RAYHAN RAMANDA

2501982582

COMPUTER SCIENCE

FINAL PROJECT REPORT

1. Snake Game Introduction

“Snake” is a game where we control a line and are prompted to eat many objects (or apples as depicted on this program) and grow. It began in 1976 as a dual player game, and became popular with early Nokia handphones.



Pic 1.1 *Snake* on Telmac 1800, CHIP-8



Pic 1.2 *Snake* in one of the Nokia handphones.

1. Program and assets used

I am only using JAVA and some of its extensions. and some simple pictures that represent snakes' heads, bodies and apples to eat.

 Pic 2.1 This red dot is used as the head.

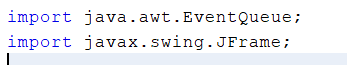
Pic 2.2 This green dot used as the body, and it added and formed a line each apple eaten by one.

 Pic 2.3 This apple used as the objective to be eaten.

1. Code explanation.

This game splitted by two files, board.java used as the game configurations, and SNAKE.java as the main driver.

First off, let's explain some of the code that was coded in SNAKE.java.

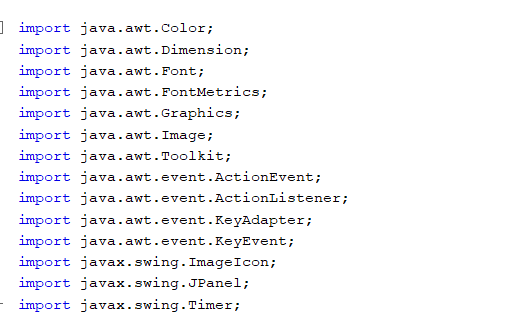


Pic 3.1 These two lines are used for calling the UI, without this, the game

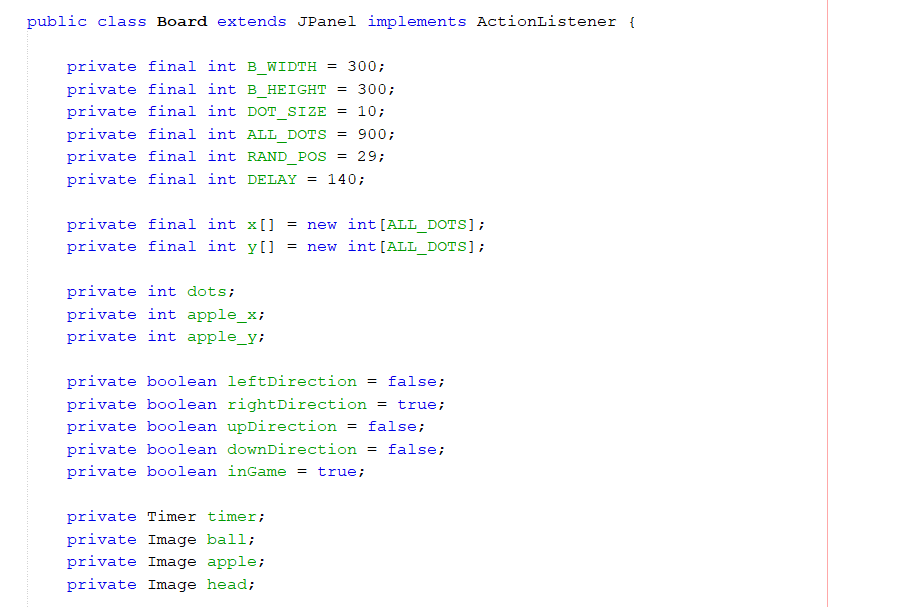
screen wont spawn/appear.

 Pic 3.2 All that we need to calling Board.java to be on the air

Now, let's see the Board.java section.

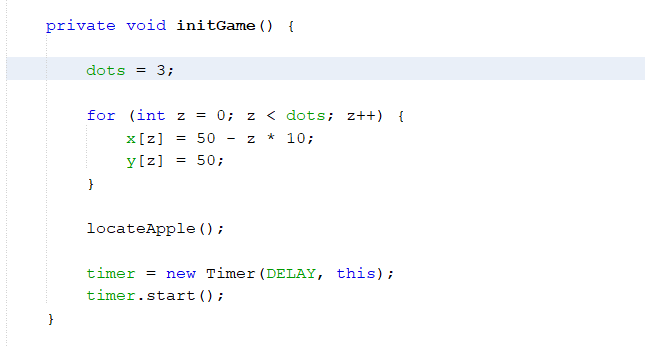


Pic 3.3 All of the modules that I need to properly spawn the board when SNAKE.java calls.

