Design Overview for Snake Game

Name: Luong Trac Duc Anh
Student ID: 103488117

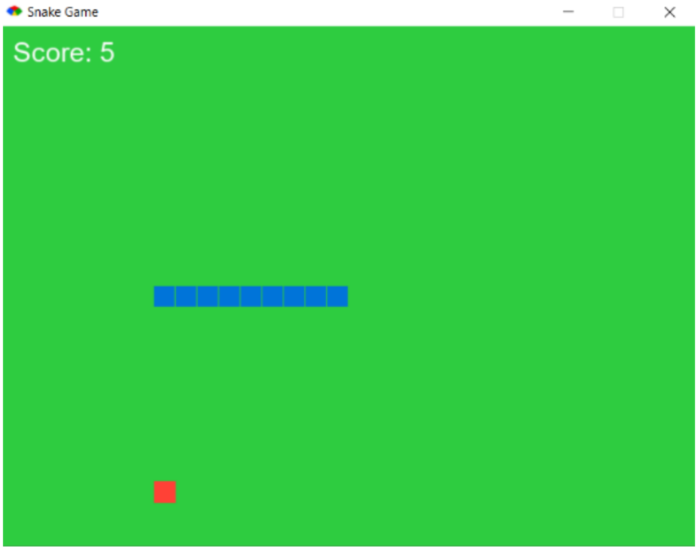
Summary of Program

This is an attempt to recreate the famous snake game that took the world by storm in the late 90s and early 00s. The game became and icon for cell phone users when the touch screen wasn't not yet introduced.

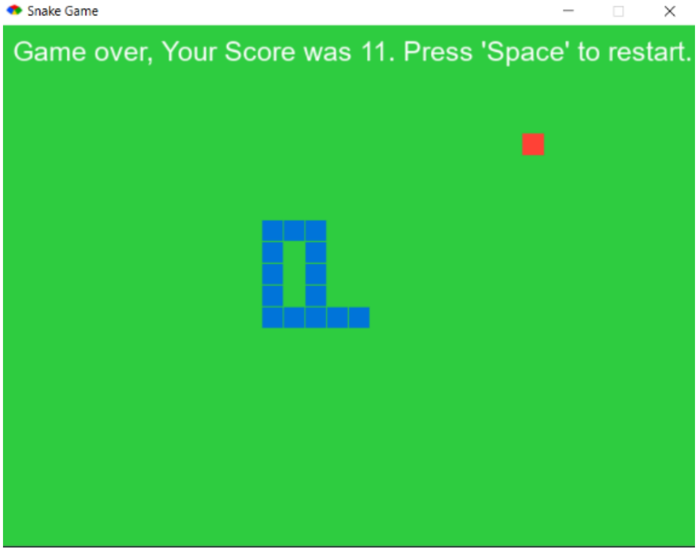
The game is fairly simple and self-explanatory. As the player, you will be a snake and try to collect as many food pieces (red square) as possible to increase your score. As this is a 2D game, your option to move in the game window will simply be "up", "down", "left", and "right", with the input being the arrow buttons on your keyboard. Each time you gain a score (eat 1 ball), you your length will be increment by 1. If the snake happens to hit itself, the game will be over with your score displayed and the option to press the "space" button to restart for a new round.

The game will use the "ruby2d" library to draw out the Graphical User Interface (GUI). As long as you have ruby2d (install by executing the command "gem install ruby2d" in your Command Prompt, with Ruby already downloaded on your computer) and the snake.rb file in your local computer, the program can run on its own without the need for external media files or other ruby libraries.

Sample screenshot of the game:



Game-over screen:



Game over, Your Score was 11. Press 'Space' to restart.

Required Data Types

Describe each of the records and enumerations you will create using the following table (one per record).

Table 1: Snake (player) details

Field Name	Type	Notes
:positions	array	Position of the snake when the game initializes
:direction	string	Control how the snake moves in the game window
:growing	Boolean	Increment both the score and the snake length by 1 (block)
:new_direction	string	Stop the snake from going through its body

Table 2: Ball details

Field Name	Type	Notes
:ball_x	integer	x-position of the ball
:ball_y	integer	y-position of the ball

Table 3: Game details

Field Name	Type	Notes
:score	integer	Player’s score obtain through eating the ball
:finished	Boolean	Begin/end the game

Overview of Program Structure

Functions:

Class Snake:

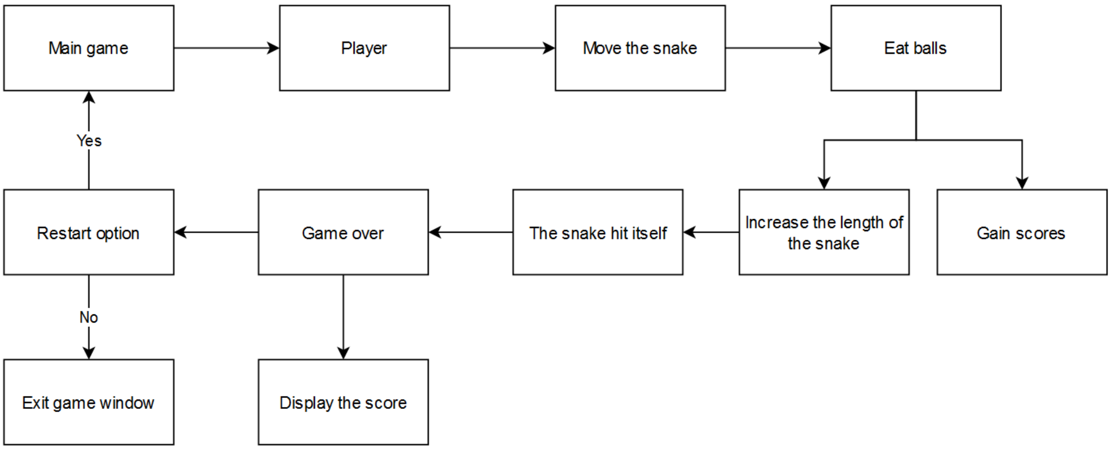
- initialize: place the snake in the default position when starting the game, stop the snake from growing at the beginning
- draw: draw out the snake and the ball (red square) with size, position, and colour
- grow: Increment the length of the snake
- can_change_direction_to?: stop the snake from moving through itself
- x and y: coordination
- next_position: draw out a new square at the snake head and remove 1 from its tail from each frame to get the moving animation
- hit_itself?: end the game when the length of the position is different from its unique position length
- new_coords: move the snake to the next position
- head: define the last positions of the snake

Class Game:

- initialize: place the ball at a fixed position to stop overlapping with the snake
- draw: draw out a square that represent the ball
- snake_hit_ball?: calculate the position of the ball at contact point
- record_hit: count the score for the player
- finish: execute the Finished? function
- finished?: end the game
- text_message: prompt the game-over text, the score obtained by the player, and how to restart the game for a new round

(The Structure chart is displayed in the next page)

Structure chart



Main game
Player
Move the snake
Eat balls
Yes
Restart option
Game over
The snake hit itself
Increase the length of the snake
Gain scores
No
Exit game window
Display the score