

Name: Paulo Roberto Xavier da Silva

Test plan for the snake game

Objective: Verify the functionality and usability of the snake game. Check possible bugs or problems in the system.

Scope: The test will track the main functionalities of the game including, the scores increment, the increasement of the snake by eating a fruit, the probability of the snake dying by touching the wall or touching itself, the fruits appearing in random places, the movement of the snake based on the keyboard pressing and the user interface.

Test strategy: Execution of unit tests, integration tests and system tests by focusing in each specific functionality of the system, including the interface and the basic functionalities. Each functionality will be tested separately from others and then, the last part of the test will be the system overall.

Tests scenario:

Movement: Test if the snake moves according to the key pressed by the user.

Scores: Check the scores of the system according to the numbers of fruits eaten by the snake. Verify if, after eating a fruit, the scores increase 1 point.

Snake size: Check if the snake size increase after eating a fruit. The recommended is to increase 1 px per fruit eaten.

Collision: Verify if the snake dies by touching itself or the edge wall. After touching one of these things, the game must stop and a game over message must appear in the user screen.

Fruits spawn: Check if the fruits appear in random places after the snake eating one of them.

Interface: Test the user interface. The interface must be friendly and easy-to-use. Verify if the components are positioned correctly.

Resources: A team of 3 testers and a Test Environment with access of the snake game code.

Test environment:

Operational system: Windows, macOS, Linux.

Browsers: Google Chrome, Mozilla Firefox, Microsoft Edge, Opera.

Devices: Desktop computers and laptops.

Test Schedules

Beginning of the tests: 11.27.2023 (MM.DD.YYYY)

Ending of the tests: 12.07.2023 (MM.DD.YYYY)

Acceptance criteria: 97% of the tests must be approved. The interface has to be easy-to-navigate and intuitive. Absence of critic defects that impact the main functionality of the game, including movement, scores and deaths.

Risks and mitigations:

Risks: Retardment in the development or execution of a specific task or functionality.

Mitigations: Communicate the development team and the test team, including the P.O or other leadership roles within the company.

Output criteria: All the tests must be executed. Final report, including the tests results and other important things about each test individually. If necessary, organize a meeting and communicate with the leaders.