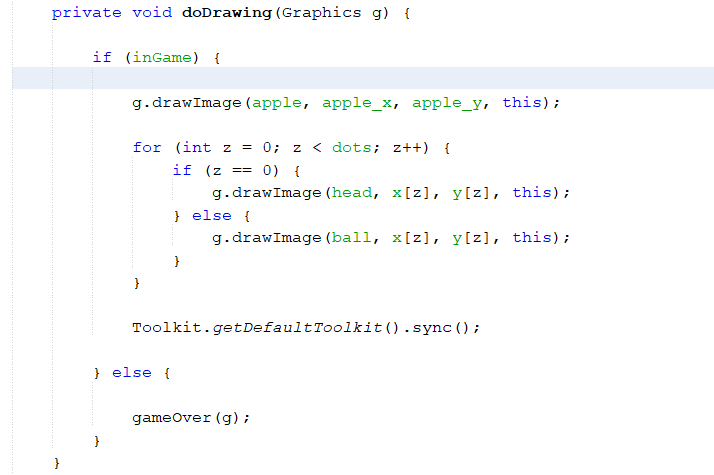


Pic 3.4 These all determine the size of the board and define maximum numbers of possible dots on the board. The DELAY constant determines the speed of the game while ALL\_DOTS coordinates all of the snake's body joints.

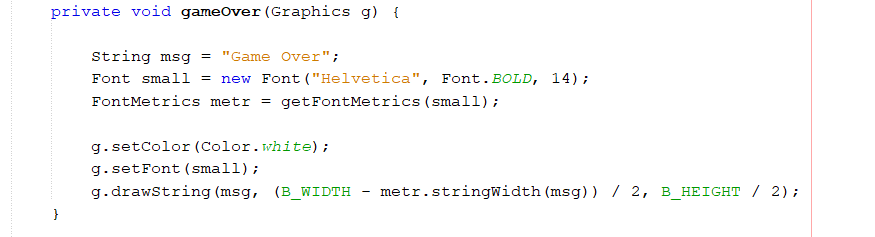
Pic 3.5 All we need to properly spawn the art assets, and load the board with black background.



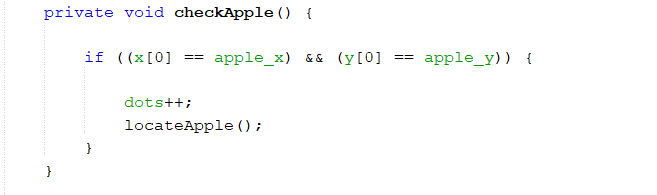
Pic 3.6 The codes we need to randomly spawn the snake and randomly put apple on the board



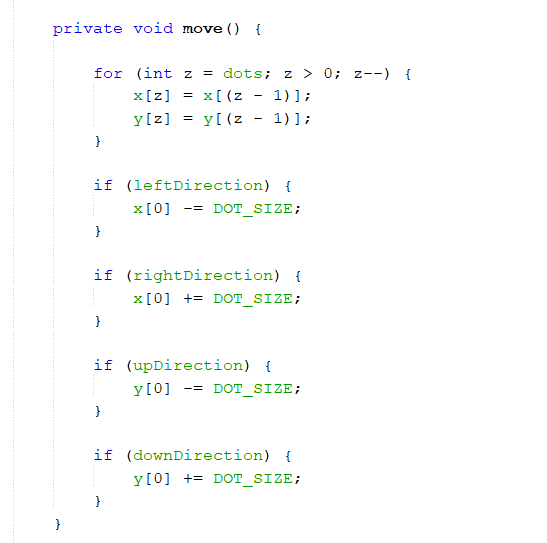
Pic 3.7 These are all for drawing each frame of the game when the snake moves and apples are eaten.



