CENG462 GAME DEVELOPMENT FINAL EXAM
Due Date and Time: 31/05/2024 23:59

Game Jam Details Document

Game Overview

This platform game aims to deliver an immersive experience with intricate camera controls, diverse levels, and engaging gameplay mechanics. Players will navigate through levels, encounter enemies, and face challenging boss fights while enjoying a seamless blend of visual and audio elements. The previous assignment may be a reference for you.

Core Features

Camera Control:

Parallax Scrolling for Depth:

Implement parallax scrolling to create a sense of depth by moving background layers at different speeds compared to the foreground.

Clamping Camera:

Ensure the camera is confined within the level boundaries to prevent showing out-of-bounds areas.

Cinemachine Camera:

Utilize Cinemachine for dynamic camera behavior, including smooth transitions and focus on key gameplay elements.

Levels:

Multiple Levels:

Design and develop multiple levels with increasing difficulty and complexity to keep players engaged.

Health System:

Player Health:

Implement a health system where the player has a certain amount of health that decreases when taking damage from enemies or hazards.

Health Pickups:

Include pickups that can restore the player's health.

Checkpoints and Respawning:

Checkpoints:

Place checkpoints throughout levels to save player progress.

Respawning:

Allow players to respawn at the last checkpoint upon death.

Enemies:

Enemy Variety:

Create diverse enemies with unique behaviors and attack patterns.

Enemy AI:

Implement AI for enemies to challenge the player effectively.

Audio:

Sound Effects:

Integrate sound effects for actions such as jumping, attacking, and taking damage.

Background Music:

Include background music that enhances the atmosphere of each level.

Menus:

Main Menu:

Design a main menu with options to start the game, select levels, and access settings.

Pause Menu:

Provide a pause menu that allows players to resume, restart the level, or return to the main menu.

Pickups:

Collectibles:

Scatter collectibles throughout levels that provide rewards such as points or power-ups.

Player Leveling:

Experience Points:

Implement a leveling system where players earn experience points (XP) from defeating enemies and completing levels.

Level Up:

Allow players to level up and gain new abilities or enhancements.

Level Select Menu:

Level Selection:

Create a level select menu that allows players to choose previously completed levels to replay.

Boss Fights:

Boss Enemies:

Design challenging boss fights with unique mechanics at the end of certain levels.

Minimap:

Level Minimap:

Include a minimap to help players navigate larger levels and locate objectives.

Software Design Patterns

Singleton Pattern: Use the Singleton pattern for managing game state and global settings.

Observer Pattern: Implement the Observer pattern for event handling, such as player health updates and enemy spawn notifications.

State Pattern: Apply the State pattern for managing player states (e.g., idle, running, jumping, attacking).

Decorator Pattern: Use the Decorator pattern to dynamically add behaviors to game objects, such as power-ups that temporarily enhance player abilities or enemy variations with additional features.

Design Pattern Implementation and Grading

Significance in Grading: Implementing software design patterns will carry significant weight in grading. Students must be consistent in applying these patterns throughout their project.

UML Class Diagrams: Support your implementation with detailed UML class diagrams to illustrate the design patterns used.

Convincing Application: Students must convincingly demonstrate the use of design patterns, showcasing how they enhance the game's structure and maintainability.

🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩

Grading: Every group will be graded relative to the performance of other groups. Your final grade will be based on how your project compares to those of your peers.

🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩🚩

Submission Requirements:

README Text File: Include a README text file in your project directory that lists all group members and their roles.

Project Files: Provide a drive link having the project files including your unity project.

Project Report: Provide a report summarizing the most important information about your project, including the design patterns used, key features, and any unique aspects of your game.

Presentation Video: All group members must participate in a recorded presentation video. This video should be uploaded to YouTube as an unlisted video, and the link must be shared with the instructor. You are responsible for ensuring that the video is accessible to the instructor. Failure to do so will result in a grade of 0 for the project.