CS152-Project

Deepesh Meena (150050041)
Anshu Ahirwar (150050077)
Siddharth Naik(160050113)

April 26, 2018

Snake Game

1 Description

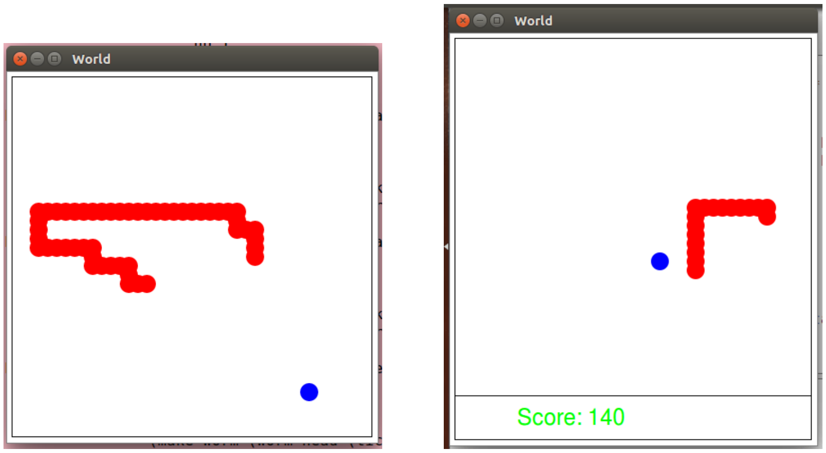
Snake is a classic video game where the player maneuvers the line which grows in length after eating the food , with the line itself being a primary obstacle and other obstacles are walls. Each item eaten makes the snake longer,so controlling is progressively more difficult.

2 Idea of the solution

In the game we have represented snake as a **structure** which contains **Head, Tail and Direction**. Head represents head of snake and tail represents rest of the snake where head has some direction (up,down,left,right) assigned to it.
Whenever the user moves the snake or the snake eats a fruit, these **members** are appropriately updated. Even when nothing happens, these members are updated so that the snake moves forward.

When the snake eats the fruit we will update Snake's head and tail position. We will create a huge box of some width and height and it is divided into some grids, some cells represent Snake's position and Fruit's location.
Fruit's location is changed using random function inside the box. We also check that fruit doesn't spawn on the snake.

Now we have created a function which checks if the snake has hit himself. This function is checking whether snake's **head's position**



(a) snake-1

(b) snake-2

Figure 1: Snake Game

is equal to the some part of the rest of snake(tail).

We have also created a function which checks if snake hits a wall. This function checks if the snake's head's position is outside the **box's dimensions**.

We have written a function to change the direction of the snake. We have used **KeyEvent** to change the direction of the head. Which takes input from the keyboard and changes the direction accordingly.We have used the **(require 2htdp/universe) racket library** for this purpose.

We have used the **(require 2htdp/image) racket library** to create simple snake and food on canvas. .
Finally we are showing **live score** whenever the snake eats food score gets update on the window. We used set! for this purpose

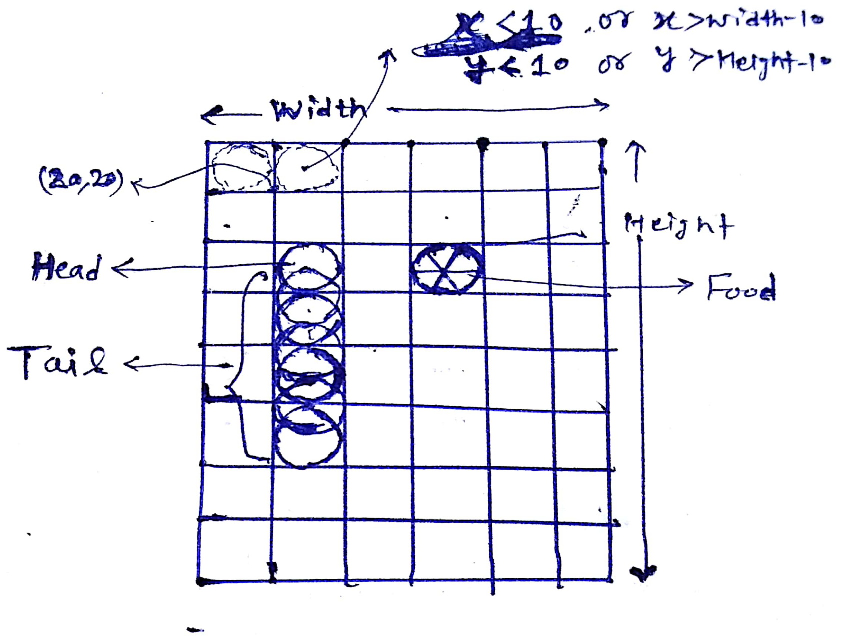


Fig: Snake Game

3 Sample input output

We will check the functionality of functions using **check-expect** statements. For every test-case we will change the position of snake and if if hits the wall or itself or also check it manually every time to make game more bug-free.

4 Limitations and bugs

Snake, being a game without any AI, has no major limitations but only limitation is creating different levels.