the results of user testing, demonstrating the game's appeal and playability.

Overall, this project report provides insights into the development of a fun and accessible PC game, showcasing the potential for success in the competitive world of PC gaming.

**TABLE OF CONTENTS**

**Declaration by Candidate** I

**Acknowledgement** II

**Abstract** III

**Chapter 1 Introduction 1**

1.1 Project Overview 2

1.2 Objective 4

1.3 Scope 5

1.4 Tools and Technologies 5

**Chapter 2 Project Management 6**

2.1 Project Planning 7

2.1.1 Project Development Approach and Justification 7

2.2 Project Work Scheduling 10

2.2.1 Flow Chart 10

2.2.2 Gantt Chart 11

**Chapter 3 System Requirements Study 12**

3.1 User Characteristics 13

3.1.1 Use Case Diagram 13

3.2 Hardware and software requirements 14

3.2.1 Hardware Specifications 14

3.2.2 Software Specifications 14

**Chapter 4 System Analysis 15**

4.1 Requirement Analysis 16

4.2 User Interface Design 16

4.3 System Architecture 16

4.4 Data and Resource Management 17

4.5 Game Mechanics and Rules 17

**Chapter 5 System Design 18**

5.1 Screen layout 19

5.2 Method Pseudo codes 22

**Chapter 6 System Implementation and testing 30**

6.1 Testing methods 31

6.2 Test Suites Design 32

6.3 Test Cases 34

**Chapter 7 Future Enhancements 37**

**Chapter 8 Conclusion 40**

**References 42**

**LIST OF FIGURES**

Fig 1 Spiral Model 7

Fig 2 Flow Chart 10

Fig 3 Gantt Chart 11

Fig 4 Use Case Diagram 13

Fig 5 Welcome Screen 19

Fig 6 Login Screen 19

Fig 7 Gameplay 19

Fig 8 Gameplay2 20

Fig 9 Gameplay3 20

Fig 10 Game Over Screen 21

**LIST OF TABLES**

Table 1 Test Case No. 01

Table 2 Test Case No. 02

Table 3 Test Case No 03

Table 4 Test Case No 04

**CHAPTER 1**

**INTRODUCTION**

* 1. **PROJECT OVERVIEW**

Cube Runner is an exciting third-person PC game project that offers a fastpaced and engaging gaming experience. This game, developed using Unity and C#, focuses on delivering thrilling.

**Game Concept**

Cube Runner is an adrenaline-pumping game where players take control of a cube-shaped runner, navigating a dynamic three-dimensional environment. The primary objective is to manoeuvre through runway, avoid obstacles, and collect rewards to achieve the highest possible score.

**Gameplay Features**

**Dynamic runway Navigation:** Players must rely on their reflexes and spatial awareness to guide their cube through labyrinthine levels filled with twists, and challenging obstacles.

**Obstacle Avoidance:** The game's core challenge lies in avoiding various obstacles, including long length barriers, all designed to keep players engaged and on their toes.

**Reward Collection:** Players can gather points from collectable cube throughout the plane, enhancing their cube's score.

**Scoring System:** Cube Runner keeps track of players' scores and achievements, encouraging replay ability

**Immersive Elements:** Cube Runner offers an immersive gaming experience through various elements:

**Engaging Visuals:** The game features visually appealing and dynamic environments, with vivid colours, intricate designs, and captivating level layouts.

**Responsive Controls:** Precise and responsive controls ensure that players can navigate the cube with ease, enhancing the overall gaming experience.

**Adaptive Soundscapes:** The game's soundtrack and sound effects adapt to the player's progress, providing an immersive audio experience that complements the action.

Cube Runner takes players on an exhilarating journey where quick thinking and reflexes are essential for success. With its captivating gameplay, stunning visuals, and immersive audio, this PC game aims to offer an exciting and addictive gaming experience that appeals to a broad audience of gamers seeking challenges and thrills.

**In contrast, Cube Runner provides a different kind of intensity, where players engage with their environment, sharpen their spatial skills, and strive for high scores in a visually stimulating world.**

* 1. **OBJECTIVE**

The primary objective of this project is to design, develop, and deliver an engaging and immersive third-person PC game experience known as "Cube Runner." This game aims to captivate players with its fast-paced gameplay, dynamic environments, and spatial challenges while providing an enjoyable and accessible gaming experience.

**Key project objectives include:**

**Game Development:** Create a fully functional and polished PC game using Unity and C#, ensuring a seamless and responsive gaming experience.

