**[name]**

**[Roll number]**

**[College name]**

**[course id]**

[Wandaring the woods based on RAD]



**[**Rapid Application Development (RAD)-based software architecture for the Wandering in the Woods game**]**

Table of Contents

[1.0 Introduction 2](#_30j0zll)

[1.1 Purpose 2](#_1fob9te)

[1.2 The AR Room Designer 2](#_3znysh7)

[2.0 Process Model 3](#_2et92p0)

[3.0 Use Cases 4](#_tyjcwt)

[3.1 Use Case 1: Create a](#_3dy6vkm) Character 4

[3.2 Use Case 2:](#_1t3h5sf) Explore the Forest 4

[3.3 Use Case 3: Battle Enemies 4](#_4d34og8)

[3.4 Use Case 4:](#_2s8eyo1) Solve Puzzles 5

[3.5 Use Case 5:](#_17dp8vu) Find Hidden Areas 5

[3.6 Use case 6:](#_3rdcrjn) Complete Game 5

[4.0 UML Model 6](#_26in1rg)

[4.1 Use Case Diagram 6](#_lnxbz9)

[4.2 Deployment Diagram 6](#_35nkun2)

[4.3 Class Diagram 7](#_1ksv4uv)

[4.4 State Diagram 10](#_44sinio)

[4.5 Activity Diagram 11](#_2jxsxqh)

[5.0 Customer Journey Map 13](#_z337ya)

[6.0 Personas 13](#_3j2qqm3)

[7.0 UI Mock-up 14](#_1y810tw)

[8.0 Testing Strategy 15](#_4i7ojhp)

[8.1 Unit Testing 15](#_1ci93xb)

[8.2 Integration Testing 16](#_2bn6wsx)

[8.3](#_1pxezwc) System Testing 17

[8.4](#_49x2ik5) User Acceptance Testing 17

9.0 Functional

10.0 Non-Functional Requirements

# Introduction

## Purpose

This document provides an overview of the Rapid Application Development (RAD)-based software architecture for the Wandering in the Woods game. quick application development (RAD) is an iterative technique that prioritizes user feedback and quick prototyping. The goal of this article is to give a thorough description of the game's architecture, user interface, and gameplay mechanics.

A mystery forest can be explored in the first-person adventure game Wandering in the Woods. Players must solve a variety of puzzles, deal with various animals, and overcome obstacles in order to advance in the game. The game can be played on several platforms, including desktops, tablets, and smartphones.

The short development cycle of the game led to the selection of the RAD technique for this project. Rapid iteration and prototyping are made possible by the RAD technique, which was crucial for this project because the developers were continuously getting user feedback.

We hope that this document will be useful for anyone who is interested in learning more about the Wandering in the Woods Game or the RAD development methodology.

Audience

This document is intended for a technical audience, such as software engineers, game developers, and students of software engineering.

Prerequisites

The reader is assumed to have a basic understanding of software engineering principles and the RAD development methodology.

## Wandering in the Woods

## 

## A mystery forest can be explored in the first-person adventure game Wandering in the Woods. Players must solve a variety of puzzles, deal with various animals, and overcome obstacles in order to advance in the game. The game can be played on a number of platforms, including desktops, tablets, and smartphones.

## A head-mounted display (HMD) and hand controllers are used to play the game. With the help of the hand controllers, the player can interact with the virtual reality environment that is projected onto their eyes by the HMD.

## The player awakens in a forest at the start of the game. The player starts out without any weapons and has no prior knowledge of the forest's inhabitants. The player will come across a variety of critters as they explore the forest, including amiable animals, frightening predators, and even mystical beings. To advance, the player will also have to solve riddles and get past obstacles.

## The game is intended to be immersive and difficult. To live in the forest and learn its secrets, the player will need to use their cunning and abilities.

## 

## The Wandering in the Woods Game's main characteristics are as follows:

## 

## **Adventure in first person:** The first-person perspective of the game gives the user the impression that they are actually in the jungle.