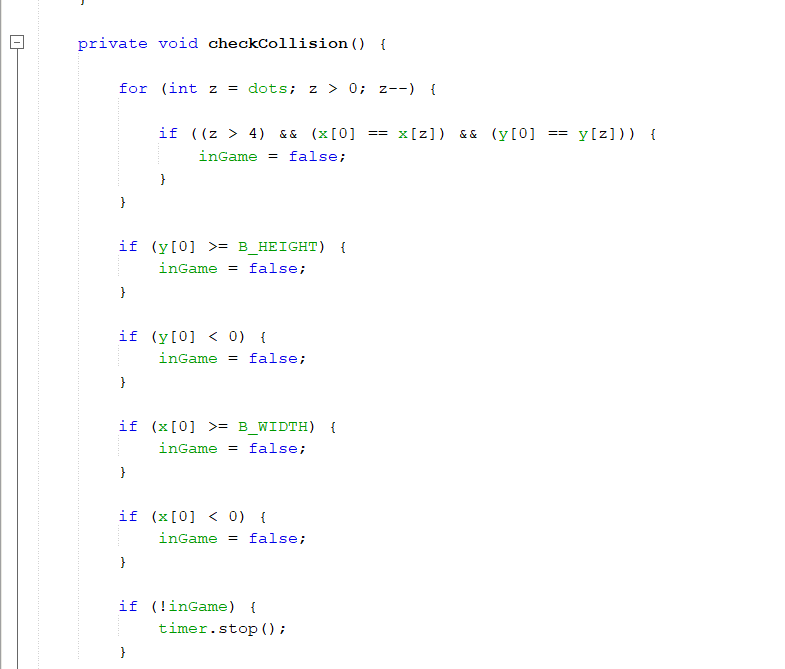
Pic 3.8 The game over screen coded in here.



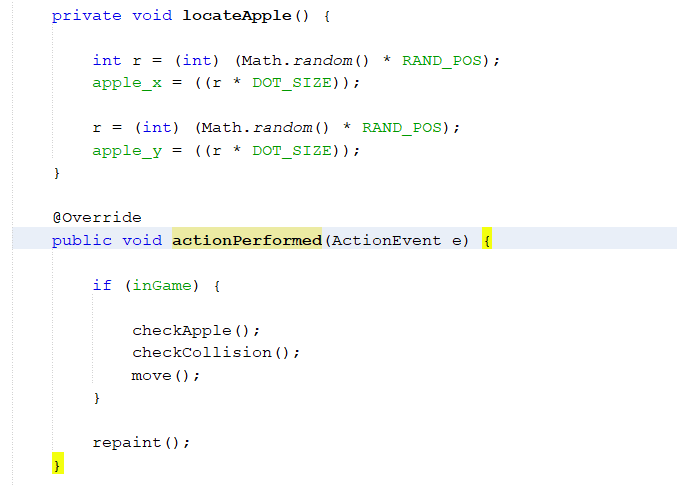
Pic 3.9 When the apples hit the snake's head, this code line performs its action to add one dot to the snake's body.



Pic 3.10 To move the snake’s direction.



Pic 3.11 To check when the snake is colliding with its own body or the screen frame and stops the game.

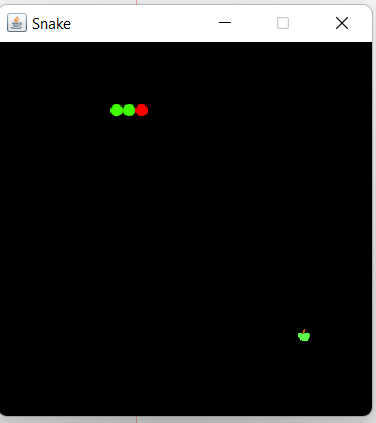


Pic 3.12 locateApple function is to calculate where the apple position to be spawned and actionPerformed is calling both checkApple and collision.



3.13 Finally the final part, these are for reading the input from arrow keys and moving the snake according to input.

1. Final Result



Pic 4.1 This is the result, as you can see, there's a snake with red head and green dots as body and apple as objective.