**Engaging Gameplay:** Develop exciting and challenging gameplay mechanics that involve navigating a cube-shaped runner object through complex runway, avoiding obstacles, and collecting point-object.

**Visual Appeal:** Design visually stimulating game environments featuring vivid colours, intricate level layouts, and dynamic elements to keep players engaged and entertained.

**Replay ability:** Create a scoring system and incorporate points to encourage replay ability.

**By achieving these objectives, this project aims to deliver a compelling and enjoyable Cube Runner PC game that appeals to a broad audience of gamers. It seeks to provide an engaging gaming experience that challenges players' reflexes, spatial awareness, and problem-solving skills within a visually stimulating and immersive digital world.**

* 1. **SCOPE**

The scope of the Cube Runner project encompasses a comprehensive set of objectives and considerations that define the boundaries and aspirations of this game development endeavour. Understanding the scope is essential for outlining the project's goals, functionalities, and limitations.

* 1. **TOOLS & TECHNOLOGY USED**

**1. Unity 3D Game Engine:**

**Description:** Unity 3D is a powerful and versatile game engine renowned for its capabilities in creating visually stunning and interactive 2D and 3D games. It offers a wide array of tools and features for game development, including rendering, physics, animation, and scripting. Unity simplifies the development process by providing a user-friendly interface and extensive documentation, making it a preferred choice for both novice and experienced game developers.

**Role in Cube Runner:** Unity 3D serves as the core development platform for Cube Runner, enabling the creation of 3D environments, character animation, and game mechanics. It facilitates the integration of assets, scripting in C#, and testing, ultimately bringing the game to life.

**2. C# Programming Language:**

**Description:** C# is a modern, object-oriented programming language known for its versatility and ease of use. It is a popular choice for game development due to its robustness, compatibility with the Unity engine, and extensive libraries. C# allows developers to create complex game logic, implement user interfaces, and manage game events efficiently.

**Role in Cube Runner:** C# serves as the primary scripting language for Cube Runner. It is utilized to define game mechanics, control player interactions, handle collisions, and manage gameplay logic. C# code is integral in shaping the player's experience and ensuring the smooth functioning of the game.

**CHAPTER 2**

**PROJECT MANAGEMENT**

**2.1 PROJECT PLANNING**

**2.1.1 Project Development Approach and Justification**

In the development of Cube Runner, a well-structured project planning approach was adopted to ensure the successful creation of the game. The chosen approach aligns with the principles of the SPIRAL model, a risk-driven process model generator frequently applied to software projects. The SPIRAL model provides a dynamic framework that allows for flexibility and adaptability based on the unique risk patterns and requirements of the project.

A diagram of a diagram

Description automatically generated

Fig1 Spiral Modal Diagram

**Key Considerations for the SPIRAL Model in Cube Runner Development:**

**Advanced Knowledge of Requirements:** Cube Runner benefits from a comprehensive understanding of its requirements from the outset, which forms the foundation for development.

**Low-Risk Requirement Implications:** The project's requirements exhibit no unresolved, high-risk implications related to factors such as cost, schedule, performance, safety, security, user interfaces, and organizational impacts.

**Stable Requirements:** The nature of Cube Runner's requirements is not anticipated to undergo significant changes during development, ensuring a stable development environment.

**Well-Understood Architecture:** The project is based on a well-understood architectural framework for implementing the requirements effectively.

**Sequential Development Timeline:** Sufficient calendar time exists to proceed with development in a sequential manner.

**Advantages of the SPIRAL Model for Cube Runner:**

**Realism:** The SPIRAL model aligns with the iterative nature of software development, particularly beneficial for projects like Cube Runner with evolving or unclear requirements.

**Flexibility:** Incorporating the strengths of both waterfall and evolutionary development methods, the SPIRAL model offers adaptability throughout the project's lifecycle.

**Enhanced Project Visibility:** The SPIRAL model provides a clear and structured framework, ensuring that project progress and risk management are consistently monitored and transparent.

**In conclusion,** the adoption of the SPIRAL model for project planning has played a pivotal role in guiding the development of Cube Runner. Its riskdriven approach, emphasis on stakeholder alignment, and commitment to addressing high-risk sub-problems have contributed to the project's success and its ability to adapt to evolving requirements. This model has been instrumental in ensuring that Cube Runner remains a dynamic and engaging PC game.