## 

## **Various creatures:** The game includes a range of hostile and helpful species.

## 

## A variety of difficulties and puzzles are present in the game, which players must solve in order to advance.

## 

## **Immersive experience:** The game is made to be immersive, immersing players in its environment through the use of VR technology.

## 

## Fans of the genre will enjoy The Wandering in the Woods Game, a difficult and engaging adventure game. The game may be played on many different types of devices, making it available to a large audience.

# 

# Process Model

Quick application development (RAD) is an iterative technique that prioritizes user feedback and quick prototyping.

The processes involved in creating the Wandering in the Woods Game with RAD are as follows:

**Gathering needs:** The first stage is to compile the game's requirements from the stakeholders. This entails determining the qualities the game ought to have and the intended audience.

**Making prototypes:** After the criteria have been acquired, the game's prototypes are made. As a result, early in the development phase, the developers can collect customer feedback.

**Iteration:** The prototypes are subsequently improved iteratively in response to user feedback. This procedure keeps going until the game satisfies the demands of the stakeholders.

**Deployment**: Once the game is complete, it is deployed to the users.

The Wandering in the Woods Game may be developed using the RAD paradigm since it provides for quick development and iteration. This is crucial for this project because, in order to make a fun and engaging game, the developers will require user feedback early on in the development process.

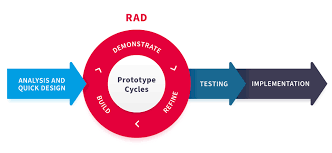
The following are some advantages of RAD for the game Wandering in the Woods:

Faster development: By enabling quick iteration and prototyping, RAD can aid in hastening the creation of the game.

Better quality: By obtaining user feedback early in the production process, RAD can help to improve the game's quality.

Lower risk: By enabling developers to modify their games in response to user feedback, RAD can help lower the risk of creating unsuccessful video games.

In general, the RAD model is a wise decision for creating the Wandering in the Woods Game since it can assist in hastening the development process, enhance the game's quality, and lessen the danger of creating an unsuccessful game.



*Figure SEQ Figure \\* ARABIC 1. The AR Room Designer evolutionary process model.*

# Use Cases

The main system requirements for the delivery of a functioning prototype were determined by the team and the client to be the following six use cases. Note that as the project develops, this basic capability may be readily expanded by adding more use cases. The name of the use case, major actors, preconditions, description, and acceptance criteria are listed for each use case.

## 

## Use Case 1:Create a Character

## 

**Primary Actor:** Player

**Preconditions:** A game has been created.

**Description:** As a player, I want to be able to create a character to play as in the game. I want to be able to choose the character's appearance, such as their gender, race, and hairstyle. I also want to be able to choose the character's starting equipment, such as their weapons and armor.

**Acceptance Criteria:** I can create a character with my desired appearance and equipment.

## Use Case 2:Explore the Forest

## 

**Primary Actor:** Player

**Preconditions:** A character has been created.

**Description:** As a player, I want to be able to explore the forest in the game. I want to be able to walk around, interact with objects, and fight enemies. I also want to be able to find hidden areas and solve puzzles.

**Acceptance Criteria:** I can explore the forest and complete the game's objectives.

## Use Case 3: . Battle Enemies

## 

**Primary Actor:** Player

**Preconditions:** A character has been created and the forest is being explored.

**Description:** As a player, I want to be able to battle enemies in the game. I want to be able to use my character's weapons and abilities to defeat the enemies. I also want to be able to collect experience points and loot from the enemies I defeat.

**Acceptance Criteria:** I can defeat enemies and level up my character.

## Use Case 4:Solve Puzzles

## 

**Primary Actor:** Player

**Preconditions:** A character has been created and the forest is being explored.

**Description:** As a player, I want to be able to solve puzzles in the game. I want to be able to use my character's intelligence and creativity to solve the puzzles. I also want to be able to find hidden clues and items that help me solve the puzzles.

