Name of the Student: Baitur Temiraliev

Title: Snake Game Development

Overview of the Project:

The objective of this project is to develop a classic Snake Game. In this game, the player controls a snake that moves around the screen, consuming apples that appear randomly. Each time the snake eats an apple, it grows longer, making the game progressively more challenging as the player must avoid colliding with the snake's own body and the walls of the game area. The project aims to recreate the nostalgic experience of the traditional Snake Game with modern enhancements in graphics and user interface.

Specific Functionalities

1. Snake Movement Control

Description: The snake moves in four possible directions: up, down, left, and right. The movement is continuous, and the player can change the direction using keyboard arrows or screen taps/swipes.

Purpose: To enable the player to control the snake and navigate it through the game area to collect apples.

Requirements:

Smooth and responsive controls.

Immediate direction change capability without causing a reverse move (e.g., moving from left to right directly should be disallowed).

User Interaction and Interface: Players use keyboard arrow keys or swipe gestures on touchscreens to change the snake’s direction. The controls will be responsive to ensure seamless gameplay.

Impact on Existing Systems: No existing systems; integral to the game’s basic functionality.

Priority and Timeline: High priority. Development completed in the first week.

Acceptance Criteria: The snake should respond immediately to user inputs, and the movement should be smooth without lag.

Feedback and Iteration: Initial feedback collected from test players to refine control responsiveness.

2. Apple Generation

Description: Apples appear randomly on the game board for the snake to consume.

Purpose: To provide the primary objective for the player to chase and consume, causing the snake to grow.

Requirements:

Apples must not appear on the snake's body.

Apples should appear in random but reachable positions.

User Interaction and Interface: The apples are represented by distinct icons that are easily recognizable.

Impact on Existing Systems: Requires integration with the game's collision detection and scoring systems.

Priority and Timeline: High priority. Functional by the end of the first week.

Acceptance Criteria: Apples should appear in valid locations and the snake should grow upon consuming an apple.

Feedback and Iteration: Player feedback will ensure apples appear in balanced locations.

3. Collision Detection

Description: The game detects when the snake collides with itself or the walls, resulting in the end of the game.

Purpose: To provide challenge and define the game's win/lose conditions.

Requirements:

Accurate and immediate detection of collisions.

Appropriate gameover logic and messaging.

User Interaction and Interface: When a collision is detected, the game will pause, and a gameover screen will display the player's score.

Impact on Existing Systems: Integral to the game loop and must be flawlessly integrated.

Priority and Timeline: High priority. Development and testing completed within the second week.

Acceptance Criteria: Collisions must be detected consistently, and the game must end appropriately.

Feedback and Iteration: Regular testing conducted to refine collision logic.

4. Score Keeping

Description: Tracks and displays the player's score, which increases each time an apple is eaten.

Purpose: To provide players with feedback on their progress and to add a competitive element.

Requirements:

Realtime score updates.

Clear display of the score on the game screen.

User Interaction and Interface: The score is displayed in a prominent position on the screen, updating in realtime as the player progresses.

Impact on Existing Systems: Requires integration with the apple consumption and gameover functionalities.

Priority and Timeline: Medium priority. Implemented in the third week.

Acceptance Criteria: Scores should update correctly and display without lag.

Feedback and Iteration: User feedback will ensure the score display is clear and informative.

5. Game Over

Description: A gameover screen appears when the player loses.

Purpose: To notify the player of the game end and display the final score.

User Interaction and Interface: A gameover message and restart button will be displayed.

Impact on Existing Systems: Resets all game states and variables, including the snake's length and position.

Priority and Timeline: Medium priority. Development and testing completed by the end of the third week.

Acceptance Criteria: The gameover screen should display correctly and allow the player to restart the game.

Feedback and Iteration: Player testing will ensure the game over functions works properly.

User Interaction and Interface

The user interface is designed to be intuitive, with clear visual indicators for the snake, apples, score, and gameover state. Controls are simple and accessible, ensuring players can easily navigate the game.

Impact on Existing Systems

There are no existing systems to integrate with, as this is a standalone game project. However, internal consistency and smooth integration of functionalities are crucial.

Priority and Timeline

1. Snake Movement Control: High priority, 1st week.
2. Apple Generation: High priority, 1st week.
3. Collision Detection: High priority, 2nd week.
4. Score Keeping: Medium priority, 3rd week.
5. Game Over: Medium priority, 3rd week.

Acceptance Criteria

Smooth and responsive snake movement.

Correct and random apple generation.

Accurate collision detection and gameover handling.

Realtime score updates.

Seamless gameover functionality.

Feedback and Iteration

Continuous feedback from test players will be encouraged to refine the game mechanics and user interface. Iterative development will allow for adjustments based on practical considerations and player input, ensuring a polished final product.