3D THE HAUNTED JAUNT ESCAPE ROOM GAME IN UNITY

Submitted in partial fulfillment of the requirements of

University of Mumbai

For the Degree of

Bachelor of Engineering in CSE (AI & ML)

Submitted by

Ms. SHRUTI KAMBALI [ROLL NO:- 24]

Under the guidance of

PROF. KRISHNAJI SALGAONKAR

DEPARTMENT OF CSE (AI & ML)

SMT. INDIRA GANDHI COLLEGE OF ENGINEERING

Ghansoli, Navi Mumbai - 400701

Academic year: 2023-2024

Project Report Approval for B.E.

This project report entitled "**3D The Haunted Jaunt Escape Room Game in Unity**"

By

Ms. SHRUTI KAMBALI [ROLL NO: - 24]

approved for the degree of Bachelor of Engineering in Computer Engineering, Semester VII, University of Mumbai.

---------- ---------- ----------

Examiner 1 Examiner 2 Prof. Krishnaji Salgaonkar

Internal Guide

---------- ----------

Prof. Sonali.Deshpande Dr. Sunil Chavan

Head of Department Principal

Date:
Place: Ghansoli, Navi Mumbai.

Declaration

We declare that this written submission represents our own ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified anyact/data/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Ms. SHRUTI KAMBALI [ROLL NO:-31]

Date:
Place: Ghansoli, Navi Mumbai.

Z

Abstract

Title: - “3D The Haunted Jaunt Escape Room Game in Unity ”

Haunted Jaunt" is a captivating 3D beginner-level virtual reality (VR) project that has been developed to provide an immersive and educational experience for novice VR enthusiasts. This project leverages the Unity game engine and VR technologies to offer participants a unique journey into the world of game development and VR content creation.

In this VR adventure, players take on the persona of John Lemon, as they navigate through a series of meticulously crafted 3D scenes, each with an eerie and immersive ambiance. The project’s primary objective is to deliver a hands-on introduction to the development of VR experiences, focusing on essential VR development principles.

It works on 12 stages:-

1) **Concept and Design:-** In this initial stage, you conceptualize the VR experience. You define the game's story, scenes, characters, and overall design.
2) **Development: -** Development involves creating the 3D scenes, models, characters, and assets required for the VR experience.
3) **Integration of VR Technology:-** VR technology integration includes setting up and configuring VR hardware, such as headsets and controllers
4) **Scene Building: -** This stage involves designing and building each individual 3D scene. You create immersive environments that are optimized for VR, considering elements like lighting, audio, and scale to enhance the sense of presence.
5) **VR Interaction: -** Implement VR interactions and mechanics that allow players to navigate, pick up objects, or trigger events using VR controllers.
6) **User Interface Design:-** Design and implement user interfaces tailored for VR. This includes menus, heads-up displays (HUDs), and other in-game UI elements that are presented in a VR-friendly way
7) **Animation and Interactivity:-** Animate characters and objects within the VR world to make the experience more engaging and dynamic. Ensure that interactive elements respond to user input.
8) **Unity Development:-** Utilize the Unity development platform to build and optimize the VR experience. Unity provides tools for VR development, including camera setup, asset management, and scripting for VR-specific functionality.
9) **Testing and Debugging:-** Regularly test the VR experience to identify and address issues related to performance, usability, and VR-specific challenges. Debug and fine-tune the project to ensure a smooth and enjoyable VR experience.
10) **Optimization for VR:-** Optimize the project for VR performance by reducing resource-intensive elements and ensuring a stable frame rate.
11) **Deployment:-**Once the VR experience is polished and tested, you prepare it for distribution on VR platforms.
12) **User Experience:-** The final stage is the user experience, where players immerses in VR adventure.

Z

List of Figures

Z

Index

Z

CHAPTER 1
INTRODUCTION

Z

1. INTRODUCTION

In the ever-expanding realm of virtual reality (VR), few experiences rival the immersive and interactive allure of an escape room. The fusion of technology and storytelling in VR escape rooms provides an unparalleled opportunity for individuals to dive headfirst into thrilling narratives while actively engaging their problem-solving skills. "John Lemon's Haunted Jaunt: The Escape Room" brings the world of virtual reality gaming to life, offering an exhilarating and educational journey for those keen to venture into the dynamic field of VR development.

This project weaves an enthralling tale where participants step into the shoes of the enigmatic John Lemon, thrust into a haunted escape room brimming with riddles, challenges, and suspense. "John Lemon's Haunted Jaunt: The Escape Room" is an exploration of foundational VR development principles, which equip participants with the knowledge and skills required to craft their own captivating VR escape room experiences.

The Enigmatic World of VR Escape Rooms:

At the core of this project lies the creation of an escape room that melds seamlessly with the immersive capabilities of virtual reality. It presents participants with a series of intricate puzzles, mysterious clues, and interactive elements that challenge them to unravel the secrets of the room. The journey unfolds within meticulously designed 3D scenes, each offering a unique piece of the overarching narrative.

1.1 Problem Statement:-

Virtual reality (VR) development offers a realm of exciting possibilities, enabling developers to create immersive, interactive experiences that transport users to entirely new worlds. However, for beginners in VR and game development, the transition from concept to creation can be daunting and filled with challenges. There exists a need for a project that addresses these challenges and serves as an instructive platform for those who are eager to venture into VR development.

The burgeoning field of virtual reality (VR) development holds immense promise, offering the potential for immersive, interactive experiences that can captivate and engage users. However, for newcomers to VR and game development, the journey from concept to realization can be a daunting challenge, marked by a steep learning curve and numerous barriers. This project addresses the key issues that often confront beginners in VR development.

These challenges include the intricate nature of VR technologies, the complexity of scripting for interactions and user interfaces, the art of crafting immersive VR scenes, and the scarcity of accessible resources tailored for novices. "John Lemon's Haunted Jaunt: 3D Beginner Project!" was conceived as a solution to these problems, providing a structured and hands-on learning experience that empowers beginners to dive into the fascinating world of VR development. This project serves as an essential stepping stone, enabling newcomers to build foundational skills and confidence to embark on their own VR development journeys.

Z

This project recognizes the following problems that beginners in VR development often face:

Learning Curve in VR Development: The world of VR development can be overwhelming for newcomers, with its unique hardware requirements, interaction mechanisms, and immersive design principles. Beginners require a structured, hands-on learning experience.

Scripting and Interaction Complexity: VR games demand complex scripting for interactions and user interfaces, which can be a barrier for those new to programming and scripting languages.

Scene Design for VR: Creating immersive scenes that take full advantage of VR capabilities, including scale, lighting, and sound design, requires expertise that many beginners lack.

Access to Resources: Access to reliable resources, including tutorials and guides tailored for beginners in VR development, is essential for a successful learning journey.

"John Lemon's Haunted Jaunt: 3D Beginner Project!" seeks to address these challenges by offering a structured and engaging introduction to VR development. It provides a safe space for beginners to explore the fundamentals of VR development, script interactions, design immersive scenes, and create compelling user experiences. This project serves as a foundation for novices to build the skills and confidence needed to venture further into the captivating realm of virtual reality.

1.2 Objectives:-

The primary objectives of "John Lemon's Haunted Jaunt: The Escape Room" are to provide an immersive, educational, and engaging experience for participants while introducing them to the world of virtual reality (VR) development. The project aims to achieve the following specific goals:

1. **Immersive Scene Design:** To teach participants the art of crafting immersive 3D scenes that fully utilize VR's unique capabilities, including spatial awareness, scale, lighting, and sound design.

2. **VR Interaction and Puzzles:** To introduce participants to VR interaction mechanisms and scripting, enabling them to create interactive objects, puzzles, and challenges that respond to player actions, thereby enhancing problem-solving skills.

3. **VR User Interface (UI) Design:** To instruct participants in the creation of user interfaces tailored specifically for VR environments, including the design of VR menus, heads-up displays (HUDs), and interactive UI components that enhance the overall user experience.

4. **VR Animation and Interaction:** To emphasize the role of animation in creating a dynamic VR environment by animating characters, objects, and interactive elements, thus enhancing immersion and engagement.

5. **Unity Development for VR:** To familiarize participants with Unity as a development platform for VR, guiding them through the setup of VR hardware, camera management, and VR-specific scripting.

6. **Problem-Solving and Critical Thinking:** To challenge participants with a series of intricate puzzles and mysteries within the escape room, fostering problem-solving skills and encouraging them to think critically to progress.

Z

7. **Project Development Skills:** To equip participants with the skills required to develop a VR project from conception to realization, including project planning, asset creation, and optimization for VR platforms.

8. **Confidence in VR Development:** To instill confidence in participants, enabling them to embark on their own VR development journeys and encouraging them to explore the limitless possibilities of VR content creation.

9. **Educational Value:** To create a project that not only entertains but also educates, introducing participants to the principles and intricacies of VR development, making it a valuable resource for those new to the field.

10. **Resource for Future Learning:** To produce a project that serves as a resource for beginners in VR development, offering a blueprint for constructing immersive VR escape room experiences and guiding them on their path toward becoming proficient VR developers.

1.3 Proposed Solution:-

In response to the challenges and objectives identified in the problem statement, "John Lemon's Haunted Jaunt: The Escape Room" offers a comprehensive and engaging solution designed to address the specific needs of novice virtual reality (VR) developers and enthusiasts. The proposed solution revolves around the creation of an immersive, educational, and interactive VR escape room experience. Here is a breakdown of the key elements of our solution:

1. **Immersive Scene Design:** The project places a strong emphasis on scene design that leverages the full capabilities of VR. By introducing participants to the principles of spatial awareness, scale, lighting, and audio design, we enable them to craft environments that engage all the senses, creating a convincing VR world.

2. **VR Interaction and Puzzles:** "John Lemon's Haunted Jaunt" introduces participants to the intricacies of VR interactions and scripting. The escape room is brimming with interactive objects, puzzles, and challenges that respond to player actions, fostering problem-solving skills, critical thinking, and an understanding of VR-specific interaction mechanisms.

3. **VR User Interface (UI) Design:** Participants learn to create VR-friendly user interfaces, including menus, heads-up displays (HUDs), and interactive UI components. These elements enhance the player experience and underscore the importance of intuitive and well-designed VR interfaces.

4. **VR Animation and Interaction:** The project harnesses the power of animation to breathe life into the escape room. Animated characters, objects, and interactive elements make the VR experience dynamic, engaging, and immersive.

5. **Unity Development for VR:** Leveraging Unity as the development platform, we guide participants through the setup of VR hardware, camera management, and VR-specific scripting. By familiarizing them with Unity's VR capabilities, we empower them to tackle VR projects with confidence.

6. **Project Development Skills:** The project not only covers VR-specific skills but also provides a holistic understanding of project development. Participants gain experience in project planning, asset creation, optimization for VR platforms, and the intricacies of managing a complete VR project from start to finish.