**Acceptance Criteria:** I can solve puzzles and progress through the game.

## Use Case 5: . Find Hidden Areas

## 

**Primary Actor:** Player

**Preconditions:** A character has been created and the forest is being explored.

**Description:** As a player, I want to be able to find hidden areas in the game. I want to be able to explore these areas and find valuable items. I also want to be able to complete side quests and challenges in these areas.

**Acceptance Criteria:** I can find hidden areas and complete the side quests and challenges in them.

## Use case 6: Complete the Game

## 

**Primary Actor:** Player

**Preconditions:** All of the other use cases have been completed.

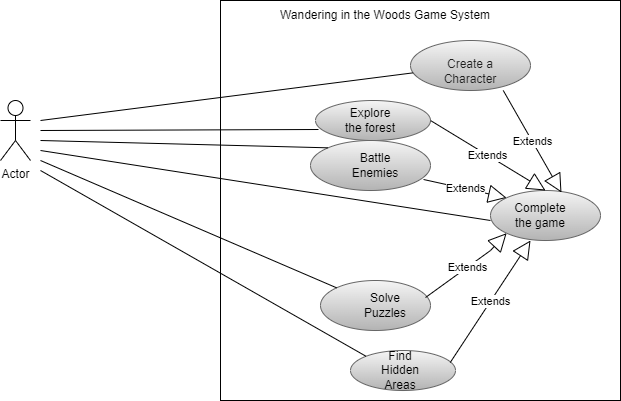
**Description:** As a player, I want to be able to complete the game. I want to defeat the final boss and save the day. I also want to be able to see the game's ending.

**Acceptance Criteria:** I can defeat the final boss and see the game's ending.

# UML Model

## Use Case Diagram

## The use case diagram for the "Wandering in Woods" game can be represented by depicting the primary actor, which is the player, and the six identified use cases as separate bubbles connected to the player. The "Create a Character" use case serves as the initial step, acting as a prerequisite for the subsequent use cases. Once a character is created, the player can "Explore the Forest," enabling interactions within the game world. This exploration can lead to "Battling Enemies," involving combat interactions and character progression. The player can also engage in "Solving Puzzles" and "Finding Hidden Areas," which enrich the gameplay experience with intellectual challenges and hidden treasures. All these experiences contribute to the player's ability to ultimately "Complete the Game," facing the final boss and achieving the game's conclusion. The diagram showcases the player's central role in the gaming experience, interacting with different aspects of the game world, and progressing towards the ultimate goal of completing the game.



## Deployment Diagram

## The distribution of software elements and hardware nodes that interact to facilitate the execution of the supplied use cases will be highlighted in the "Wandering in Woods" game's deployment diagram.

## The deployment diagram would show the interaction between the Character Creation component and the Player's interface (user's device), starting with the Create a Character use case. This element might be a character customization service or a component of the game server. The link between the Player's device and the relevant backend services in charge of controlling character creation options is shown in the diagram.

## The graphic would show interactions between the Player's device, the Game World component, and perhaps a Navigation module for the Explore the Forest use case. This would demonstrate the relationship between the user interface—where the player controls the character—and the system components in charge of managing the dynamics of the forest environment.

