Technical Design Document – Outline

# Title Page

# Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** |
| 0.04 | 2023-12-14 | Diego Portella |  |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

Contents

[1 Title Page 1](#_Toc153485133)

[2 Document History 1](#_Toc153485134)

[3 Table of Contents 1](#_Toc153485135)

[4 Game Summary 2](#_Toc153485136)

[5 Development Environment 2](#_Toc153485137)

[5.1 Development Hardware 2](#_Toc153485138)

[5.2 Programming Languages 2](#_Toc153485139)

[5.3 Development Tools 2](#_Toc153485140)

[5.4 External Code 2](#_Toc153485141)

[5.5 Game Engine 2](#_Toc153485142)

[6 Architectural Analysis 2](#_Toc153485143)

[6.1 Entities 2](#_Toc153485144)

[6.2 Components 3](#_Toc153485145)

[6.3 Systems 3](#_Toc153485146)

[6.4 Behavioral Analysis 3](#_Toc153485147)

[7 Technical Risks 3](#_Toc153485148)

# Game Summary

A frog tries to cross several lanes of busy traffic without getting run over. Then the frog tries to cross a river without drowning (this frog can’t swim). The frog must cross the river by hopping on logs or turtles that are floating down the river. If the frog falls off the log or turtle, the frog drowns in the river.

# Development Environment

## Development Hardware

PC windows 10 or 11

## Programming Languages

This game engine and game will primarily be written in C++23 with SFML 2.6.1

## Development Tools

Visual Studio

## External Code

SFML (https://www.sfml-dev.org)

## Game Engine

GEX game engine developed in prog1266. This is an Entity-Component-Systems based game engine.

# Architectural Analysis

## Entities

|  |  |
| --- | --- |
| Entities | Responsabilities |
| spawnPlayer | It’s used to create the player |
| spawnLane1 | Spawns raceCarL |
| spawnLane2 | Spawns tractor |
| spawnLane3 | Spawns car |
| spawnLane4 | Spawns raceCarR |
| spawnLane5 | Spawns truck |
| spawnLane6 | Spawns 3turtles |
| spawnLane7 | Spawns tree1 |
| spawnLane8 | Spawns tree2 |
| spawnLane9 | Spawns 2turtles |
| spawnLane10 | Spawns tree1 |
| spawnGoal | Spawns lillyPad |
| spawnLives | Spawns lives |
|  |  |

## Components

|  |  |
| --- | --- |
| Component | Purpose |
| CTransform | Holds the position, velocity, and direction of entity |
| CBoundingBox | Used for collision detection. All entities that participate in collisions must have this component. |
| CSprite | The sprite used to draw the entity, to be drawn an entity must have a sprite component. |
| CAnimation | Represents an animation component. |
| CState | Represents a state component for storing the state of an entity. |
| CInput | Represents an input component for user input. |

## Systems

|  |  |
| --- | --- |
| System | Purpose |
| sMovement | Moves all entities that have a CTransform component |
| sCollision | Detects collisions between entities that have a CBoundingBox component and takes appropriate action when collisions occur. |
| sUpdate | Updates the scene's state over time. |
| onEnd | Defines actions to be taken when the scene ends. |
| resetPlayer | Resets the player's position and state. |
| killPlayer | Handles necessary actions when the player dies. |
| updateScore | Updates the player's score based on the current position. |
| init | Initializes the scene based on a configuration file specified by the path. |
| loadLevel | Loads scene elements from a configuration file. |
| playerMovement | Handles player movement based on keyboard input. |
| adjustPlayerPosition | Ensures the player's position is within the screen boundaries. |
| checkPlayerState | Checks and manages the player's state, such as whether it's alive or dead. |
| registerActions | Registers player actions associated with control keys. |

## Behavioral Analysis

1. Start Button to start the game on Menu.
2. Receive inputs to move the frog.
3. Moving the character
4. Checking collisions during the frog pursuit
5. If frog collides with something, it should lose points and dies.
6. While the frog advances to its goals, each lane must count 10 points.
7. If the frog has remained lives or still hasn’t completed the all-lily pads, the game goes on
8. End

# Technical Risks

|  |  |  |
| --- | --- | --- |
| Risk | Severity | Mitigation (what is to be done to eliminate or minimize this risk) |
| Uncertainty in Implementing SpawnFly Feature | Moderate | Detailed Planning: Break down the feature into smaller tasks and plan the implementation details.  Prototyping: Create a prototype or proof of concept to test the feasibility and performance of spawning flies on lily pads.  Research: Explore existing solutions or similar implementations to gain insights.  Code Reviews: Involve experienced team members in code reviews to catch potential issues early. |
|  |  |  |
|  |  |  |
|  |  |  |