**2.2 Project Work Scheduling**

**2.2.1 Flow Chart**

A diagram of a flowchart

Description automatically generated

Fig2 Flow Chart

**2.2.2 Gantt Chart**

A screenshot of a graph

Description automatically generated

Fig3 Gantt Chart

**CHAPTER 3**

**SYSTEM REQUIREMENTS STUDY**

**3.1 USER CHARACTERISTICS**

* **End Users**

This product is designed mainly for END USERS. So, it is feasible for the endusers to directly play the game and interact with the virtual 3D environment.

* **Administrator**

He can add, delete or update any functionalities he wants. He has access to the core scripts and he can do optimization the way he wants. He is responsible for the design of the environment.

**3.1.1 Use Case Diagram**

A diagram of a game

Description automatically generated

Fig 4 Use Case Diagram

**3.2 HARDWARE AND SOFTWARE REQUIREMENTS**

**3.2.1 Hardware specification**

Minimum of 8GB RAM

A capable machine with both CPU and GPU

Access to a keyboard

Unity game engine

C# runtime environment

**3.2.2 Software specification:**

OS: Windows 7 or above

Unity 3D Game Engine version 2021.3.30f1

**CHAPTER 4**

**SYSTEM ANALYSIS**

**SYSTEM ANALYSIS**

System analysis is a crucial phase in the development of Cube Runner, as it lays the foundation for creating a well-structured and efficient game. During this phase, the project team assesses and defines the requirements, constraints, and objectives that guide the entire development process. The analysis encompasses several key aspects:

* 1. **Requirement Analysis:**

Functional Requirements: Identifying the core functionalities of Cube Runner was a primary focus. This includes determining how players navigate the cube, interact with obstacles, and progress through the game levels. Non-Functional Requirements: In addition to functionality, non-functional requirements such as performance, responsiveness, and graphical quality were considered. These requirements ensure a smooth and engaging gameplay experience.

* 1. **User Interface Design:**

User Experience (UX): Understanding the user's perspective was integral to the system analysis. Design decisions related to the user interface, and controls were made to enhance the overall user experience and make the game accessible and enjoyable. Usability Testing: The analysis phase may have included usability testing to gather user feedback and refine the interface for better intuitiveness and player engagement.

* 1. **System Architecture:**

Game Engine Selection: Choosing the appropriate game engine, such as Unity, was a critical decision. The analysis phase considered the engine's capabilities, compatibility with the project's goals, and the ease of development. Game Components: Identifying the core game components, including player-controlled cube mechanics, obstacle generation, scoring, and level progression, was a fundamental part of the system architecture analysis.

* 1. **Data and Resource Management:**

Data Storage: Determining how game progress, high scores, and player achievements are stored and managed was a key consideration. This analysis ensures that player data is saved and retrieved accurately.

* 1. **Game Mechanics and Rules:**

**Game Logic:** Analysing and defining the game's rules and mechanics, including cube movement, collision detection, scoring, and difficulty scaling, ensured a balanced and engaging gameplay experience.

**Level Design:** Considering the design of game levels, including obstacles' placement and progression in difficulty, was a significant part of the analysis to maintain player interest and challenge.

**In conclusion,** system analysis for Cube Runner was a comprehensive process that encompassed various critical aspects of game development. It laid the groundwork for defining requirements, designing the user interface, structuring the system architecture, managing data and resources, establishing game mechanics, and planning for future scalability and enhancements. This analysis phase was instrumental in shaping Cube Runner into an engaging and well-structured PC game.

**CHAPTER 5**

**SYSTEM DESIGN**

* 1. **SCREEN LAYOUT**

A blue background with white text

Description automatically generated

Fig 5 Welcome Screen

A blue screen with white text

Description automatically generated

Fig 6 Login Screen

A long shot of a colorful building

Description automatically generated with medium confidence

Fig 7 Gameplay1

A video game of a video game

Description automatically generated

Fig 8 Gameplay2

A video game of a city

Description automatically generated

Fig 9 Gameplay3

A screenshot of a video game

Description automatically generated

Fig.10 Game Over Screen

* 1. **METHOD PSEUDO CODE**

**followplayer.cs**

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class followplayer : MonoBehaviour  {  public Transform follow;  public float offset;  // Start is called before the first frame update  void Start()  {  }  // Update is called once per frame  void Update()  {  Vector3 camerapos = transform.position;  camerapos.x = follow.position.x + offset;  transform.position = camerapos;  }  } |

