Technical Design Document

# Game Project Overview

This project is to create an engaging and immersive experience using SDL2 in C++. The focus areas are graphics, audio, rendering, UI, and AI. The game will feature various AI components such as steering behaviors, state-driven agents, sports simulation, graphs, and goal-driven agent behavior. To showcase these AI features, the game will include multiple modes, including a 2D sports simulation mode. This document provides an extremely detailed technical design for each of these aspects.

## 1. Graphics

The graphics subsystem will be responsible for rendering visual elements on the game screen, including sprites, animations, and user interface elements. SDL2's rendering capabilities will be utilized for efficient and platform-independent graphics rendering.

### 1.1 Sprites and Animations

1. The game will support sprite-based animations using sprite sheets.
2. Sprites will be managed through a sprite manager that loads and organizes sprite sheets.
3. The sprite manager will handle animation frames, timing, and sprite rendering.
4. Sprites will be rendered using SDL2's rendering functions and textures.

### 1.2 User Interface (UI)

1. The UI system will include elements such as buttons, text boxes, and menus.
2. UI elements will be represented by separate classes, allowing easy manipulation and rendering.
3. The UI system will handle user input events, such as button clicks.
4. UI elements will be rendered on top of the game scene using the same rendering functions used for sprites.

### 1.3 Rendering Techniques

1. The rendering subsystem will utilize modern rendering techniques, such as sprite batching, to optimize performance.
2. Rendering will support different layers to ensure proper rendering order of game elements.
3. Shader programs will be used to apply visual effects such as lighting or post-processing.
4. The rendering system will make efficient use of SDL2's rendering functions and rendering targets.

## 2. Audio

The audio subsystem will handle sound effects and background music to enhance the gaming experience. SDL2's audio capabilities will be utilized for cross-platform audio support.

### 2.1 Sound Effects

1. Sound effects will be represented as separate audio files, such as WAV or MP3.
2. The sound manager will load and manage sound effects.
3. Sound effects will be played using SDL2's audio functions.
4. The sound manager will handle volume, playback, and mixing of sound effects.

### 2.2 Background Music

1. Background music will be represented as separate audio files, such as OGG or MP3.
2. The music manager will load and manage background music tracks.
3. Background music will be played using SDL2's audio functions.
4. The music manager will handle volume, playback, and looping of background music.

## 3. Rendering

The rendering subsystem will handle the display of the game scene, including the background, sprites, and user interface. SDL2's rendering functions will be utilized for efficient rendering.

### 3.1 Game Scene

1. The game scene will be represented as a 2D space with a fixed size.
2. The game scene will be rendered using SDL2's rendering functions and textures.
3. The game scene will support multiple layers for rendering different elements.
4. The rendering order of elements will be determined by their layers and positions.

### 3.2 Camera

1. A camera system will be implemented to control the view of the game scene.
2. The camera will allow panning and zooming to focus on different areas of the scene.
3. The camera will be responsible for transforming world coordinates to screen coordinates.

## 4. UI

The UI subsystem will handle the user interface elements, including menus, buttons, and text boxes.

### 4.1 Menu System

1. The menu system will provide a hierarchical structure for organizing different screens and menus.
2. Each menu screen will be represented by a separate class, allowing easy management and rendering.
3. The menu system will handle user input events, such as button clicks or text input.
4. The menu system will support transitions between screens with animations and effects.

### 4.2 Buttons and Text Boxes

1. Buttons and text boxes will be separate UI elements with their own classes.
2. Buttons will respond to user clicks and trigger corresponding actions or events.
3. Text boxes will handle user input for text-based interactions, such as name input or chat messages.
4. Both buttons and text boxes will be rendered using SDL2's rendering functions and textures.

## 5. AI

The AI subsystem will provide intelligent behaviors for non-player characters (NPCs) and agents within the game. It will include steering behaviors, state-driven agents, sports simulation, graphs, and goal-driven agent behavior.

A diagram of a process

Description automatically generated

It splits the AI task into three sections: movement, decision making, and strategy. The first two sections contain algorithms that work on a character by-character basis, and the other section operates on a team or side.

First, we must have some kind of infrastructure in two categories: a general mechanism for managing AI behaviors (deciding which behavior gets to run when, and so on) and a world interfacing system for getting information into the AI. Every AI algorithm needs to honor these mechanisms.

There must be means to turn whatever the AI wants to do into action on-screen. This consists of standard interfaces to a movement and an animation controller, which can turn requests such as “pull lever 1” or “walk stealthily to position x, y” into action.

### 5.1 Steering Behaviors

1. Steering behaviors will enable NPCs to navigate the game world intelligently.
2. Common steering behaviors will include seek, flee, arrive, pursue, evade, and wander.
3. Each NPC will have a steering behavior component that controls its movement.
4. Steering behaviors will be implemented using vector operations and mathematical algorithms.

