ICS 168 Requirements Document

Members

Kathy Chiang	55935182
Victor Hsiao	82542906
William Yang	83111331

Game Choice

Two-player Snake

Gameplay

- Two players each control one Snake object.
- One point is received with each Food object consumed by a Snake.
- Snakes must avoid Boundary objects (walls, their own tails, or the other Snake) or die.
- The objective is to get the most points and avoid dying.
    - Dying incapacitates the Snake for a few seconds.
    - This allows the enemy Snake to gain points with no competition.
- The game ends when one Snake reaches the target score (for example, first to 25).

Future/Optional Plans

- Some Food objects may have power-up effects.
    - This will make them more valuable than normal Food and highly contested.
    - Examples: Increase Snake speed, decrease tail length, slow the opponent
- There may be a separate game mode where, instead of a target score, the game lasts for a set amount of seconds (for example, a 30-second death match).

Game Objects

Snakes, Food, Boundaries, and Scenes (game screens) are the main objects of this game.

Snakes

Snakes start with one head piece that gains one tail piece for each Food object consumed by the head. Food objects are consumed when the head collides with the Food object.

When a Snake dies, a Ghost Snake remains. The Ghost Snake will not be considered a Boundary and cannot move (it just marks the placed where the Snake died). After a set amount of seconds (2-3), the Snake respawns and continues moving..

Snakes can be controlled to move North, South, East, and West (WASD or Arrow Keys).

Food
Static Food objects will spawn at random locations on the screen. When a Snake eats a Food object, the Food object will disappear and the Snake will gain one tail piece and one point to the player's score.

Boundaries
The game Boundaries (walls) are set by static horizontal and vertical lines which determine the width and height of the available space for Snakes to move. Snakes are also considered Boundary objects. A Snake dies when it runs into any Boundary.

Scenes

- Waiting for Log-In
    - The game will wait while the user logs in via web interface.
    - Upon logging in, the game will transition to Game Mode Selection.
- Game Mode Selection
    - The first version of the game has the standard score-based game mode.
    - *Future versions may include a Time-based game mode.*
    - Upon selecting a mode, the game will move to the Connection screen.
- Connection/Game Start
    - This screen will attempt to match two logged-in players to the same match.
        - Once connected, the screen will show the username (*and optionally, win history and other statistics*) for each player and a Ready button.
        - The game begins (transitions to the Game scene) when both players are ready.
- Game
    - The Game screen contains all the typical game objects. Each player will control their Snake. The game transitions to the Game End screen when the game end conditions are met.
- Game End
    - When the game ends, this screen will display the score for each player.
    - The player's statistics will also be updated (win/loss, average score, etc).

Multi-user Features

The game will support two players randomly matched in order of log-in.
Users will need to log in through a web interface to play the game.

The local version will have separate control schemes for each Snake (WASD and Arrow Keys).
The networked version will have one control scheme (WASD) and each player controls their own Snake.

Adding a Snake to the classic game creates a new dynamic where players need to dodge more objects, and there is more emphasis on the efficiency of acquiring Food.

In a one-player Snake game, the game will end when the Snake dies by colliding into a Boundary. In a multiuser Snake game, however, the game must continue. Therefore, our version rewards Snakes that avoid death since they will be able to acquire more points than an incapacitated Snake.

Data Stored
Each player needs to be kept in a database with their username and password.
Optionally, for each player, we can store game statistics such as wins, losses, average score, and average number of deaths.

In each individual match, we need to keep track of these things for each player:

- The player's score
- The player's death state
- The player's tail length

Optionally, we can record:

- The time elapsed for the time-based mode
- The number of deaths

Network Interactions
Our basic idea is to have the client receive all the information from the local player, then send that information to the server to keep track of the data and update both players on the current state of the match. The client will take care of screen transitions. The server will keep track of all user information and statistics in a database.

Some examples:

<u>Log-in</u>
Client receives username and password information from user
Client sends information to Server
Server checks the information with the database
Server sends Client accept or deny
Client relays accept or deny to user
If accepted, Client will move the user to the right screen

During a Match

Server keeps track of game state for both players

Client updates game state locally and sends data to Server

- Including Snake location, player score, player dead/alive boolean

Server updates the game state and sends it back to both players

Server takes care of most game logic (such as checking for collisions)