Z

1.4 Report Organization

- **In Chapter 2,** we will see the literature survey which will tell us more about the background of the projectincluding the work that has already been done in this field.
- **In Chapter 3,** Planning and Formulation of the project is given. Usage of Agile model and how weintegrated and worked around the model.
- **In Chapter 4,** shines light upon the Requirements that are needed and analysis of the system to uncover theadditional requirements of the project.
- **In Chapter 5,** the system proposed is introduced which will tell the deep specification of the project and willtell how the different modules of the system will work, the flow of the project regarding data flow, control flow and other flow of the system.
- **In Chapter 6,** we see the implementation of the algorithm of the project and process of model building.
- **In Chapter 7,** Conclusion and Future Scope of this project is mentioned.

1.5 Technology/Platform:-

Unity for VR Development: Unity is widely recognized for its powerful and accessible tools for VR development. It supports various VR platforms, including Oculus, HTC Vive, and Windows Mixed Reality, making it a versatile choice for creating VR content.

Comprehensive Asset Library: Unity boasts an extensive asset store with a wide range of 3D models, animations, and VR-specific assets. This resourceful library significantly expedited the development process, allowing for quick prototyping and implementation of assets for "The Haunted Jaunt Escape Room Game."

VR Interaction Framework: Unity offers a robust VR interaction framework, simplifying the implementation of VR-specific interactions and controls. This was instrumental in creating a seamless and engaging escape room experience that was responsive to player actions.

Cross-Platform Compatibility: Unity enables cross-platform development, meaning that the escape room game can be enjoyed on a variety of VR devices and platforms. This inclusivity enhances the reach and accessibility of the game.

User-Friendly Development Environment: Unity's user-friendly interface and intuitive development environment were conducive to a smooth and efficient development process. It allowed for quick iteration and testing, saving valuable development time.

Z

CHAPTER 2

REVIEW OF LITERATURE

Z

2. Review of Literature

2.1 Research Paper Analysis

Virtual Reality (VR) as an Educational and Entertainment Platform:

The use of VR as an educational and entertainment platform has witnessed significant growth in recent years. Scholars such as Slater and Wilbur (1997) and Riva et al. (2011) have explored the potential of VR in creating immersive experiences that engage users on both educational and entertainment fronts. The immersive nature of VR allows users to be transported to virtual worlds, making it an ideal medium for educational games and experiences.

- **VR Development in Unity:** Unity, a popular game development engine, has played a pivotal role in simplifying the development of VR content. Academic works like "Unity Virtual Reality Projects" by Linowes (2015) emphasize the utility of Unity in creating VR experiences. The wide range of assets, development tools, and support for various VR platforms has made Unity a go-to choice for VR development.

- **Escape Room Games in VR:** The concept of escape room games, which require players to solve puzzles and challenges to escape from a confined space, has gained immense popularity in the world of gaming. Researchers like Herrington et al. (2015) have examined the potential of escape room games for educational purposes, including problem-solving and critical thinking skills. The fusion of VR and escape room games provides an opportunity for immersive, interactive, and educational experiences.

- **Educational Benefits of VR in Gaming:** The educational value of VR gaming has been well-documented. Literature, including studies by Anderson et al. (2017) and de Melo et al. (2017), highlights the potential of VR gaming to enhance learning outcomes. It encourages active participation and engagement, making it a powerful tool for conveying complex concepts and fostering problem-solving skills.

- **Interaction Design in VR:** The success of VR experiences heavily relies on the quality of interaction design. Notable works by LaViola Jr. (2000) and Bowman et al. (2005) delve into the principles of interaction design in VR, including the importance of intuitive controls and user interfaces to create a seamless and immersive experience.

- **Scripting in VR Games:** Scripting is a critical component of VR game development. Literature on scripting in VR games, such as the works of Chen et al. (2015) and Stuerzlinger et al. (2006), outlines the challenges and opportunities in creating interactive and responsive VR content. Efficient scripting ensures that the virtual environment responds to user actions, enhancing immersion and engagement.

The literature review provides a backdrop for "The Haunted Jaunt Escape Room Game" in Unity, which combines VR technology, Unity development, and escape room gameplay. This project leverages the immersive and educational potential of VR to create a game that engages players while fostering problem-solving skills and critical thinking. The subsequent sections of this project report will delve into the development process, challenges faced, solutions devised, and the successful outcomes achieved, building upon the foundation laid by this literature review.

Z

2.2 Methodology:-

Steps to develop haunted jaunt escape room game:-

Fig 1. Flowchart of haunted jaunt escape room game

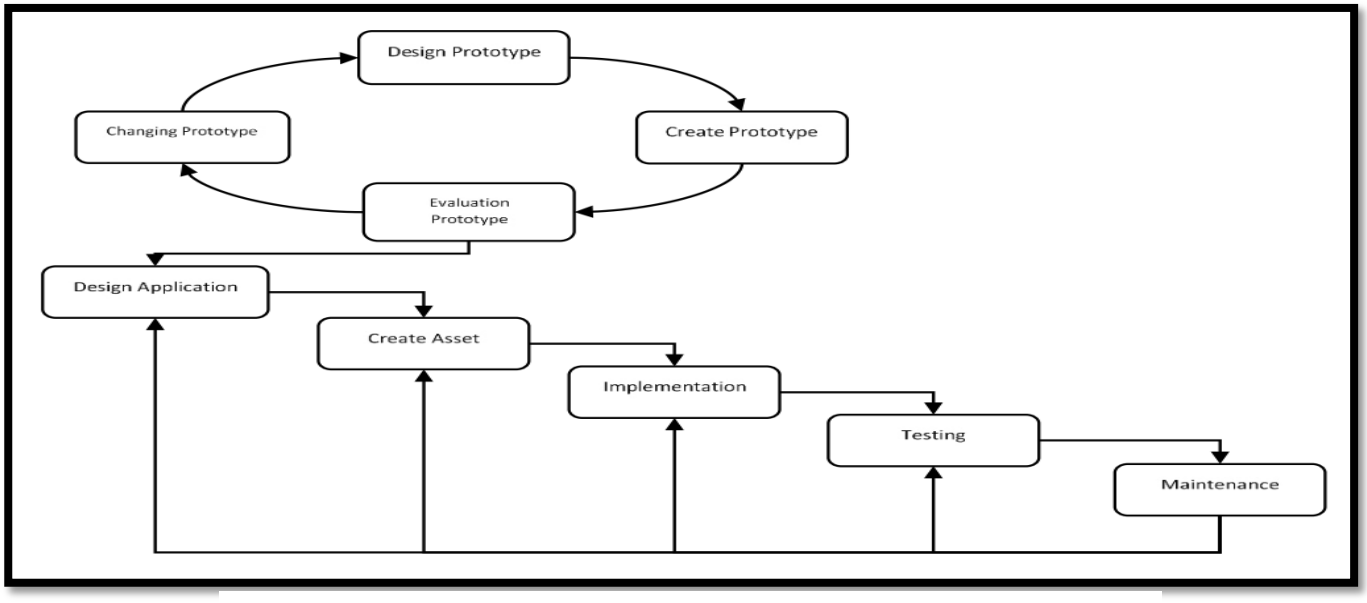
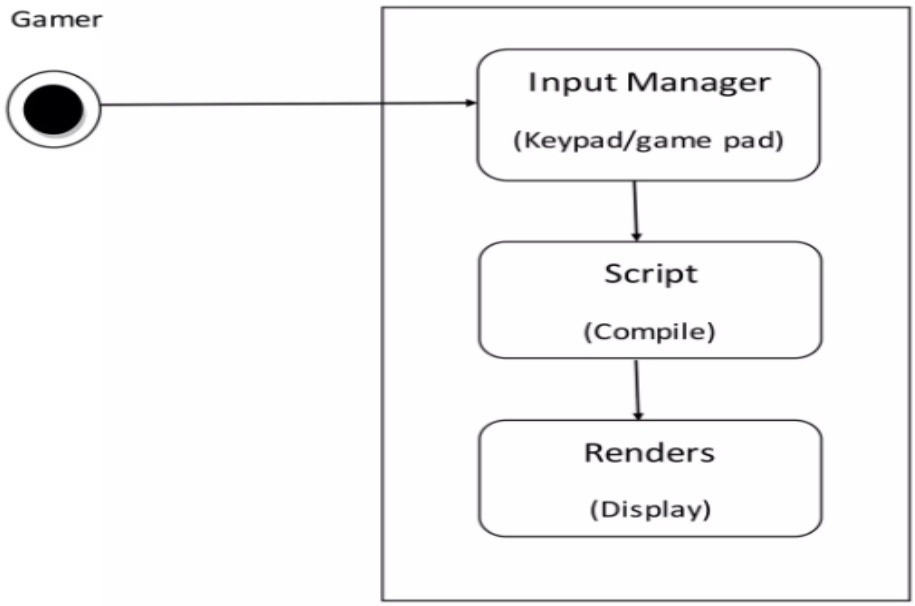


Fig 2. Architecture Diagram

Z

2.2.1 System Environments:-



Gamer can interact with system by giving input (press key to start game) to the system. System give those inputs to script, if any change occur (if the value is changed) this object send to renders to display the things (a character can change its place).

Level – 0	Level – 1	Level – 2
Game (Ghost in the Town)	Play	New Game
		Resume Game
		Select Level
		Exit Game
	Options	Show Control
		Change Configuration (Graphics)
		Change Sound/ Music Volume
	Score Board	View Scores
		Reset Score Board
	Story	View Story
	Quit	-