**GameController.cs**

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine.SceneManagement;  using UnityEngine;  public class gamecontroller : MonoBehaviour  {  public GameObject gameoverpannel;  public GameObject winp;  public GameObject tts;  public GameObject scoret;  public GameObject stp;  public GameObject sup;  public GameObject sip;  public GameObject inputf1;  public GameObject inputf2;  public GameObject inputf3;  public GameObject inputf4;  public playerscript pk;  public GameObject lvl1;  public GameObject lvl2;  public void Start()  {  stp.SetActive(true);  sup.SetActive(false);  sip.SetActive(false);  tts.SetActive(false);  pausegame();  pk.enabled = false;  }  public void Update()  {  if (Input.GetKeyDown(KeyCode.Mouse0) || Input.GetKeyDown(KeyCode.Space))  {  startgame();  }  }  public void signup()  {  stp.SetActive(false);  sup.SetActive(true);  sip.SetActive(false);  }  public void signin()  {  stp.SetActive(false);  sup.SetActive(false);  sip.SetActive(true);  }  public void enter()  {  stp.SetActive(false);  sup.SetActive(false);  sip.SetActive(false);  scoret.SetActive(false);  tts.SetActive(true);  pausegame();  gameoverpannel.SetActive(false);  winp.SetActive(false);  pk.enabled = true;  }  public void gameover()  {  scoret.SetActive(false);  gameoverpannel.SetActive(true);  }  public void win()  {  scoret.SetActive(false);  winp.SetActive(true);  }  public void restart()  {  SceneManager.LoadScene("lvl 1");  }  public void quitgame()  {  Application.Quit();  }  public void pausegame()  {  Time.timeScale = 0f;  }  public void startgame()  {  scoret.SetActive(true);  tts.SetActive(false);  Time.timeScale = 1f;  }  public void next()  {  }  } |

**player\_collision.cs**

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class playercollision : MonoBehaviour  {  public playerscript kp;  public score lion;  public gamecontroller gmc;  public gamecontroller wmc;  public AudioSource cubecollect;  public AudioSource gmo;  public AudioSource winnn;  private void OnTriggerEnter(Collider other)  {  if (other.gameObject.tag == "collectable")  {  cubecollect.Play();  lion.addscore(1);  Destroy(other.gameObject);  }  if (other.gameObject.tag == "win")  {  winnn.Play();  wmc.win();  Destroy(other.gameObject);  kp.enabled = false;  }  }  private void OnCollisionEnter(Collision collision)  {  if (collision.gameObject.tag == "obstacles")  {  gmo.Play();  gmc.gameover();  kp.enabled = false;  }  }  } |

**playerscript.cs**

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class playerscript : MonoBehaviour  {  public Rigidbody body;  public float force = 100f;  public float l = 100f;  public float speed = 10f;  public float speed22 = 1f;  public float max\_z;  public float min\_z;  void Update()  {  Vector3 playerpos = transform.position;  playerpos.z = Mathf.Clamp(playerpos.z, min\_z, max\_z);  transform.position = playerpos;  if (Input.GetKey(KeyCode.RightArrow))  {  transform.position = transform.position - new Vector3(0, 0, speed \* Time.deltaTime);  }  if (Input.GetKeyDown(KeyCode.D))  {  transform.position = transform.position - new Vector3(0, 0, speed \* Time.deltaTime);  }  if (Input.GetKey(KeyCode.LeftArrow))  {  transform.position = transform.position + new Vector3(0, 0, speed \* Time.deltaTime);  }  if (Input.GetKeyDown(KeyCode.A))  {  transform.position = transform.position + new Vector3(0, 0, speed \* Time.deltaTime);  }  if (Input.GetKey(KeyCode.Space))  {  transform.position = transform.position + new Vector3(0, speed \* Time.deltaTime, 0);  }  //body.AddForce(l \* Time.deltaTime, 0, 0);  }  /\*private void FixedUpdate()  {  body.AddForce(l \* Time.deltaTime, 0, 0);  }\*/  public void FixedUpdate()  {  body.AddForce(force \* Time.deltaTime, 0, 0);  }  } |

**scoretext.cs**

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine.UI;  using TMPro;  using UnityEngine;  public class score : MonoBehaviour  {  public TextMeshProUGUI scoretext;  public TextMeshProUGUI fst;  public TextMeshProUGUI wfst;  int myscore = 0;  // Update is called once per frame  void Update()  {  scoretext.text = myscore.ToString();  fst.text = "Score :-" + myscore.ToString();  wfst.text = "Score :-" + myscore.ToString();  }  public void addscore(int score)  {  myscore = myscore + score;  }  } |

**CHAPTER 6**

**SYSTEM TESTING**

**6.1 TESTING METHOD**

In this project, we have used the white box testing because the source code is completely developed by us and available to us. We have chosen these techniques because of following reasons:

• Easy to find out errors

• Make it efficient for use

• Optimize the code

**Advantages:**

* As the tester has knowledge of the source code, it becomes very easy to find out which type of data can help in testing the application effectively.
* It helps in optimizing the code.
* Extra lines of code can be removed which can bring in hidden defects.