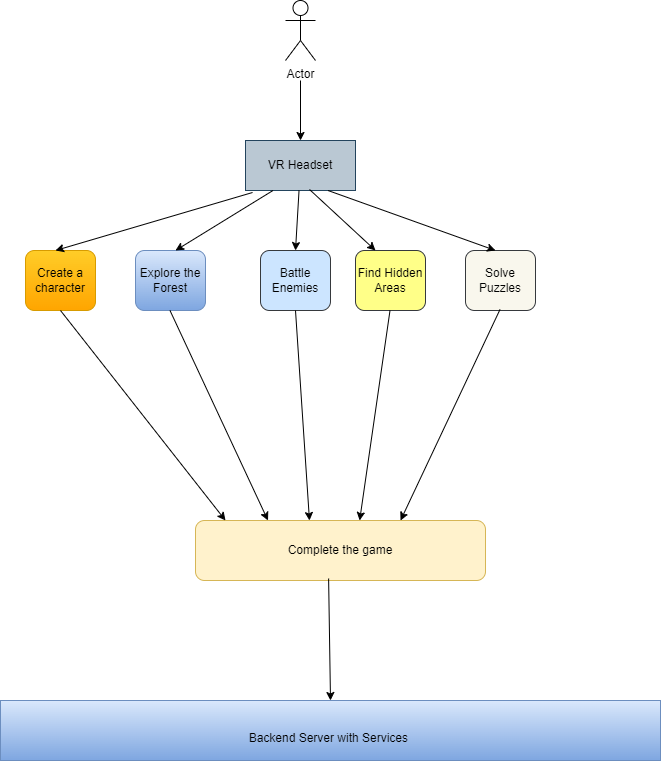
## The Battle System component would communicate with the Player's device in the Battle Enemies use case. This would include enemy interactions, experience point and loot calculation, and combat mechanics. The illustration would show how the player's actions cause the battle system to react in certain ways.

## The connectivity between the Player's device and the Puzzle Solver component would be demonstrated by the Solve Puzzle use case. This element could control the clues, item interactions, and puzzle logic. The deployment diagram would show how the game's architecture is used to process the puzzle-solving elements.

## The graphic would show interactions between the Player's device and the Exploration Manager component for the Find Hidden Areas use case. This element may regulate the accessibility of challenges, side quests, and hidden regions. The connection between the player's exploration and the dynamic features of the game world will be shown in the diagram.

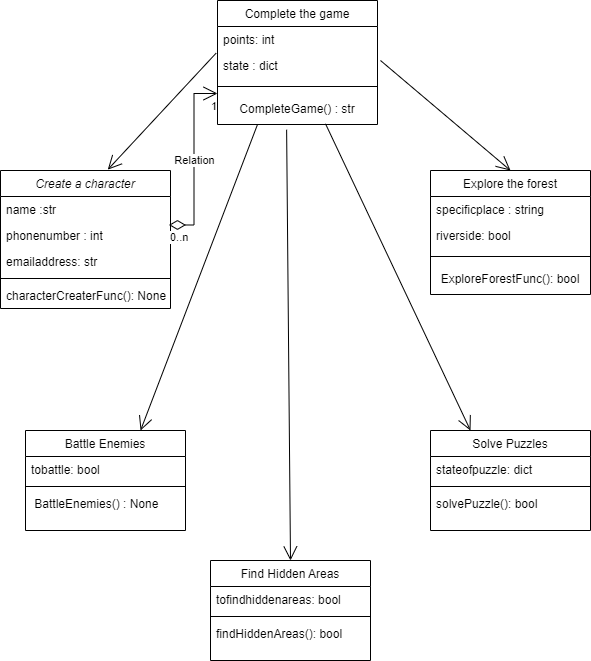
## Last but not least, the deployment diagram for the Complete the Game use case would depict the communication between the Player's device and the Game Completion component. The final boss fight, progress monitoring, and showing the game's conclusion would all be handled by this component. The diagram would highlight the conclusion of the player's trip and the factors contributing to the game's resolution.

## In conclusion, the deployment diagram would show clearly how the player's interactions with various software components inside the game's architecture connect with their device. Each use case would relate to a particular interaction scenario, demonstrating how the various parts interact to deliver the complete gaming experience.



## Class Diagram

The class diagram for the core system of the Wandering in the Woods game is depicted in Figure 4. The classes in the diagram are described below the figure.



**Complete the Game**: a class that has a CompleteGame function that returns the user won or not so string return type. It has two variable points and a state.

points: stores the current points so its state is int

State: stores the current state in a dictionary so its return type is a dict

**Create a Character:** a class that has a method characterCreateFunc() which just creates a character so its return is None type. It has two 3 variables.

