

Sheep and Wolves Game Design Specification

1. Introduction

The Sheep and Wolves Game is a web-based puzzle game built using Flask, HTML, CSS, JavaScript, and SQLite. The game challenges players to transfer sheep and wolves across a river while adhering to specific rules. This document outlines the features, architecture, development process, deployment, and maintenance of the game.

2. Features

2.1 Basic Features

User Authentication: Users can sign up, log in, and log out of the game.

Gameplay: Players can transfer sheep and wolves across the river while adhering to game rules.

Solution Generation: The system generates a solution for the puzzle using semantic nets.

Timer: A timer tracks the time taken by the player to solve the puzzle.

Error Handling: Proper error messages are displayed for invalid input and other errors.

2.2 Additional Features

Reset Game: Players can reset the game to its initial state.

View Solution: Players can view the solution to the puzzle. Upon clicking "View Solution," the product generates and displays the solution for the given number of sheep and wolves.

3. Design Requirements

3.1 Frontend Design

HTML Templates: Flask serves HTML templates for signup, login, and gameplay screens.

CSS Styling: CSS is used for styling the interface and making it visually appealing.

JavaScript Interaction: JavaScript handles user interactions and game logic on the client side.

3.2 Backend Design

Flask Application: Flask handles routing, user authentication, and gameplay logic.

Database: User data (fullname, email, password) is stored in an SQLite database.

Semantic Nets Agent: The `SemanticNetsAgent` class generates solutions for the puzzle using semantic nets. The solution is fetched and displayed upon user request.

4. Architecture

4.1 Client-Server Architecture

Client: The client interacts with the game through a web browser.

Server: Flask serves as the backend server, handling HTTP requests from the client and responding with data or HTML templates.

4.2 Communication Protocol

HTTP: The communication between the client and server is done via the HTTP protocol.

5. Development Process

5.1 Agile Methodology

Sprints: Development follows Agile principles with regular sprints, typically lasting 1-2 weeks.

Iterations: The project undergoes multiple iterations based on user feedback and changing requirements.

5.2 Testing

Unit Tests: Unit tests are written to test individual components and functions.

Integration Tests: Integration tests verify the interaction between different modules of the application.

User Acceptance Tests (UAT): UAT is conducted to ensure that the game meets the requirements and expectations of end users.

6. Deployment

6.1 Deployment Platform

The Flask application is deployed to a web server or platform where users can access it.

Cloud platforms like Heroku, AWS, or Azure may be used for deployment.

7. Conclusion

The Sheep and Wolves Game is designed to provide an engaging and challenging experience for players while demonstrating the capabilities of Flask and web development technologies. By following the design specification outlined in this document, the development team aims to deliver a high-quality game that meets user expectations and delivers value.