Fig. Use Case Scenario

Z

DFD Diagrams

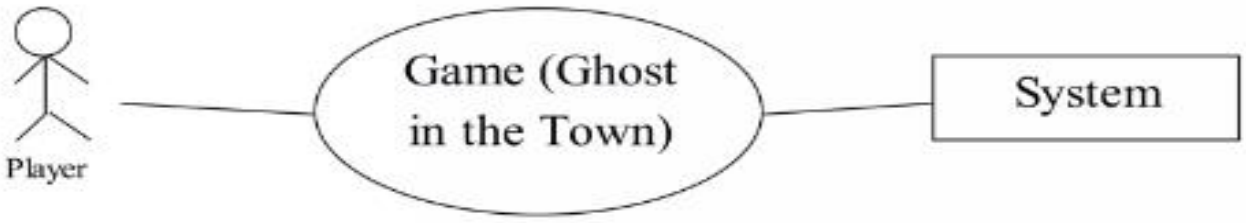


Fig 1: Level 0 for Game

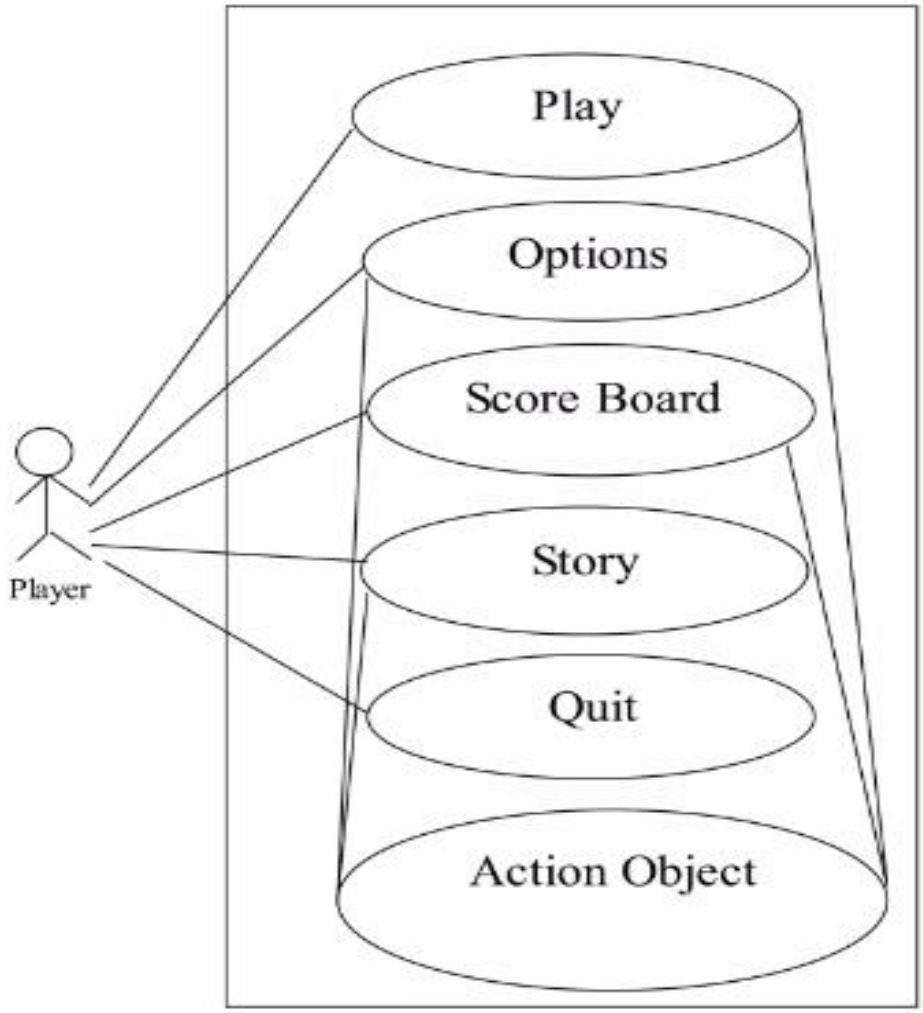


Fig 2: Level 1 for Game UX

Z

This Diagram of Level 2.1(Fig 3) leads us to the "Play" module of the use cases. These use case descriptions are given here -

Play

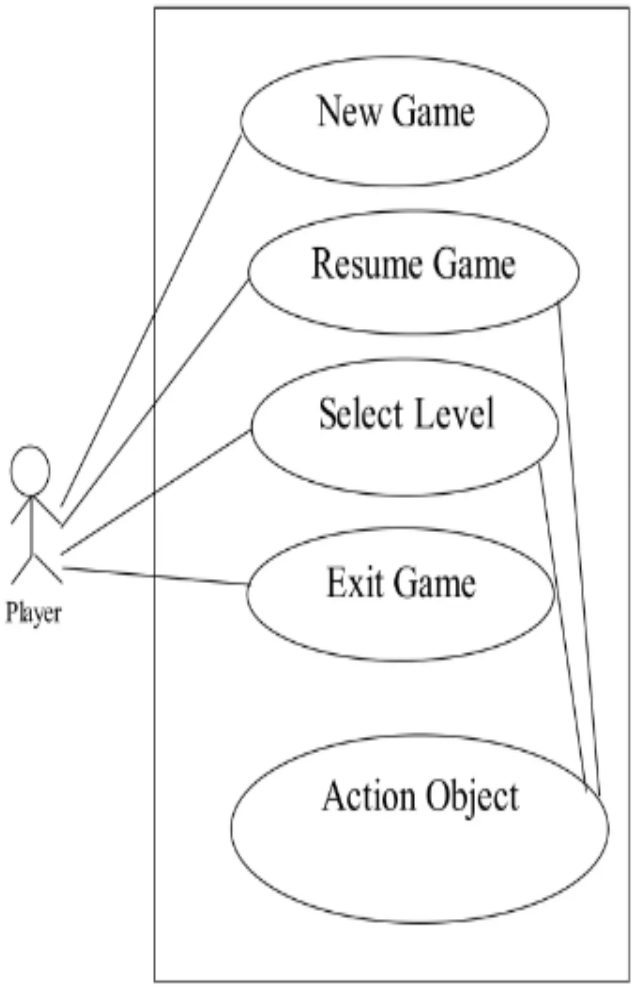


Fig 3: Level 2.1(Play) for Game UX

i. Use case: New Game

Primary Actors: Any one playing the game

Goal in context: To start a new game

Precondition:

1. System supports the game configuration

2. The file has been triggered to run and the game screen has appeared

Triggers: The player needs to start a new game

Scenario:

1. Go to the main menu of the game

2. Click new game button

3. New game is loaded on system

Exception: Game crushed

Priority: Essential, must be implemented

When Available: First increment

ii. Use case: Resume Game

Primary Actors: Any one playing the game

Goal in context: To resume game from previous play

Precondition:

1. Game was played before

2. Game supports to have a checkpoint to start from

Triggers: Need to resume game

Scenario:

1. Go to the main menu of the game

2. Click the resume game button

3. Game is loaded from the last checkpoint

Exception:

1. Level cannot be loaded

2. Game crushed

Priority: Essential, must be implemented

When Available: First increment

Z

iii. Use case: Select Level

Primary Actors: Any one playing the game

Goal in context: To load the game from a required level

Precondition:

1. Required level has been unlocked

2. Game supports loading levels

Triggers: Need to load a level

Scenario:

1. Go to the main menu

2. Click the select level button

3. Select a level

4. The level is loaded for play

Exception: Level cannot be loaded

Priority: Essential, must be implemented

When Available: First increment

iv. Use case: Select Level

Primary Actors: Any one playing the game

Goal in context: To load the game from a required level

Precondition:
3. Required level has been unlocked
4. Game supports loading levels

Triggers: Need to load a level

Scenario:
5. Go to the main menu
6. Click the select level button
7. Select a level
8. The level is loaded for play

Exception: Level cannot be loaded

Priority: Essential, must be implemented
When Available: First increment

V. Use case: Exit Game

Primary Actors: Any one playing the game

Goal in context: To exit from the game level

Precondition: A game level is being played

Triggers: Player needs to exit from the game level

Scenario:

1. Press game pause

2. When Pause Menu appears, click Return to Menu button

3. Game is exited and Title screen appears

Priority: Essential, must be implemented

When Available: First increment

Z

This Diagram of Level 2.2(Fig 4) connects with the "Option" module of the use cases. These use case descriptions are given here -

Options

i.Use case: Show Controls

Primary Actors: Any one playing the game

Goal in context: To know the controls of playing the game

Precondition: Game provides control information

Triggers: Player needs to know the controls to play the game

Scenario:

1. Go to the main menu

2. Click the Options button

3. When Option menu appears click the show control button

4. Game controls are being showed

Exception: No control information

Priority: Essential, must be implemented

When Available: First increment

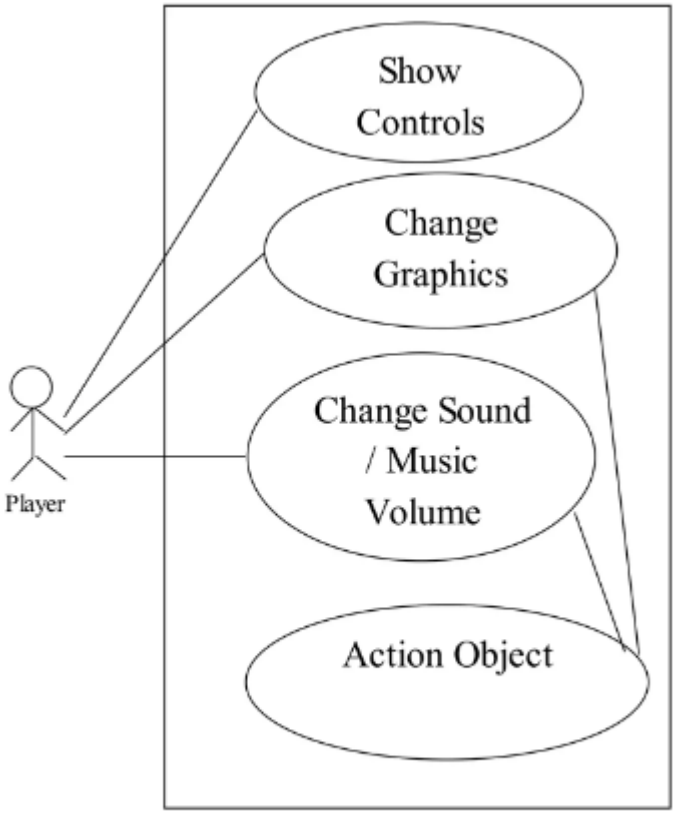


Fig 4: Level 2.2(Options) for Game UX

ii. Use case: Change Graphics Configuration

Primary Actors: Any one playing the game

Goal in context: To change the graphics configuration of the game

Precondition:

1. Player is allowed to change configuration

Triggers: Player has a need to configure graphics

Scenario:

