



Final Project
wandering in the woods

Software Engineering
Instructor's name: Ziad Al Sharif

Team members:

- Ali Ahmed Bin Khaled Banagga
- Ali Mir Gufran
- Anjaneyulu Kambampati
- Chaitanya Obiliseti
- Sai Sampath Chowdary Pusuluri

Table of Contents

Abstract
1. Introduction:	3
2. Roadmap:.....	3
3. Scope:	3
4. Agile Software Development	3
4.1 Principles of Agile development.....	4
5. Project Requirements	5
6. Features of cloud application:.....	5
7. Use Stories for cloud application	5
8. Mobile Development Stack	5
9. Infrastructure for cloud Application.....	5
10. Security and Privacy Considerations... ..	6
11. Testing strategy	6

Abstract

This project introduces the creation of a compelling and enigmatic game titled "Wandering in the Woods." The game has users navigate through a virtual forest while competing against an AI opponent, resulting in surprising and exciting situations. Every motion made by a player causes visual modifications in the grid boxes, delivering a dynamic and immersive experience.

1. Introduction

"Wandering in the Woods" is appropriate for kids of various grade levels, with variable levels of complexity and involvement. For K-2 kids, the game has a basic interface in which two players start diagonally and finally collide, resulting in a positive message. Grades 3-4 introduce 3 to 4 players, more teaching aspects, and accessible statistics. Students in grades 6-8 can customize grids and explore more complex features.

2. Roadmap

- The project roadmap outlines the implementation phases:
- Identification: Determine the technological stack and the implementation approach.
- Infrastructure Setup: Identify critical service catalog offers and set up the necessary infrastructure.
- Architecture and Development Model: Define the development model, implement backup and recovery mechanisms, and install required binaries.
- Environment Build: Perform pre-installation checks, configure cloud components, and provision required use cases.
- Advanced Features and Deployment: Implement sophisticated usage and performance monitoring, as well as project live deployment.

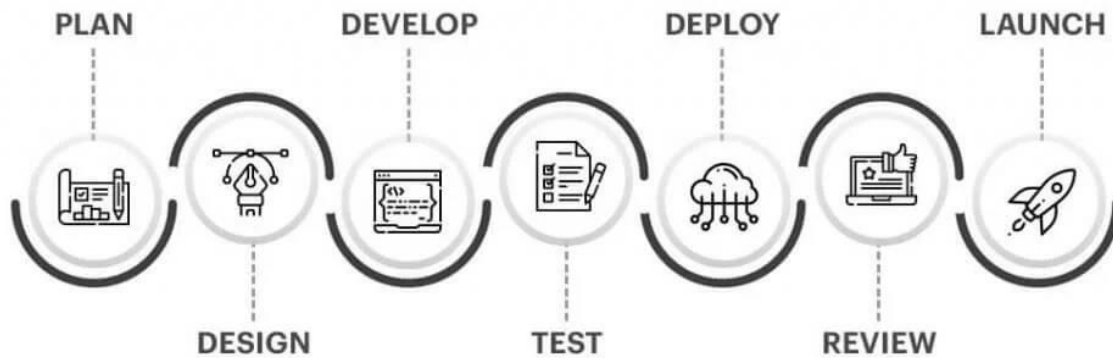
3. Scope

The game's scope is designed for children in grades K-2, 3-4, and 6-8. Its primary goal is to provide a fascinating and mysterious gameplay experience, stimulating curiosity and involvement among players.

4. Agile Software Development

The project supports the ideas of Agile software development, a dynamic methodology that promotes adaptability and customer-centricity. Agile methodologies emphasize iterative development with short, customer-focused sprints. It embraces changing requirements, values customer collaboration, and prioritizes features based on user desire. User stories guide

development, cross-functional teams foster collaboration, and continuous testing ensures quality. Adaptive planning and open communication result in adaptable and meaningful outcomes. Regular demos and retrospectives allow for progress tracking and continual improvement. This Agile strategy allows the project to respond quickly to changes, generate user value, and build a functioning and innovative game experience.



Agile Methodologies

4.1.Principles of Agile Development:

Agile development is founded on ideas that drive its effectiveness. One guiding philosophy is an uncompromising commitment to client satisfaction through timely and incremental product release. This guarantees that clients obtain concrete value early in the process and may provide quick input for modification.

Another essential principle is the realization that requirements are not fixed. Agile acknowledges the realities of changing demands by allowing changes to be introduced at any level of development, even after the initial development is complete. This adaptability not only accommodates shifting market demands, but also stimulates continual improvement based on real-world usage and feedback.

Agile development also encourages collaboration and active participation of stakeholders throughout the project. This guarantees that the product satisfies the actual demands of people and aligns with their expectations. Regular communication and transparency develop confidence and allow the project team to pivot as needed, resulting in the creation of a product that is both relevant and valuable.



5. Project Requirements

Multi-tenancy support, easy and fast delivery, on-demand integration capabilities, robust security measures, and the construction of a sustainable infrastructure are all critical needs for the cloud-based system project.

6. Key Features of the Game Application

The game application includes a number of critical features, such as resource pooling to optimize system usage, on-demand self-service for easy interaction, streamlined maintenance procedures, robust security measures to protect user data, and ensuring high availability for uninterrupted gameplay.

7. User Stories for Cloud Application

User Story 1: As a player, I want the ability to select from a range of grid sizes, allowing me to adjust the game's difficulty level to my preferences.

8. Mobile Development Stack

The mobile development stack relies around user personas, the Arcade library for front-end development, and user interactions, ensuring a seamless and entertaining gameplay experience.

9. Infrastructure for Game Application

The game app is intended for use on both Android and iOS smartphones. The basic development framework makes use of the Arcade library, which is well-known for its speedy front-end implementation.

10. Security and Privacy Considerations

The application prioritizes data security. To protect user data both on the device and in the cloud, stringent data encryption methods are in place.

11. Testing Strategy

The project implements a thorough testing technique that includes both functional and non-functional elements. In the field of functional testing, rigorous review is carried out to assure the application's seamless performance at every developmental phase.

This method is divided into several steps. To begin, Unit Testing thoroughly evaluates each individual component to ensure that they function as intended. Integration Testing examines the cohesion and interaction of these units when they are combined into modules. The following stage, System Testing, evaluates the program holistically, discovering any potential faults caused by module integration.

Furthermore, Acceptance Testing mimics real-world scenarios to see whether the application meets user expectations and does its designated functions efficiently.

For a robust application, the project prioritizes non-functional testing alongside functional testing. Performance testing assesses responsiveness and stability under various pressures, such as stress, load, and endurance. Security testing is critical for protecting sensitive data by examining vulnerabilities and potential breaches.

Usability Testing investigates user friendliness by having real users engage with the product, whereas Compatibility Testing ensures consistent functionality across various devices and platforms.

This comprehensive testing approach, which includes both functional and non-functional evaluations, aims to uncover and correct flaws. It ensures that the program runs smoothly while also being secure, trustworthy, and user-friendly.