CS 5410 – Mini-Game Test

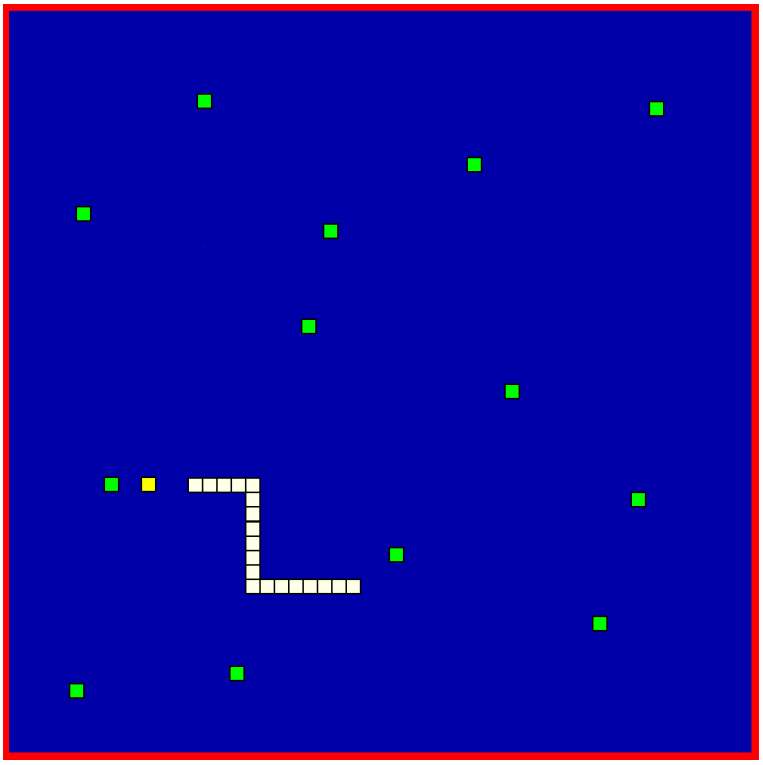
Introduction

A mini game is a game that lasts only a few seconds to a few minutes per play. Nintendo released an entire "game" made up of mini games called, "WarioWare". Each mini game lasted at most 5 seconds; actually, these are called micro games.

Mini Game

You are going to create an HTML5 web-based version of a game that I used to play in my childhood on my TRS-80. The TRS-80 has limited graphics, 128 x 48 (rectangular) pixels. While it was limited in capability, some good games were made for it, including a snake style game.

Create a mini game that follows the mock-up shown below and meets the specified gameplay and technical requirements.

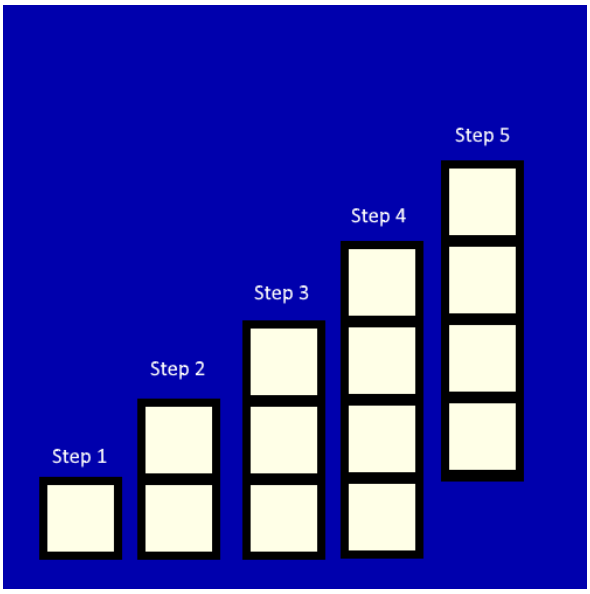


- The green squares are obstacles
- The yellow square is the food
- The white squares are the snake segments
- The red is the arena border
- The blue is the arena background

Gameplay Requirements

The goal of the game is to build the longest snake possible. The game starts with the snake of a single block in size and growing by 3 blocks each time a piece of food is eaten. If the snake runs into the border of the arena, an obstacle, or itself, the game ends. The score for the game is how many segments long the snake is when the game completes.

- There are 10 single square obstacles at the start of the game. These obstacles are randomly placed throughout the arena, and can not occupy the same location.
- The initial size of the snake is 1 segment. The snake does not start on an obstacle or a piece of food.
  - The snake initially is not moving, it only starts moving once the player presses an arrow key in the direction they want it to move. Once started, the snake never stops moving, until the game ends.
- There is always 1 piece of food available.
  - The location of a piece of food is randomly chosen.
    - A piece of food can never be on an obstacle.
    - A piece of food can never be located on the snake.
  - As soon as a piece of food is eaten, a new one created.
- When a piece of food is eaten, it adds three segments to the snake. This is done by continuing to move the snake in whichever direction it is moving and adding the segments, one at a time, at the end of the snake as the tail moves forward. In other words, the snake continues to move as normal, but the tail won't appear to move for three times while the three new segments are added, one at a time where the tail is. The next image illustrates how the snake would grow/move after eating its first piece of food.
  - Step 1, the food is eaten.
  - Step 2, the snake moves forward, a segment is added.
  - Step 3, the snake moves forward, a segment is added.
  - Step 4, the snake moves forward, a segment is added.
  - Step 5, the snake moves forward.
  - **Important note:** The player can still change the direction of the head, even as new segments are being added at the spot where the food was consumed.



- You can choose the speed the snake moves, find the right balance of challenge, not boringly slow and not blisteringly difficult.

Technical Requirements

- HTML5 Canvas Rendering & JavaScript based; as we have been doing all semester.
- Menu and game screen system
  - New Game
  - High Scores
  - Credits
- Visual Components
  - I have provided a mock-up, you are welcome to use other (not horrible) colors.
  - Rendering of the squares must have a black border around them in order to help them stand out properly from the background. The mock-up uses and demonstrates this.
  - The snake should have a segmented look as the mock-up shows; if the square rendering is doe as described above, it will appear this way.
  - Game start countdown 3, 2, 1
- High scores must persist to the browser's local storage; keep the top 5 scores.

Grading Breakdown

- Menus: 10%
- Scoring: 5%
- High scores: 15 % (Requires scoring)
- Game start countdown: 10%
- Random placement of obstacles: 10%
- Random placement of food: 10%
- Snake length increase when food is eaten: 15%
- Snake movement: 10%
- Rendering of… 10%
  - Arena background and borders
  - Obstacles
  - Food
  - Snake