1. Go to the main menu

2. Click on Options button

3. Click on Graphics slider and set the required value

4. Game is updated

Exception: System doesn't support given graphics configuration

Priority: Expected

When Available: Second increment

Use case: Change Sound/Music Volume
Primary Actors: Any one playing the game

Goal in context: To change the sound or music volume

Precondition: Player is allowed to change volume of game

Triggers: Player has a need to change volume of the game

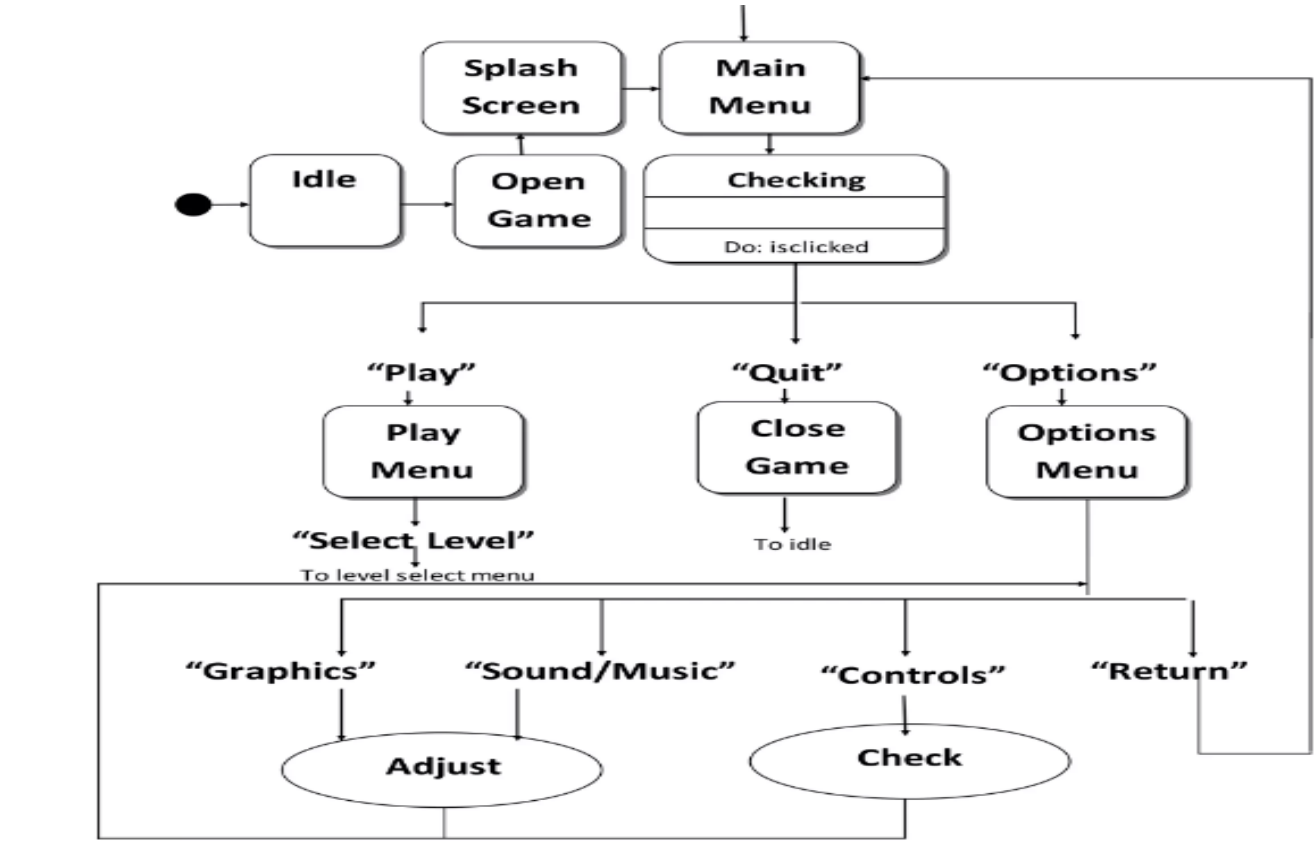
Scenario:
1. Go to the main menu
2. Click on Options button
3. Click on Music/ Sound Slider and change the value
4. Music or Sound Volume is changed

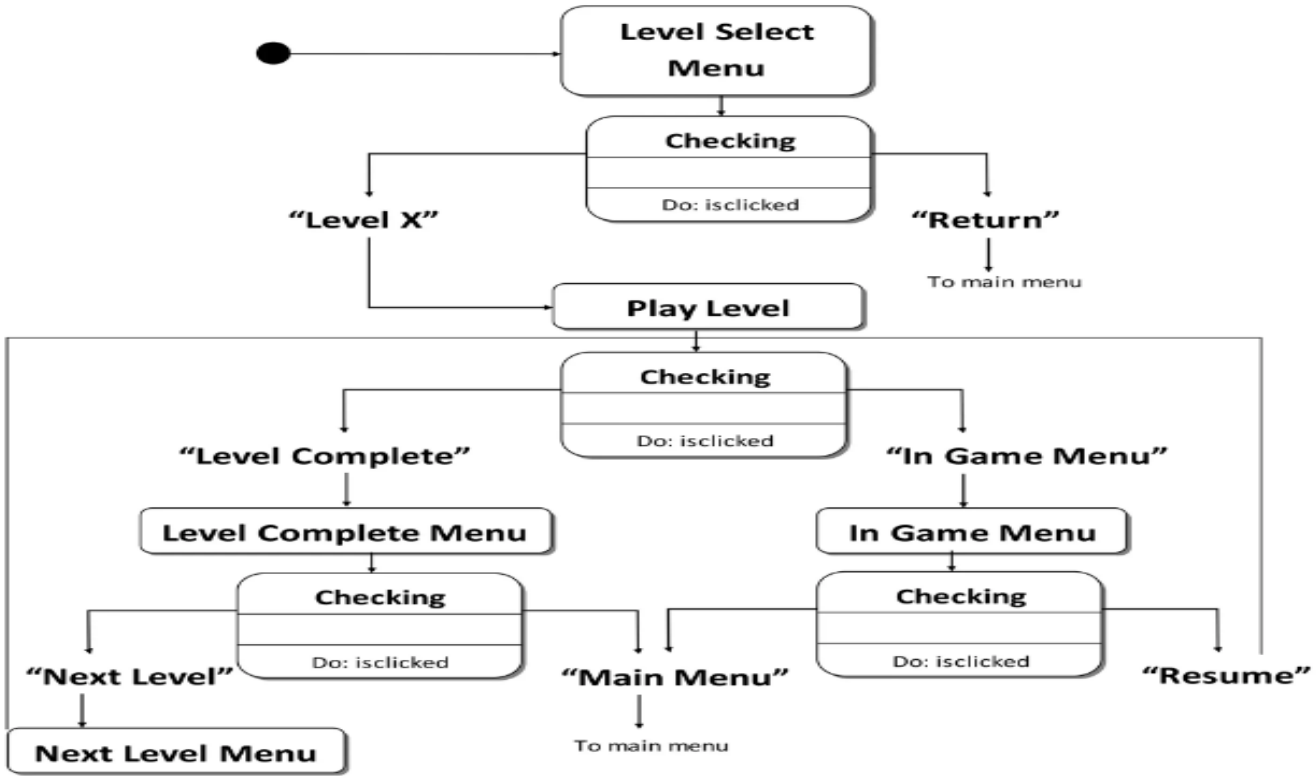
Exception: System is in mute mode, cannot increase volume

Priority: Expected
When Available: Second increment

Z

2.2.2 State Diagram





Play Level State Diagram

Z

2.3 Software Requirements

The development of "The Haunted Jaunt Escape Room Game" in Unity necessitated a specific set of software tools and platforms. The following software requirements were essential for the successful creation of this virtual reality (VR) project:

1. **Unity (Version X.X.X)**: Unity, a powerful and versatile game development engine, served as the core software platform for creating "The Haunted Jaunt Escape Room Game."

2. **Oculus Integration Package (Version X.X.X)**: Unity, a powerful and versatile game development engine, served as the core software platform for creating "The Haunted Jaunt Escape Room Game."

3. **3D Modeling and Animation Software (e.g., Blender, Maya, 3ds Max)**: 3D modeling and animation software were utilized for creating custom 3D models and animations for the game.

4. **Adobe Photoshop (Version X.X)**: Adobe Photoshop was used for texture and image editing, creating assets such as textures, UI elements, and promotional materials.

5. **Sound Editing Software (e.g., Audacity, Adobe Audition)**: Adobe Photoshop was used for texture and image editing, creating assets such as textures, UI elements, and promotional materials.**Version Control System (e.g., Git)**:.

6. **Text Editors and Integrated Development Environment (IDE)**: A version control system was utilized to manage and track changes in the project's source code and assets.

7. **Documentation Tools (e.g., Microsoft Word, Google Docs)**: Text editors and IDEs, such as Visual Studio, were used for writing, editing, and debugging scripts and code.

Z

CHAPTER 3
PLANNING AND FORMULATION

Z

3. Planning and Formulation

3.1 Project Development Model

1. Set up your Unity project

To set up your Unity project:
1. Install **Unity 2020.3**, if you haven't already done so.
2. Create a new Unity project using one of the **Core Templates** (2D or 3D). It doesn't matter which Core Template you select; the package that you import will override the default Template settings.
3. Go to the **3D Beginner: Tutorial Resources** in the Unity Asset Store
4. Download and import the assets into your Unity project.

2. Save the Scene

Now it's time to add GameObjects to your own Scene!
When a new Project is created, Unity automatically creates an empty Scene called "Untitled".

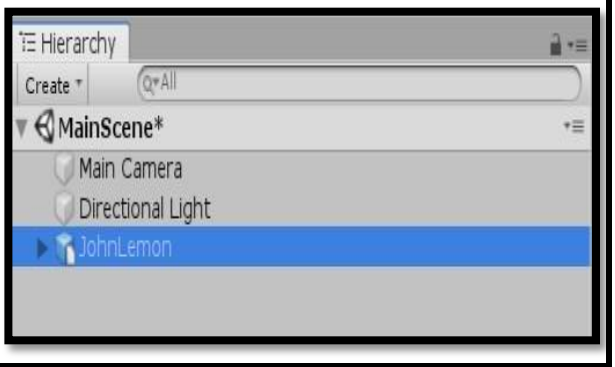
Go to **File > Save** or press Ctrl/Cmd + S, then select where you want to save the file. There is already a folder called Scenes, so let's save this Scene in that folder as "**MainScene**".

3. Add a Character Model

Let's add John Lemon, the player character:
1. In the Project window, go to the Assets > Models > Characters folder and find the model called JohnLemon.
2. Drag the model from the Project window into the Scene view. This enables you to choose exactly where you want to place the model. (You can also drag models into the Hierarchy to create GameObjects at the default position.)
3. With your cursor over the Scene view, press F to focus. You should now see the player's character in the scene! You can also find information about the GameObject you just instantiated in the Inspector. Currently it has two components:

- A Transform component, which means it has a location and size in the Scene

An Animator component, which means that it can be animated.

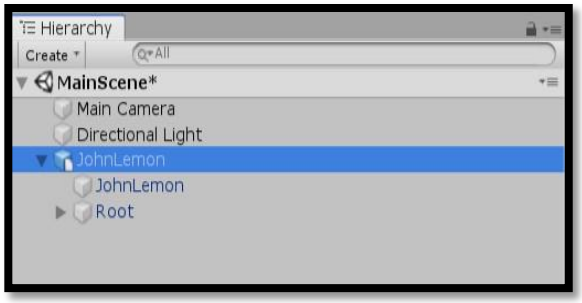


4. In the **Hierarchy window**, find the JohnLemon GameObject.

Select image to expand
There is an arrow next to the name of the GameObject — this means that a GameObject has children.