### 5.2 State-Driven Agents

1. State-driven agents will have different behavioral states that determine their actions.
2. States will include idle, roaming, attacking, defending, and more.
3. The state manager will control the transitions between states based on game events and conditions.
4. Each state will define the agent's behavior, movement, and interaction with the game world.

### 5.3 Sports Simulation

1. The sports simulation component will provide realistic gameplay for sports-related activities.
2. It will include physics calculations for ball movement, player interactions, and scoring.
3. The sports simulation will handle collision detection and resolution between game objects.
4. AI-controlled players will utilize the sports simulation to make intelligent decisions during gameplay.

### 5.4 Graphs

1. Graphs will be used to represent spatial relationships and navigation paths within the game world.
2. Graph nodes will represent points of interest, such as waypoints or key locations.
3. Graph edges will define connections between nodes and their associated costs.
4. Pathfinding algorithms like A\* or Dijkstra's algorithm will be implemented to find optimal paths.

### 5.5 Goal-Driven Agent Behavior

1. Goal-driven agents will have specific objectives or goals they strive to achieve.
2. Agents will evaluate the current game state and make decisions based on their goals.
3. The goal-driven behavior system will prioritize and execute actions to achieve the agent's goals.
4. Goal progress and completion will trigger state changes and reactions from other game systems.

# Conclusion

This technical design document provides an extensive and detailed plan for the game project. By utilizing SDL2 in C++, the game will feature impressive graphics, audio, rendering, UI, and AI components. Multiple game modes will showcase the different AI features, including a 2D sports simulation mode. The detailed specifications provided will guide the development process and ensure the successful realization of the project's goals.

# Portfolio stuff

This is the stuff we can say/ put on our portfolio to showcase and describe what we did to seem impressive

This project demonstrates my expertise in game development, specifically focusing on advanced AI-driven features. By leveraging SDL2 in C++, the project showcases high quality graphics, audio, rendering, UI, Gameplay, and AI components. The game includes various modes, 2D sports simulation experience. This project highlights my ability to design and implement complex AI systems, pushing the boundaries of player engagement and interaction.

Technical Details and Learning Objectives:

1. Graphics, Audio, and Rendering:
   * Utilize SDL2's rendering capabilities to deliver visually stunning graphics, sprite-based animations, and efficient rendering techniques.
   * Implement advanced rendering techniques such as sprite batching, shader programs, and rendering layers to optimize performance and visual effects.
   * Leverage SDL2's audio capabilities to create immersive sound effects and background music, enhancing the overall gaming experience.
2. User Interface (UI):
   * Design and develop a robust UI system capable of handling user input events and rendering various UI elements such as buttons, menus, and text boxes.
   * Implement a flexible menu system with smooth transitions, animations, and visual effects for seamless user navigation.
   * Focus on creating an intuitive and user-friendly UI design that enhances the gameplay and overall aesthetic appeal.
3. Artificial Intelligence (AI):
   * Implement advanced AI components, including steering behaviors, state-driven agents, sports simulation, graphs, and goal-driven agent behavior.
   * Master steering behaviors such as seek, flee, arrive, pursue, evade, and wander to create intelligent and lifelike NPC movements.
   * Develop a sophisticated state-driven agent system that controls NPCs' behavioral states, allowing them to interact with the game world dynamically.
   * Create a realistic sports simulation component that handles physics calculations, collision detection, and scoring mechanics.
   * Utilize graph-based algorithms like A\* or Dijkstra's algorithm to enable efficient pathfinding and navigation for AI-controlled agents.
   * Design goal-driven agent behavior to create intelligent decision-making based on the game state and objectives, leading to dynamic and challenging gameplay.

Learning Objectives:

1. Gain expertise in SDL2 library, specifically its rendering and audio capabilities, to create visually appealing and immersive gaming experiences.
2. Acquire in-depth knowledge of rendering techniques, such as sprite batching and shader programs, to optimize performance and create visually stunning effects.
3. Develop proficiency in designing and implementing complex UI systems, including menus, buttons, and text boxes, to enhance user interaction and navigation.
4. Master AI techniques such as steering behaviors, state-driven agents, graph-based algorithms, and goal-driven agent behavior to create intelligent and engaging NPCs and gameplay mechanics.
5. Showcase the ability to design and implement a realistic sports simulation component, complete with physics calculations, collision detection, and scoring mechanics.
6. Demonstrate strong problem-solving skills by overcoming technical challenges related to AI and integrating it seamlessly with other game systems.

This portfolio project presents a comprehensive showcase of my technical skills and expertise in game development, particularly in the areas of graphics, audio, rendering, UI, and AI. By focusing on advanced AI-driven features, including steering behaviors, state-driven agents, sports simulation, graphs, and goal-driven agent behavior, this project stands out as a testament to my ability to deliver engaging and immersive gaming experiences. Through extensive learning and implementation, I have gained valuable technical knowledge and practical experience that can be a valuable asset