**name:** name of character so string type

**phone number:** stores the phone number so integer type

**email address:** stores email address so string type

**Explore the forest**: has a method ExploreForestFunc which just explores the forest and returns True or False so its type is boolean. It has 2 parameters

**specific place:** a nearby place to be given so string type

**Riverside:** whether to look near the riverside or not so bool type

**Battle Enemies:** It has a method called BattleEnemies, to just battle. It has a variable to battle so it's bool type.

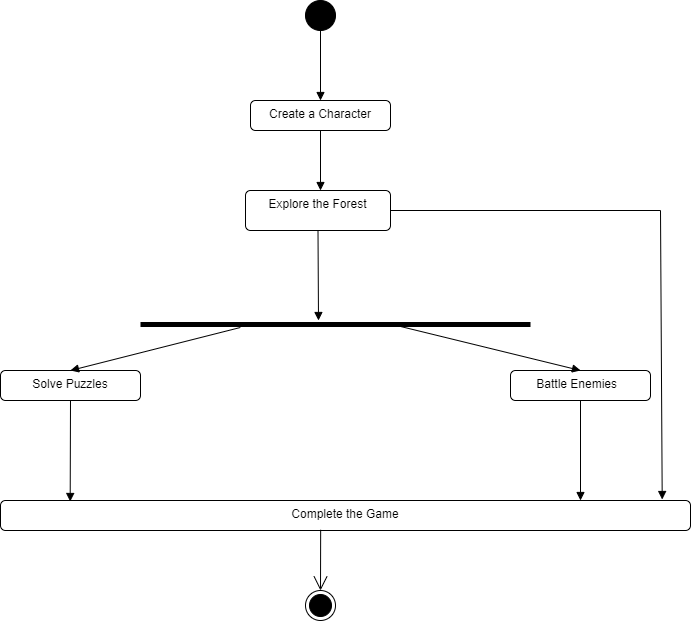
**Solve Puzzle**: It has a method solvePuzzle. Of type bool to return solved or not, True or False. It has a variable state of Puzzle to store its state in a dictionary.

**Find Hidden Areas:** It has a method find hidden areas to find hidden areas and return True or False whether found or not. It has a variable to FindHiddenAreas to take True or False whether to find areas or not.

*Figure SEQ Figure \\* ARABIC 4. Class diagram for the AR Room Designer System*

## State Diagram

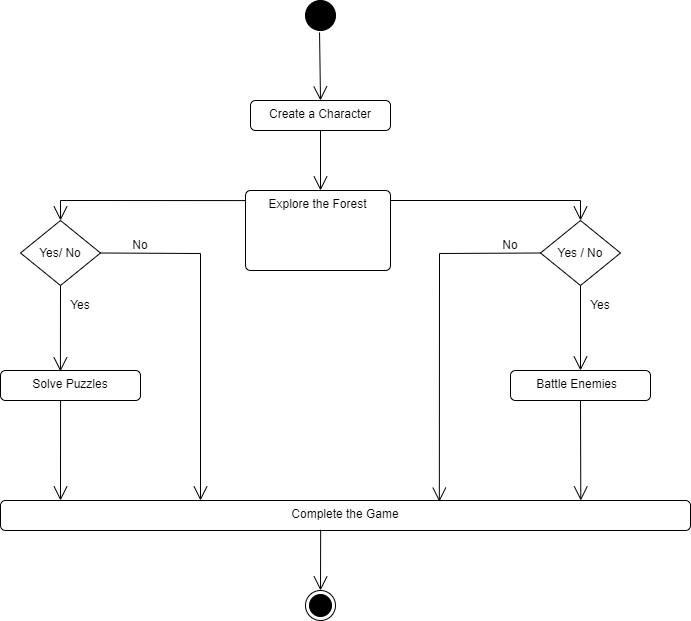
## Here each use case is represented as a state and each state has its own behavior which could finally affect the points of a character. Thus finally the character could be winning or losing at the time of ending the game.