Z

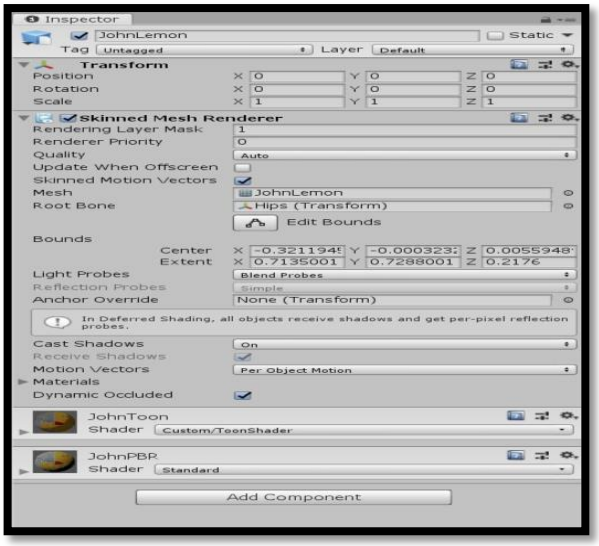
5. Click on the arrow to expand JohnLemon and see its children.

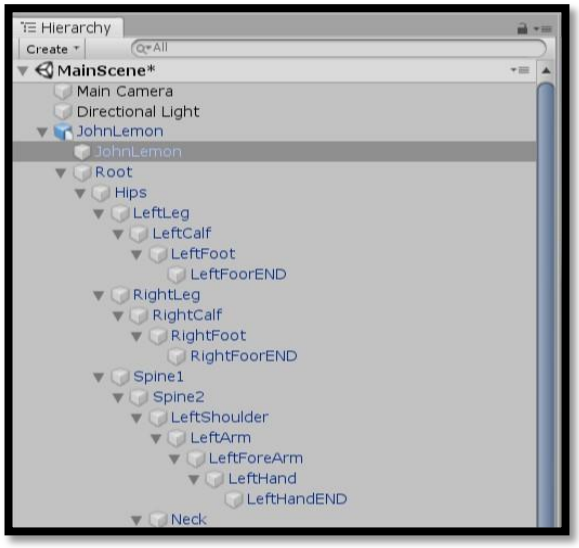


This GameObject has two children: another GameObject called JohnLemon and one called Root. Select the child GameObject called **JohnLemon**.

Select image to expand
This GameObject has a component called a **Skinned Mesh Renderer**.

6. In the Hierarchy, select the **Root** GameObject. Hold Alt (Windows) or Option (macOS) and click the arrow to the left of its name to expand all its child GameObjects.





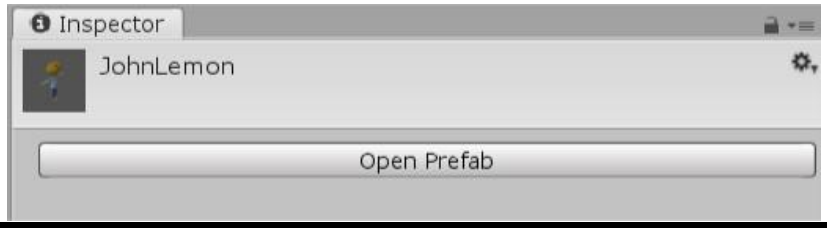
Click the **Play** button in the toolbar to enter Play Mode.

4. Turn the Character into a Prefab

. Drag the GameObject from the Hierarchy into the **Assets > Prefabs** folder in the **Project window**. A dialogue box will appear asking if you want to make an Original Prefab or a Prefab Variant — select **Original Prefab**.

2. Now the JohnLemon Prefab has been created, any changes you make to that Prefab will be reflected on the instance of the JohnLemon Prefab in the Scene. In order to make changes to a Prefab, you will need to open the Prefab for editing in Prefab Mode. Before you do that, save the Scene by pressing Ctrl + S (Windows) or Cmd + S (macOS). Now you can open the JohnLemon Prefab!

3. In the Inspector window, click the **Open Prefab** button. Unity Editor is now in Prefab Mode. This mode takes you out of the Scene you were editing before, and puts you in a temporary Scene with just the Prefab. The Scene view has changed slightly: at the top there is a new bar which says **Scenes | JohnLemon** on the left and has a checkbox labeled Auto Save on the right.

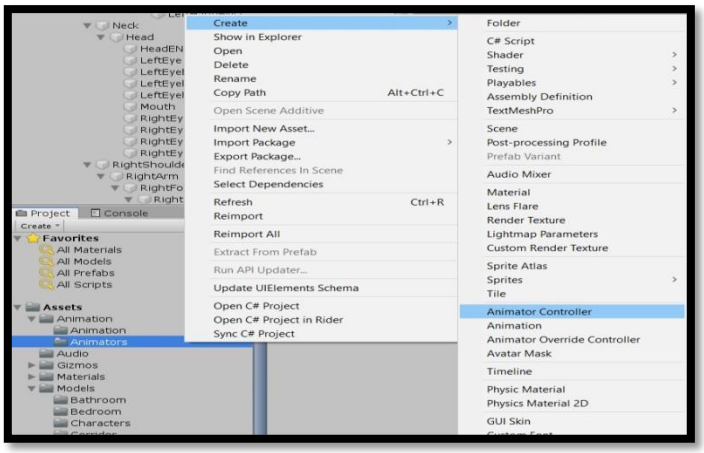


Z

5. Animate your character

Select the JohnLemon GameObject and take a look at its Animator component in the Inspector. The first property is called Controller. This takes a reference to a type of Asset called an Animator Controller, which you're going to use to get JohnLemon moving.
Animator Controllers contain a state machine which determines what animation the Animator component should be setting for its hierarchy at any given time. This animation is based on animation clips which have been set up on the Animator Controller.

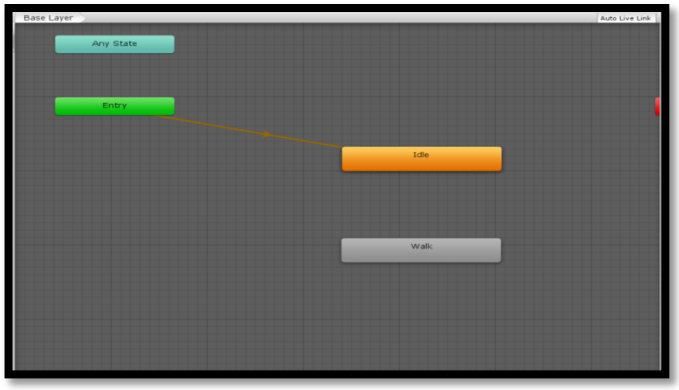
6. Create the Animator Controller



1. In the Project Window, find the Assets > Animation > Animators folder. Right click on it and select Create > Animator Controller.
2. Name the Animator Controller "JohnLemon", then double click on it to open it for editing in the Animator window. The Animator window has two main sections:

- A panel for editing Animator Layers and Animator Parameters on the left
- An area which displays the state machine itself on the right

7. Set Up the Animations



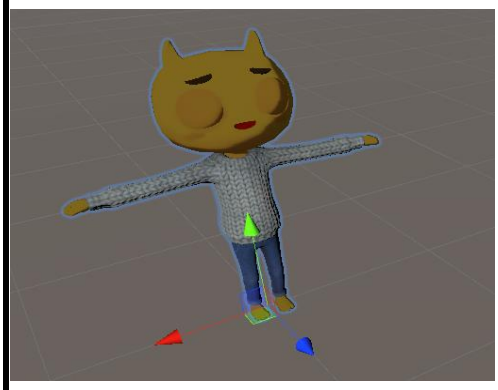
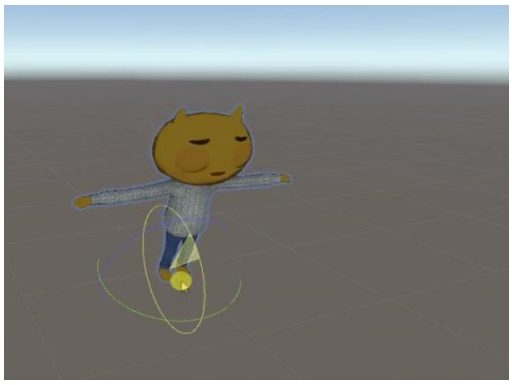
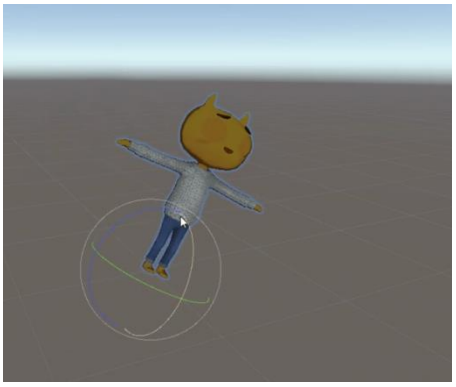
1. In the Project window, go to the Assets > Animation > Animation window.
2. You should see five models, two of which start with John@. This naming convention lets you know that these are animations for JohnLemon.
3. In order to use those animations in your Animator Controller, drag them from the Project window to the Animator window. Start with **Id**
4. Animations exist in an Animator Controller in Animator States. When you dragged in the Idle and Walk animations, the Animator Controller created two states containing them and named them after the animations.

8. Positions and Rotation

1. Select the JohnLemon GameObject, and then look at the Scene view**.**

You will see that:

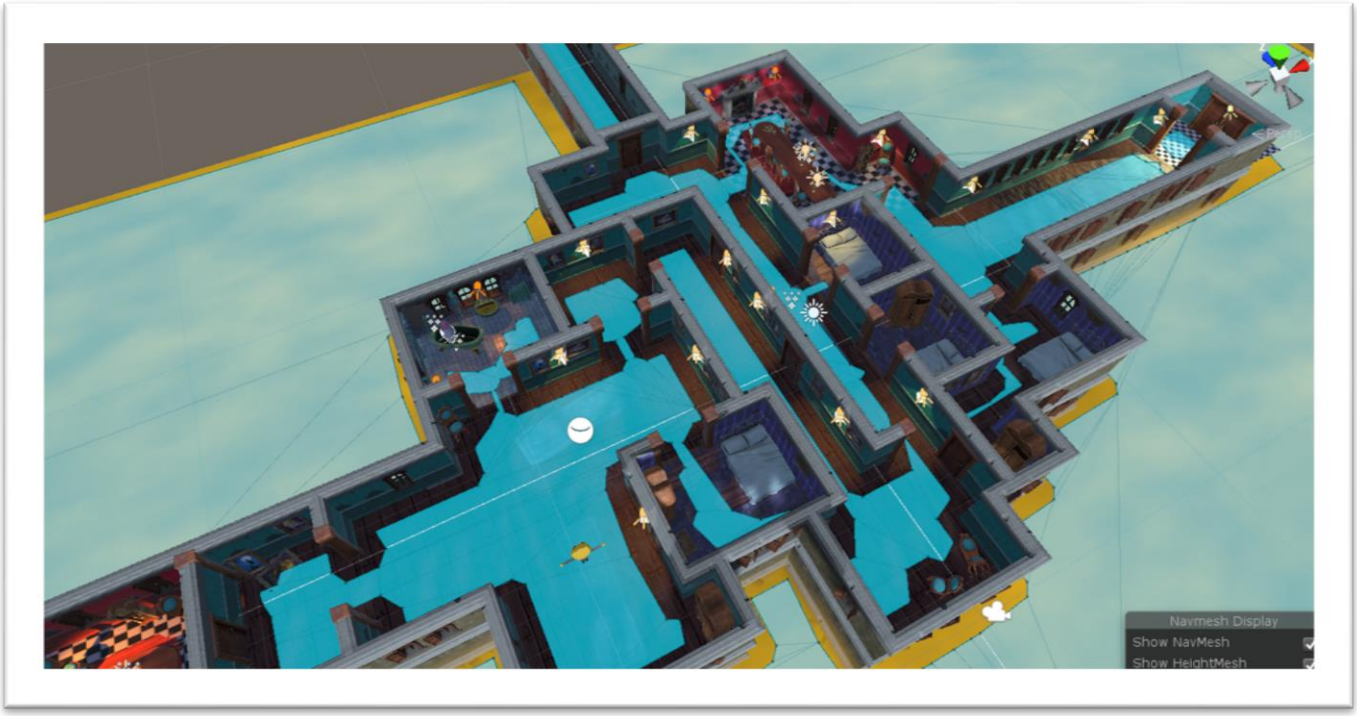
- the x-axis (red) is pointing to the character's **right**
- the y-axis (green) is pointing **upward**
- the z-axis (blue) is pointing **forward**