**6.2 TEST SUITES DESIGN**

In the context of Cube Runner, test suites are instrumental in organizing and executing a comprehensive testing strategy. Each test case within a test suite represents a specific aspect or functionality of the game. The design of test suites ensures thorough coverage of the game's functionality while facilitating efficient testing and issue identification.

Test Suite Structure: Test suites are organized to cover various aspects of the Cube Runner game. Each test suite may contain multiple test cases, and these test cases collectively assess different components and scenarios within the game.

**Test Suite 1: Gameplay Mechanics**

Test Case 1: Cube navigation and control

Test Case 2: Collision detection and response

Test Case 3: Score functionality

Test Case 4: Game difficulty levels

**Test Suite 2: Visual and Aesthetic Testing**

Test Case 1: Graphics rendering and performance

Test Case 2: Visual effects and animations

Test Case 3: User interface elements and layout

**Test Suite 3: Audio and Sound Testing**

Test Case 1: Soundtrack integration

Test Case 2: Sound effects synchronization

Test Case 3: Audio quality and immersion

**Testing Scenarios:**

Test suites are essential for identifying dependencies and sequential execution scenarios within the game. For instance, successful completion of a navigation and control test case (Test Case 1) may be necessary before proceeding to test collision detection (Test Case 2).

**Types of Test Suites:**

Within the Cube Runner testing framework, different types of test suites serve specific purposes:

**Build Verification Tests (BVT):** A collection of test cases that perform basic validation of fundamental game functions. These tests are executed after each build to ensure the core functionality is intact.

**Smoke Tests:** A set of test cases that verify basic game functionality, serving as the initial test level after system changes.

**Level Integration Tests:** Test suites that evaluate cross-level functionality and ensure seamless transitions between game levels.

**Feature-specific Tests:** Test cases within dedicated test suites that focus on specific game features, ensuring thorough testing of each element.

**Effective test suite design in Cube Runner facilitates efficient testing, aids in identifying critical issues, and ensures the overall quality and stability of the game. Each test case contributes to a comprehensive evaluation of the game, ensuring it meets the desired standards of gameplay and functionality.**

**6.3 TESTING CASES**

In the context of software engineering and game development, test cases are a vital component of ensuring that a game, such as Cube Runner, functions as intended. Each test case represents a specific scenario or condition under which the game's features and functionalities are evaluated to determine whether they meet the established criteria.

**Test Case Structure:**

A typical test case comprises several key elements:

**Test Case ID:** A unique identifier for the test case, allowing for easy reference and tracking.

**Module Name:** The specific module or aspect of the game being tested.

**Pre-condition:** The necessary conditions that must be met before executing the test case.

**Expected Result:** The anticipated outcome or behavior of the game when the test case is executed.

**Actual Result:** The observed behavior of the game during the test case execution.

**Pass/Fail:** A determination of whether the actual result aligns with the expected result.

**Table.1 Test Case No. 01**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Module Name | Pre-condition | Expected Result | Actual Result | Pass/Fail |
| T01 | Cube Navigation and Control | The game has been launched, and the player is in the starting position | The cube's navigation and control function correctly | The cube's navigation and control function correctly | Pass |

**Table.2 Test Case No. 02**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Module Name | Pre-condition | Expected Result | Actual Result | Pass/Fail |
| T02 | Visual and Graphics Rendering | The game has been launched, and the player is in the starting position | Graphics rendering is smooth and without flickering | Graphics rendering is smooth and without flickering | Pass |

**Table.3 Test Case No. 03**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Module Name | Pre-condition | Expected Result | Actual Result | Pass/Fail |
| T03 | Audio and Sound Effects | The game has audio elements enabled | Sound effects are synchronized with gameplay | Sound effects are synchronized with gameplay | Pass |