*Figure SEQ Figure \\* ARABIC 5. State diagram for the AR Room Designer System.*

## Activity Diagram

## In this Activity diagram, each use case specified above is taken as a state and each state has some input and output flow. Some states have a boolean input so that to use those use cases or not. If we use that use case then Yes should be there or else we can simply go to the final use case and end the game. The node starts from the character creation and then moves further for other use cases as other activities



*Figure SEQ Figure \\* ARABIC 6. AR Room Designer high-level activity diagram.*

# Customer Journey Map

The "Wandering in Woods" game's customer journey map tracks the player's progress across several important use cases:

After a game has been made, the player begins by making a character with their preferred appearance and gear. This includes making choices for gender, race, hairstyle, armor, and weaponry.

After creating a character, the player enters the forest and sets out on a trip. They are able to move around, interact with things, and engage in combat. The forest encourages curiosity by providing puzzles and secret spaces.

Enemies are encountered by the player while they are exploring. As players engage in combat utilizing weapons and skills, they gain experience points and loot, strengthening the character.

Solve Puzzles: In the forest, the player finds puzzles to solve. They find hidden things and clues to solve these puzzles, moving them along by utilizing their intellect and inventiveness.

Discover Hidden Areas: Exploring leads to the discovery of secret locations that contain useful things and provide chances for side quests. Challenges and missions are undertaken by the player, increasing immersion.

Completing the game requires beating the ultimate boss once the player has encountered all use cases. Victory results in seeing how the game ends, capping off the enjoyable voyage.

As they explore, solve, and conquer problems along the way, players are more engaged, which makes for a satisfying gaming experience.

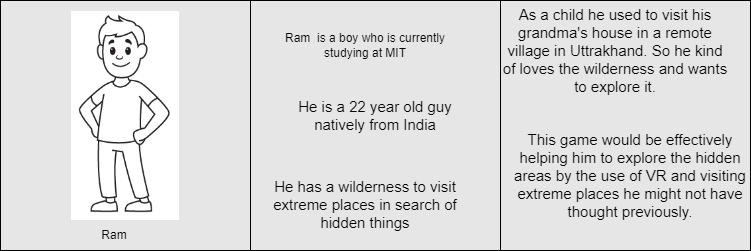
Here I have made it mandatory to solve puzzles and find hidden areas otherwise the creator can choose according to his likes.



*Figure SEQ Figure \\* ARABIC 7. Customer journey map for the AR Room Designer system.*

# Personas

Here I am taking a person Ram as a user who will be using this game to relieve his brain from studying all full day as he is a student at MIT. Now he has also been finding hidden secrets in the US at various historical places. So in a way, this game will be a perfect match for these types of users. Now Ram might use this game to explore his wilderness all along. If he is using this game always he might develop a disorder or problem related to VR usage and all. So he must use it not very often.

****

*Figure SEQ Figure \\* ARABIC 8. Sample persona to evaluate the AR Room Designer system.*

# UI Mock-up

# Here we have created an initial mockup of the game where you can see the pinpoint trees, rivers, mountains, sun, and other things that are usually present in the forest.

# Now when the user is creating their own game they can easily remove one or more things apart from the background so that they can attach the things of their own liking.



*Figure SEQ Figure \\* ARABIC 9. User interface mock-up for AR Room Designer system.*

# Testing Strategy

The importance of testing in the creation of the Wandering in the Woods game is highlighted here.

* To guarantee the game's quality: Testing enables the removal of flaws and errors from the game. The player experience depends on this because bugs can make a game frustrating or unusable.
* Testing can assist in identifying usability problems, such as unclear menus or challenging instructions, and helping to find solutions. This is crucial for increasing the game's usability for players of all skill levels.
* To gather player input: Testing can assist in gathering user feedback on the features, gameplay, and overall experience of the game. Before the game is made available to the general public, this feedback can be used to enhance it.
* To conform to rules: In some circumstances, games may need to go through testing to conform to regulations, like those concerning accessibility or violence.

Overall, testing is a crucial step in the creation of video games. It aids in ensuring the game's quality, identifying usability problems, gathering player input, and adhering to rules. A positive gaming experience can be helped by creators by giving the game a comprehensive test run.