Z

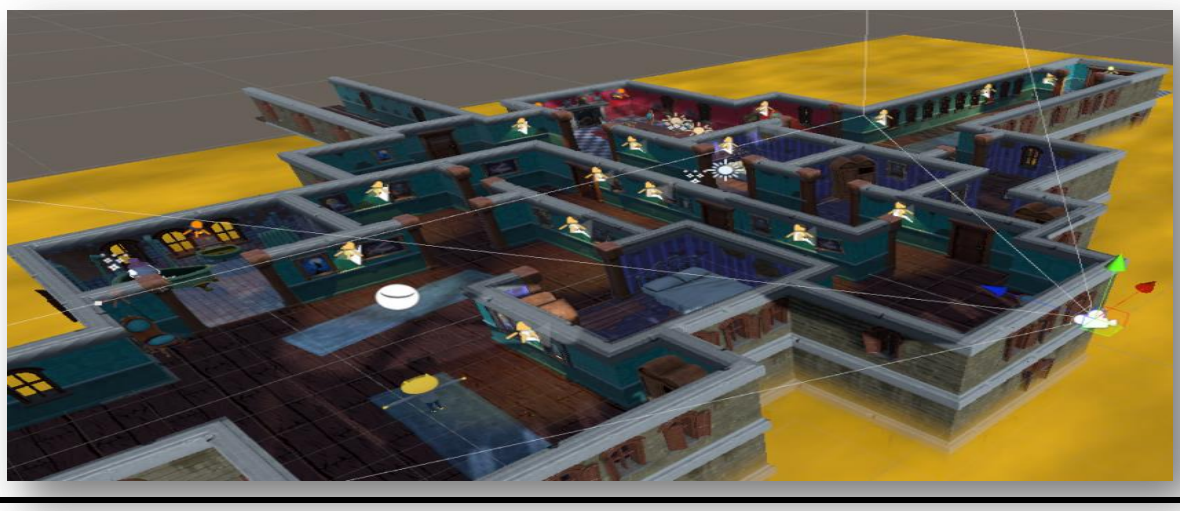
9. Create the NavMesh

1. In the Menu bar, go to Window > AI > Navigation to open the Navigation window. The window should dock itself with the Inspector window. If it doesn't, drag and dock it there.
2. There are 4 tabs at the top of the Navigation window: Agents, Areas, Bake and Object.**Select the Bake tab.**
3. The Bake settings control the details of how the NavMesh will be constructed. The first settings refer to the agents that will traverse (the ghosts that will move around) the NavMesh — the NavMesh Agents. They specifically refer to the size of the agents and the terrain they can move across.
Select the **Bake** tab.
The NavMesh will only be visible when the Navigation window is open and active (if you switch to the Inspector tab, the mesh will disappear from the Scene view). Don't worry — even when you can't see the NavMesh, it's still there!



10. Explore the Camera Component

To view the Scene, a GameObject in the Scene must have a Camera component. When a new Scene is created, a GameObject is added called Main Camera which has a Camera component.



Z

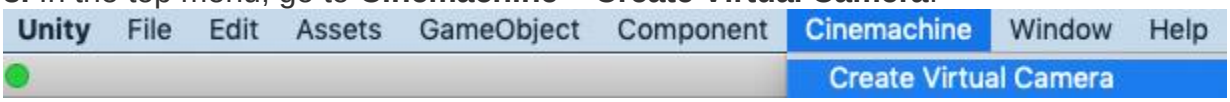
11. Set up a Virtual Camera using Cinemachine

n the Project window, go to **Assets > Scenes** and double click **MainScene**.

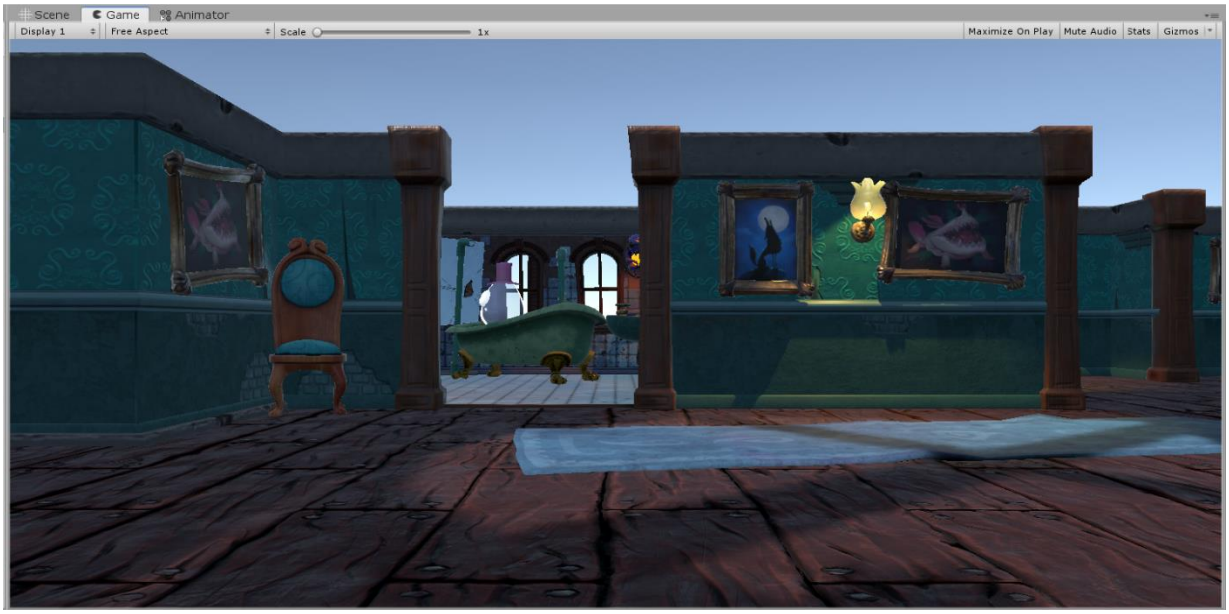
1. In the Hierarchy, select the JohnLemon GameObject.

2. Move your cursor over the Scene window and press the **F** key.

3. In the top menu, go to **Cinemachine > Create Virtual Camera**.



This will create a new GameObject in the scene called **CM vcam1**:



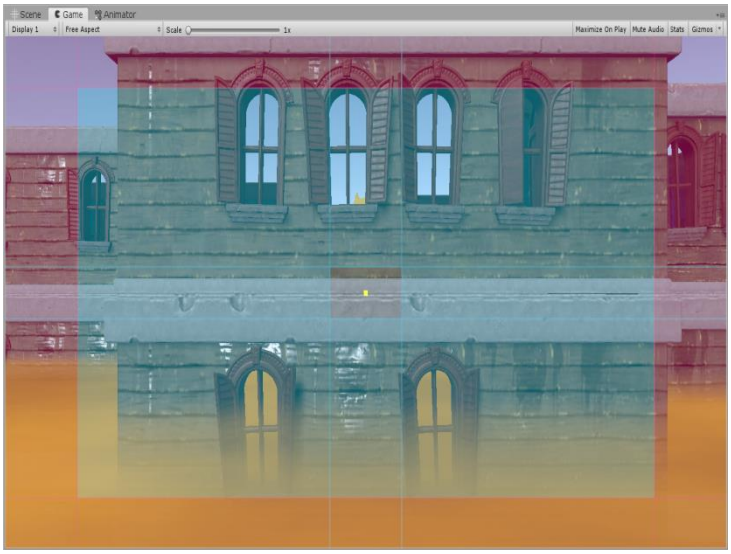
12. Change the Cinemachine Virtual Camera Component Settings

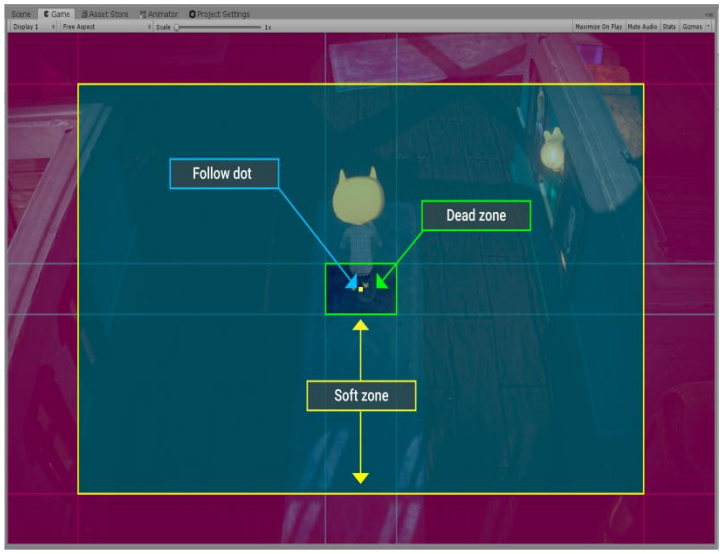
1.In the Hierarchy, select CM vcam1.

2. In the Aim section, change the drop-down menu at the top right from Composer to Do Nothing.

3. Drag and drop the JohnLemon GameObject from the Hierarchy window onto the Follow property of the Cinemachine Virtual Camera Component. This will change the Follow setting to reference JohnLemon's Transform.

4. In the Body section, change the drop-down at the top right of the section from Transposer to Framing Transposer.





Z

13. Setting up the Gargoyle Prefab

In the Project window, navigate to the Assets > Models folder and find the Gargoyle Asset.