**Table.4 Test Case No. 04**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Module Name | Pre-condition | Expected Result | Actual Result | Pass/Fail |
| T05 | Replay and End Game | The player has completed a game session | The player is prompted to replay or exit the game | The player is prompted to replay or exit the game | Pass |

**These test cases represent a subset of the comprehensive testing strategy for Cube Runner. They are designed to assess critical aspects of the game, including navigation, graphics, audio, and replay functionality. Through meticulous testing and analysis, these test cases ensure that Cube Runner meets the desired standards of functionality and player experience. Additional test cases can be created to further scrutinize the game's features and behaviours.**

**CHAPTER 7**

**FUTURE ENHANCEMENT**

**FUTURE ENHANCEMENTS**

As the development of Cube Runner progresses and the game evolves, several avenues for future enhancements and improvements come to the forefront. These enhancements aim to elevate the player experience, expand the game's accessibility, and enrich its overall quality. Below are key areas for potential future development.

1. **Expanding Game Levels:**

To enhance player enjoyment and prolong engagement, consider expanding the game with additional levels featuring diverse themes and challenges. These new levels can introduce fresh gameplay mechanics, obstacles, and rewards, keeping players engaged and eager to explore new content.

1. **Cross-Platform Optimization:**

To broaden the game's reach and cater to a wider audience, optimizing Cube Runner for a broader range of platforms, including iOS and Android, is a promising avenue. This expansion allows players to enjoy the game on a variety of devices, from personal computers to smartphones and tablets, thereby increasing accessibility and player base.

1. **Enhanced Visual Realism:**

Continuing to enhance the visual realism of the game's assets can contribute to a more immersive experience. Factors such as higher polygon counts, improved textures, advanced lighting techniques, and realistic animations can be integrated to make the game's visuals more convincing and appealing.

1. **Immersive Soundscapes:**

Building upon the game's audio elements, the incorporation of immersive soundscapes can elevate the auditory experience. Realistic and dynamic sound effects, spatial audio, and an adaptive soundtrack can further immerse players in the game's environment, heightening their engagement and emotional connection.

1. **Multiplayer and Social Features:**

Introducing multiplayer and social features can transform Cube Runner into a more interactive and socially engaging experience. Implementing competitive or cooperative multiplayer modes, leader boards, and the ability to challenge friends can foster a sense of community and competition among players.

1. **Continuous Performance Optimization:**

Efforts to optimize the game's codebase should remain ongoing to ensure efficient performance and a seamless gaming experience. Continuously improving code quality, reducing memory consumption, and enhancing execution speed can lead to a smoother and more responsive gameplay experience.

**These future enhancements represent opportunities to refine and expand the Cube Runner experience, appealing to a broader audience and ensuring that the game remains engaging and captivating for both existing and new players. By embracing these possibilities, Cube Runner can evolve into an even more exciting and immersive gaming adventure.**

**CHAPTER 8**

**CONCLUSION**

**CONCLUSION**

Cube Runner represents an exciting venture into the world of gaming, where immersive experiences and captivating gameplay take centre stage. As the gaming industry continues to evolve, this project has aimed to contribute to the growing medium of PC gaming by offering a unique and thrilling gaming experience. It is important to reflect on the journey and achievements of the project. Cube Runner is a testament to the dynamic and ever-evolving landscape of PC gaming. While it has successfully realized its primary functionalities, it also signifies the ongoing journey to unlock its full potential. With dedication, continued development, and a commitment to delivering a captivating gaming experience, Cube Runner can make a lasting impact in the world of gaming. It stands as a promising venture into the realm of immersive and engaging gameplay, setting the stage for future advancements and innovation.

**REFERENCES**

* Unity Real-Time Development [Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine](https://unity.com/)
* Unity User Manual [Unity - Manual: Unity User Manual 2022.3 (LTS) (unity3d.com)](https://docs.unity3d.com/Manual/index.html)
* Learning Pathways <https://learn.unity.com/>
* C# for Game Development <https://www.javatpoint.com/c-sharp-game-development>
* C# with Unity <https://unity.com/how-to/beginner-game-coding-resources>
* “The Art of Game Design: A Book of Lenses” By Jesse Schell published in 2008
* “GAMEDEV: 10 Steps to Making Your First Game Successful” by Wlad Marhulets
* “GAME PROGRAMMING PATTERNS” by Robert Nystrom published in 2 November 2014
* “Unity From Zero to Proficiency (Beginner): A step-by-step guide to coding your first game with Unity in C#” by Patrick Felicia