## Unit Testing

## A fundamental testing procedure called unit testing involves testing each component, or unit, of a software application separately. Every unit must function as intended, and any flaws or problems must be discovered early in the development process. To ensure the functionality of each tiny component of the "Wandering in the Woods" game, unit testing would entail testing each component separately.

## **Illustration:** Player Movement

## Take a unit test for player movement in the game as an example. The gameplay depends heavily on the player's capacity for fluid movement and environment interaction. A controlled environment where the player character can walk in all directions, jump, and interact with items might be created for the test.

## **The unit test would cover situations like** determining if the player character reacts to keyboard or gamepad inputs effectively. Checking sure the character accurately bumps into objects and surfaces.

## Ensuring that the mechanic for jumping functions as planned and preventing undesirable behavior, such as double jumps.

## Demonstrating that object interactions have the desired consequences.

## The only thing that would be tested throughout the unit test is player interaction and movement. Before moving on to more extensive testing, any problems relating to these aspects would be found and resolved.

## Integration Testing

## Testing the cooperation and interactions between various software units or components is known as integration testing. Making sure that these parts function together as naturally as they would in the actual application is the main objective. Integration testing in the game would involve analyzing the relationships between various gameplay components.

## Example: Player and animal interaction

## Integration testing for the "Wandering in the Woods" game may concentrate on how the player character interacts with the various forest species. The player character and the various animal species are the different units in this scenario.

## The integration test may consist of:

## Determining if the player character's actions—such as approaching, engaging with, or avoiding creatures—cause the creatures to respond in a suitable manner.

## Confirming that friendly animals react favorably to the player's actions and perhaps even provide aid.

## Ensuring that frightful predators respond to the player's presence realistically, either by attacking or running away.

## Verify the unique responses of mythical beings that fit the story of the game.

## This test makes sure that player-creature interactions are fluid and improves the immersive experience the game seeks to deliver.

## System Testing

## System testing assesses the entire application or system as a whole. It focuses on ensuring that all parts and units cooperate well and adhere to the standards. System testing for the game would entail assessing its general usability, dynamics, and performance.

## Full Gameplay Scenario, for instance.

## During system testing for "Wandering in the Woods," testers would complete the entire game in order to simulate the player's experience. This entails navigating a variety of situations, cracking puzzles, engaging with wildlife, and getting past obstacles.

## The system evaluation would comprise:

## Game advancement is being tested to make sure that players can go through stages by resolving puzzles and conquering challenges.

## Examining the game for any flaws or problems that might appear after prolonged playtime.

## Confirming the consistency and enjoyment of the immersive experience, including VR technologies.

## Evaluating the overall performance, taking into account frame rates, load times, and the responsiveness of the controls.

## System testing is to make sure that the entire game runs properly and offers players an enjoyable and challenging experience.

## User Acceptance Testing

## User acceptability testing (UAT) assesses the program from the viewpoint of the end user to ascertain whether it satisfies their needs and expectations. This final step, which takes place before the game is released, involves actual users or representatives of the intended market.

## Player Experience Evaluation, for instance.

## User acceptance testing for "Wandering in the Woods" might choose a set of players who enjoy adventure and VR games. They would participate in the game and share their thoughts afterward.

## The UAT procedure entails: Allowing participants to complete the game while watching their responses and interactions.

## Gathering opinions on the gameplay's structure, challenges, and general fun.

## Identifying any usability problems, such as perplexing menus or imprecise directions.

## Ensuring that the immersion level of the game matches the expectations and tastes of the players.

## UAT demonstrates that the game is ready for launch and that it meets players' expectations.

## In conclusion, each testing phase plays a crucial role in ensuring the quality and functionality of the "Wandering in the Woods" game. Unit testing ensures that individual components work correctly, integration testing focuses on interactions, system testing assesses the game as a whole, and user acceptance testing ensures that the game meets players' expectations. Together, these testing phases help create an engaging, immersive, and enjoyable gaming experience.