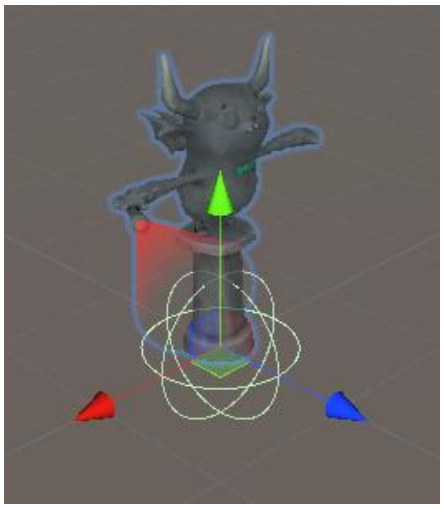
2. Drag the Asset from the Project window into the Hierarchy window, to create an instance of the model.

3. This haunted house needs multiple gargoyles. Remember, a model is read-only — to make edits you will need to create a Prefab. Drag the Gargoyle GameObject from the Hierarchy window into the **Assets > Prefabs** folder in the Project window. When the Create Prefab dialogue box appears, select **Original Prefab**.

4. Now that you have a Gargoyle Prefab, you can open it for editing. This time you're going to use a shortcut! In the Hierarchy, click the **arrow to the right of the Gargoyle GameObject**:

14. Animate the Gargoyle

1. In the Project window, go to the **Assets > Animation > Animators** folder and right-click on it.
2. In the context menu, select **Create > Animator Controller.** Name the new Animation Controller "**Gargoyle**".
3. Double click on **Gargoyle** to open the Animator window.
4. In the Project window, navigate to the **Assets > Animation > Animations**.
5. Expand the **Gargoyle@Idle** Model Asset.
6. In the Project window, go to the **Assets > Animation > Animators** folder and right-click on it.
7. In the context menu, select **Create > Animator Controller.** Name the new Animation Controller "**Gargoyle**".
8. Double click on **Gargoyle** to open the Animator window.
9. In the Project window, navigate to the **Assets > Animation > Animations**.
10. Expand the **Gargoyle@Idle** Model Asset.



15. Write a Custom Observer Script

1. In the Project window, go to **Assets > Scripts**.
2. Right-click on the Scripts folder and select **Create > C# Script**. Name the new script "**Observer**". This name describes what the script will do: observe the player's character and cause the game to restart if it's spotted. You'll be able to reuse this script for other enemies too, so it makes sense not to link its name to the Gargoyle.
3. Drag the Observer script Asset from the Scripts folder onto the **PointOfView** GameObject in the Hierarchy window to add it as a component.
4. Double click on the script Asset to open it for editing.

Your Observer script is nearly finished! Let's review what you have so far

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Observer : MonoBehaviour
{
    public Transform player;
    public GameEnding gameEnding;
```

Z

```
bool m_IsPlayerInRange;

void OnTriggerEnter (Collider other)
{
   if (other.transform == player)
   {
      m_IsPlayerInRange = true;
   }
}
void OnTriggerExit (Collider other)
{
   if (other.transform == player)
   {
      m_IsPlayerInRange = false;
   }
}
```

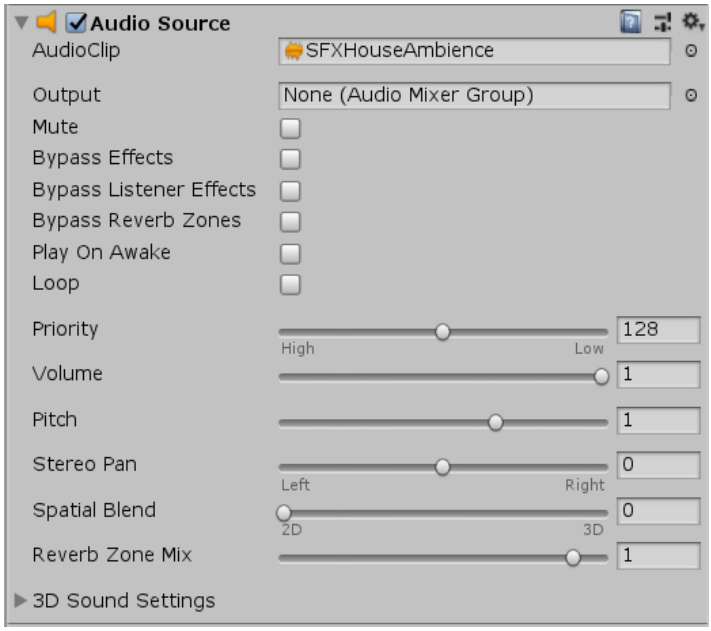
```
void Update ()
    {
       if (m_IsPlayerInRange)
       {
          Vector3 direction = player.position -
transform.position + Vector3.up;
          Ray ray = new Ray(transform.position,
direction);
          RaycastHit raycastHit;

          if(Physics.Raycast(ray, out raycastHit))
          {
             if (raycastHit.collider.transform ==
player)
             {

             }
          }
       }
```

16. Create Audio Sources for your Game

1. In the Hierarchy window, click on the Create menu and select **Create Empty**. Rename the GameObject "**Ambient**".

2. In the Inspector, set the position of Ambient to **(0, 0, 0)**. The position of the GameObject doesn't technically matter, since the volume of the audio will be the same wherever it is. However, it is always helpful to keep your GameObjects organised positionally in case this matters later.

3. Next, you need to add an Audio Source component. Normally you would do this with the Add Component button, but when you want to assign an Audio Clip as well there is a shortcut you can use. In the Project window, go to **Assets > Audio**. Drag the **SFXHouseAmbience** Audio Clip from the Project window onto the **Inspector window** to create an Audio Source component and automatically assign it as the AudioClip.



17. Build & Run:-

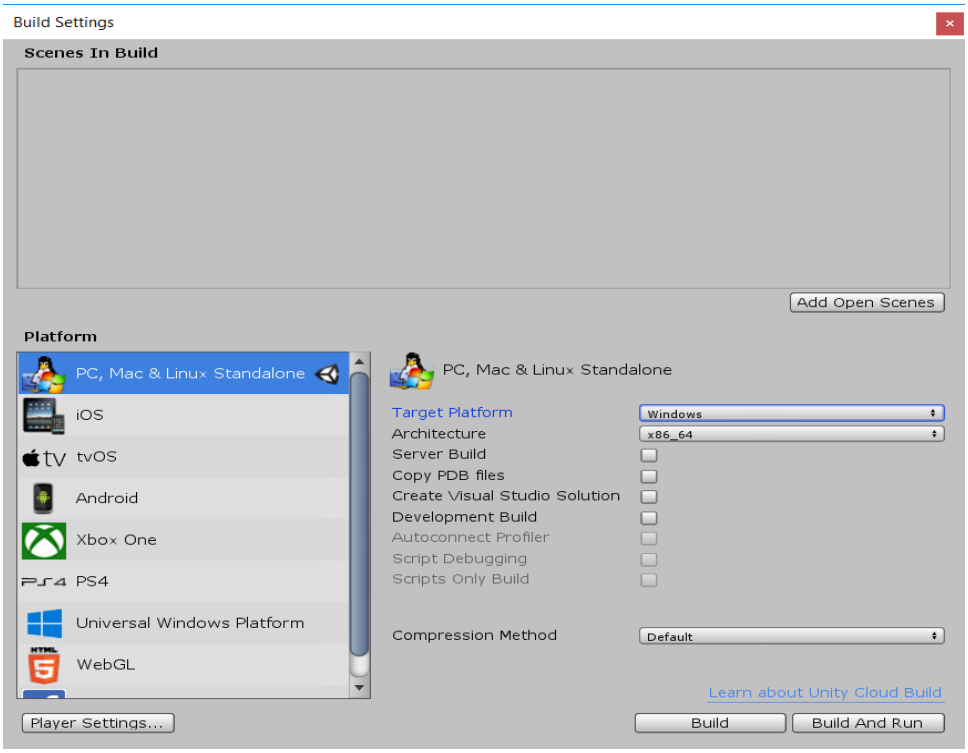
In the top menu, select File > Build Settings.

2. The Scenes In Build section at the top lists all the Scenes that will be included in your game. You can have Scenes in your Project that you only use to test features or to debug, so Unity needs to know which ones to include in the final product.

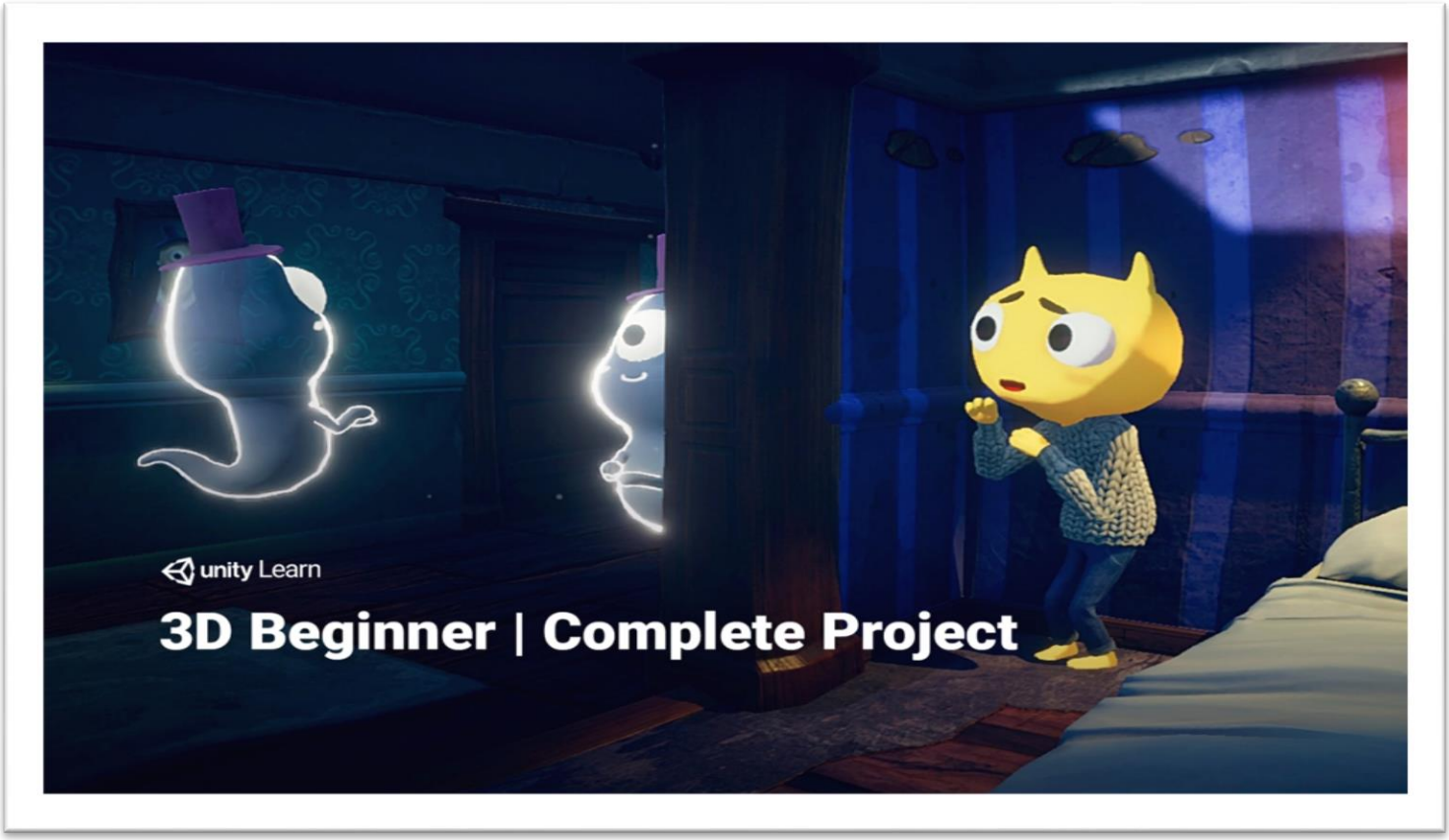
If your MainScene is still open, click Add Open Scenes to add it to the list. Alternatively, you can drag and drop your Scenes from the Project Window to the Scenes In Build section of the Build Settings window.

Z

3. The Platform section at the bottom left allows you to choose which platforms you want your game to run on. By default, the Editor only supports the platform that it is installed on.

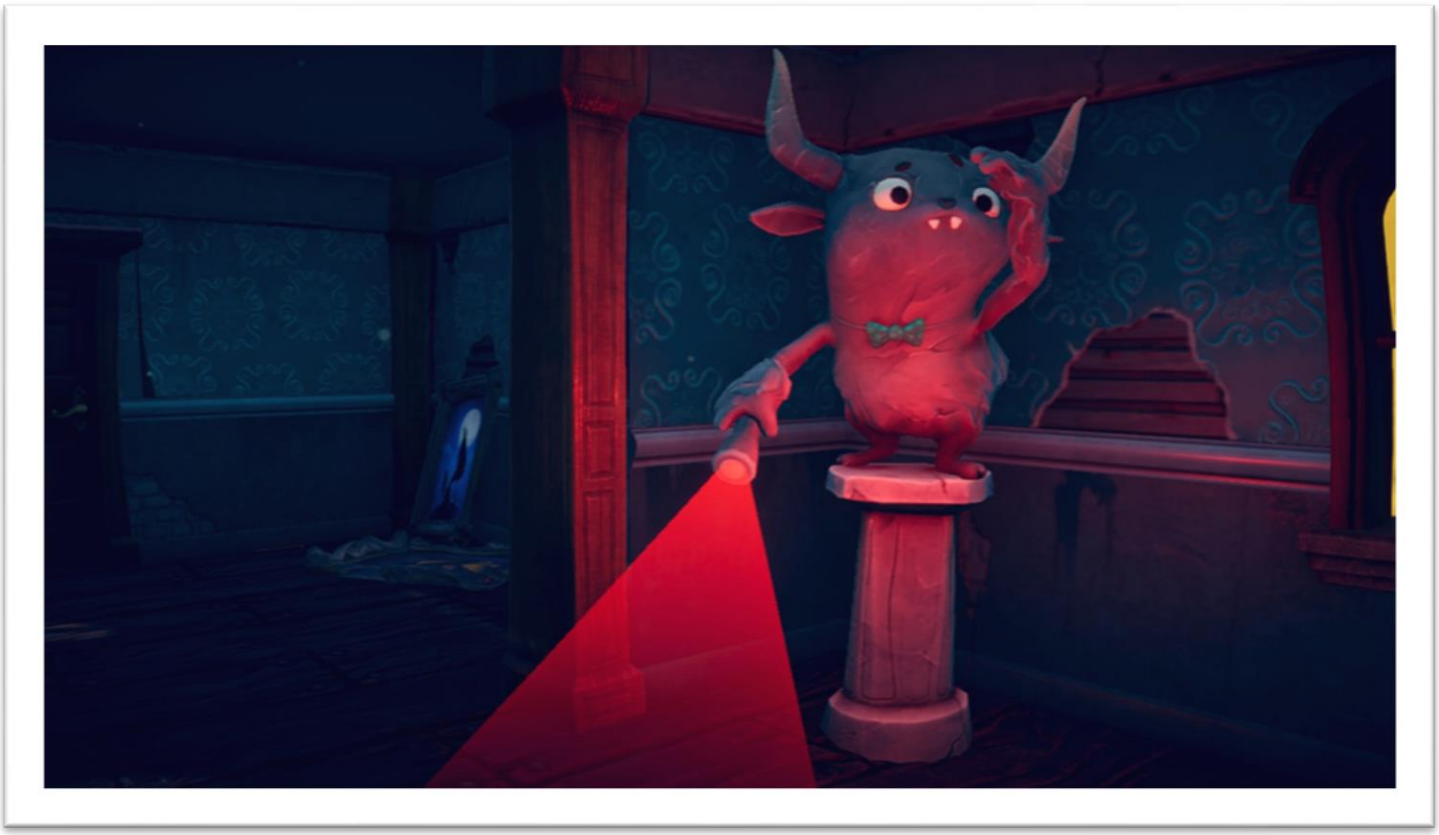


OUTPUTS;-



SCENE 1

Z



Character Monster SCENE 3

SCENE 4

Z

SCENE 5

Dining Room Scene

Z

Bathroom Scene

Corridor Scene End State

Z

Bedroom Scene

Exterior Walls Scene

Z

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

Z

CONCLUSION:-

In conclusion, the development of "The Haunted Jaunt Escape Room Game" in Unity stands as a testament to the creative and educational possibilities offered by virtual reality (VR) technology. This project successfully merged entertainment and learning, providing players with an immersive and intellectually stimulating experience. Throughout this journey, we've not only achieved our goals in developing an engaging escape room but also acquired valuable VR development skills, empowering us for future endeavors in the dynamic world of virtual reality.

Our interactions with the VR and gaming community have been enriching, and the feedback received has greatly contributed to the project's refinement, ensuring it can reach a broader audience on various VR platforms. While we encountered and overcame challenges in scripting complex VR interactions and optimizing assets, each obstacle served as a stepping stone to enhancing our skills and problem-solving capabilities.

As we look to the future, this project is a springboard for us and, we hope, for others in the VR development landscape. The project's success exemplifies the vast potential of VR in education and entertainment and its capacity to provide engaging, immersive, and educational experiences. Our journey in VR continues with enthusiasm, driven by the belief that the technology offers limitless opportunities for creativity and learning.

Future Scope:-

- **Content Expansion**:
  - *Additional Rooms and Puzzles*: Expanding the game by introducing new rooms and puzzles can provide a more extensive and varied experience for players. Each new room can present unique challenges and educational opportunities.
  - *Narrative Development*: Building a more intricate and engaging narrative can enhance player engagement and immersion. A compelling storyline can guide players through the escape room, creating a more immersive experience.
- **Multiplayer Integration**:
  - *Multiplayer Mode*: Implementing a multiplayer mode can transform the escape room into a collaborative experience. This can be valuable for educational settings, encouraging teamwork and communication among participants.
  - *Competitive Challenges*: Introducing competitive elements can make the escape room more engaging for players. Time-based challenges or player vs. player puzzles can add a competitive dimension.
- **Educational Applications**:
  - *Customizable Learning Modules*: Adapting the game for educational use by creating customizable learning modules that align with curricular objectives. This can cater to a wide range of educational institutions and settings.
  - *Integration with E-Learning Platforms*: Linking the game to e-learning platforms, allowing educators to track progress and assess students' performance in the virtual escape room.
- **Cross-Platform Compatibility**:
  - *Expanding Compatibility*: Ensuring compatibility with an even broader range of VR platforms and devices, reaching a wider audience.
- **User-Generated Content**:
  - *Content Creation Tools*: Developing user-friendly tools that allow players to create and share their escape room scenarios. This can enhance the community aspect of the game and contribute to its longevity.
- **Integration of Emerging Technologies**:
  - *Incorporating Augmented Reality (AR)*: Combining VR with AR elements to create a hybrid reality experience, further blurring the lines between the virtual and physical worlds.

Z

Acknowledgement

As every project is ever complete with the guidance of experts. So we would like to take this opportunity to thank all those individuals who have contributed in visualizing this project.

We express our deepest gratitude to our project guide Prof----------(CSE(AIML) Department, Smt. Indira Gandhi College of Engineering, University of Mumbai) for her valuable guidance, moral support and devotion bestowed on us throughout our work.

We would also take this opportunity to thank our project coordinator **Prof. Krishna Salgaonkar** for her guidance in selecting this project and also for providing us all the details on proper presentation of this project.

We extend our sincere appreciation to our entire professors from Smt. Indira Gandhi College of Engineering for their valuable inside and tip during the designing the project. Their contributions have been valuable in many ways that we find it difficult to acknowledge them individually.

We are also grateful to our HOD **Prof. Sonali Deshpande** for extending his help directly and indirectly through various channels in our project.

If I can say in words I must at the outset my intimacy for receipt of affectionate care to Smt. Indira Gandhi College of Engineering for providing such a simulating atmosphere and wonderful work environment.

Z

References:-

1. Anderson, C. A., Dill, K. E., & Dill, J. C. (2000). Video games and aggressive thoughts, feelings, and behavior in the laboratory and in life. Journal of Personality and Social Psychology, 78(4), 772-790.

2. Bowman, D. A., & McMahan, R. P. (2005). Virtual reality: How much immersion is enough? Computer, 38(7), 36-43.

3. Herrington, J., Herrington, A., Mantei, J., Olney, I., & Ferry, B. (2009). Using virtual reality to support teaching and learning in history. In Hello! Where are you in the landscape of educational technology? Proceedings ASCILITE Auckland 2009 (pp. 437-446).

4. LaViola Jr, J. J. (2000). A discussion of cybersickness in virtual environments. ACM SIGCHI Bulletin, 32(1), 47-56.

5. Linowes, J. (2015). Unity Virtual Reality Projects. Packt Publishing Ltd.

6. Riva, G., Waterworth, J. A., & Waterworth, E. L. (2011). The layers of presence: A bio-cultural approach to understanding presence in natural and mediated environments. CyberPsychology & Behavior, 4(4), 402-416.

7. Slater, M., & Wilbur, S. (1997). A framework for immersive virtual environments (FIVE): Speculations on the role of presence in virtual environments. Presence: Teleoperators and Virtual Environments, 6(6), 603-616.

8. Stuerzlinger, W., Bryson, S., & Riecke, B. E. (2006). A comparison of techniques for virtual environment navigation. Proceedings of the ACM symposium on Virtual reality software and technology, 31-39.

9. Unity Technologies. (n.d.). Unity. Retrieved from https://unity.com/

10. Unity Technologies. (n.d.). Oculus Integration. Retrieved from https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022