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## **9.0 Functional Requirements**

**Player Interaction and Movement:**

An immersive experience requires fluid and quick player movement. The game world should be easily navigable by players utilizing keyboard, mouse, or controller inputs. Players should be able to explore the forest and engage with numerous things thanks to this movement's naturalness and intuitiveness.

**Logic puzzle design:**

Through their ability to test players' problem-solving abilities, puzzles give the game more depth. There should be a variety of problems in the game, from easy to difficult. Different skill levels of players should be engaged by the challenge level. The immersion in the game can be maintained by making sure that problems have logical solutions.

**Creature Behavior:**

Creatures' behaviors should match their traits in order to bring the forest habitat to life. Friendly animals should help or give players suggestions to increase the sensation of involvement. Fearsome predators ought ideally to respond to the player's activities in a dynamic way, such as by pursuing them if they provoke them or fleeing if they intimidate them. Mystical creatures ought to exhibit distinctive traits that are connected to the game's plot and provide intrigue.

**Obstacle Challenges:**

Players must go over dynamic challenges presented by obstacles. Physical obstacles, environmental dangers, or moving items are a few examples. In order to motivate players to consider other answers and routes, challenges should call for strategic thinking and timing.

**Narrative Progression:**

The game's narrative carries players along as they progress. The tale, the characters, and the goals should be clearly communicated through dialogue, texts, and visual signals. Players are kept interested in the mystery playing out in the forest thanks to effective pace and storytelling strategies.

**Visuals and graphics:**

Visual appeal considerably enhances player immersion. High-quality graphics that capture the mystery and adventure elements are required for the game. To create a captivating and visually appealing atmosphere, the visual style—which includes lighting, textures, and art design—should complement one another.

**Virtual Reality Integration:**

The game heavily utilizes virtual reality. Players donning head-mounted displays (HMDs) ought to have a seamless and immersive experience thanks to the incorporation of VR gear. Using hand controllers, players should be able to interact with the virtual environment while feeling present and involved.

**Save and load capabilities:**

Players can keep making progress between play sessions thanks to saving and loading features. In order to give players a comfortable gaming experience, this capability must allow them to resume where they left off. The effective use of save and load systems eliminates frustration brought on by lost progress.

**10.0 Non-Functional Requirements**

**Performance:**

Maintaining a high level of performance is essential, particularly in a VR environment where frame rate directly affects player experience. Motion sickness is less likely to occur when the frame rate is steady, and a fluid experience is guaranteed. To keep players from getting impatient, load times for scenes and levels should be kept to a minimum.

**Immersion and audio:**

The audio design enhances immersion. Ambient music, wildlife noises, and realistic environmental sounds all help to create a sense of presence in the forest. Players can more easily locate various game elements thanks to directional audio cues, which improves the entire experience.

**Compatibility**:

The game should function flawlessly on desktops, tablets, and smartphones. It is available on a variety of devices. The potential player base is increased by ensuring device compatibility, increasing the game's reach.

**Usability**:

Players should be able to interact with the game without encountering extra impediments thanks to a straightforward user interface and simple controls. By guiding them through fundamental interactions, mechanics, and controls, in-game tutorials can help players feel less frustrated and accelerate their learning curve.

**Comfort and Safety in VR**

Experiences in virtual reality must put player comfort first. Motion sickness should be avoided by design decisions, and VR interactions should resemble real-world motion to lessen discomfort. With this strategy, players can play the game for a long time without suffering any unfavorable bodily repercussions.

**Accessibility:**

Players with disabilities should be catered to in the game. The game may be played by a wider spectrum of players thanks to adjustable settings for visual and aural impairments, such as color-blindness modes, subtitles, and alternative control schemes.

**Data privacy and security:**

The importance of safeguarding user data and abiding by privacy laws cannot be overstated. If there are online features, communication and data exchange should be encrypted to avoid unauthorized access. The game must manage user data securely.

The "Wandering in the Woods" game may give an immersive, engaging, and user-friendly experience that is in line with the objectives and expectations of both players and the developers by abiding by these thorough functional and non-functional requirements.