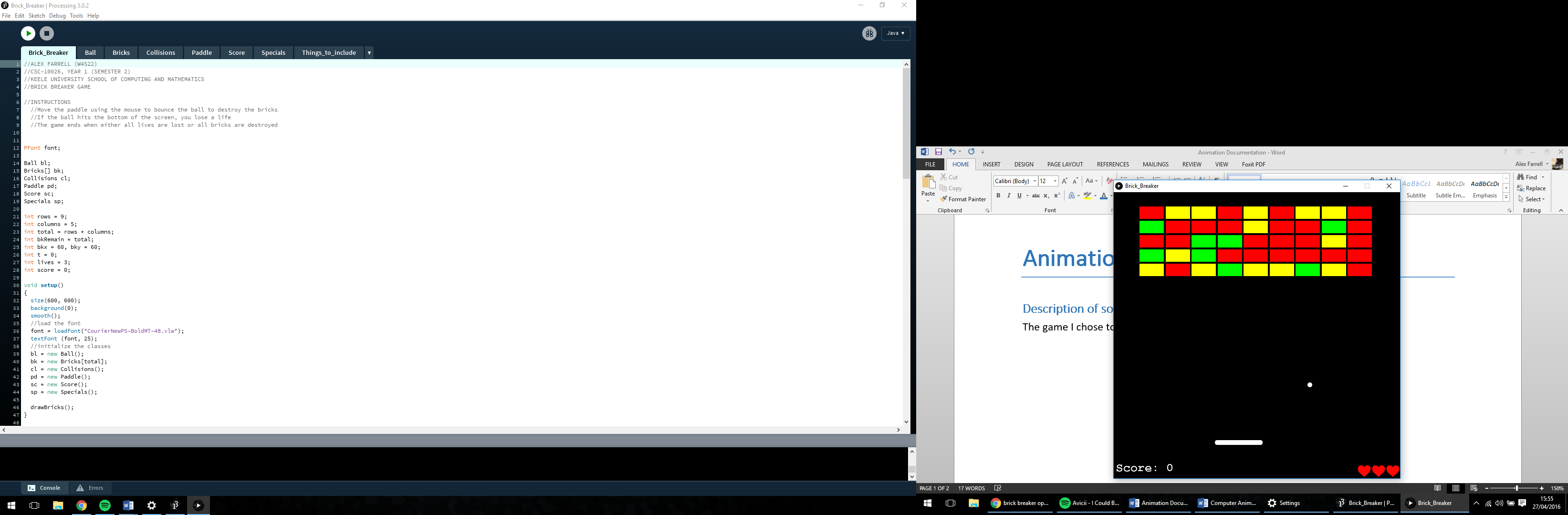
**Animation Documentation**

# Description of solution

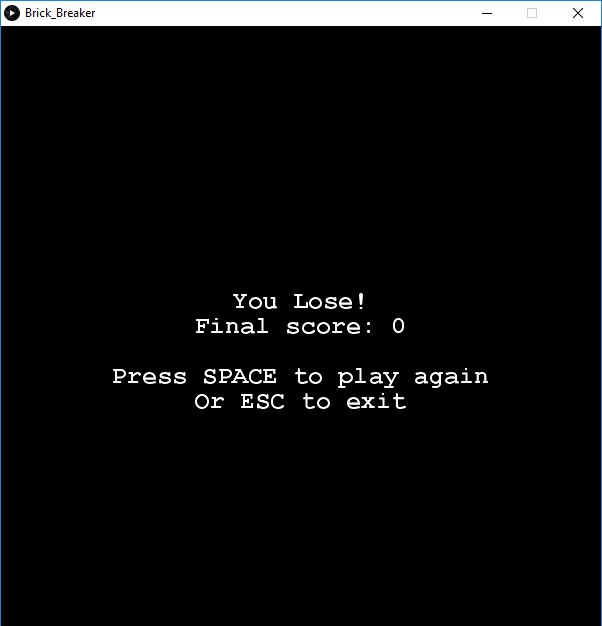
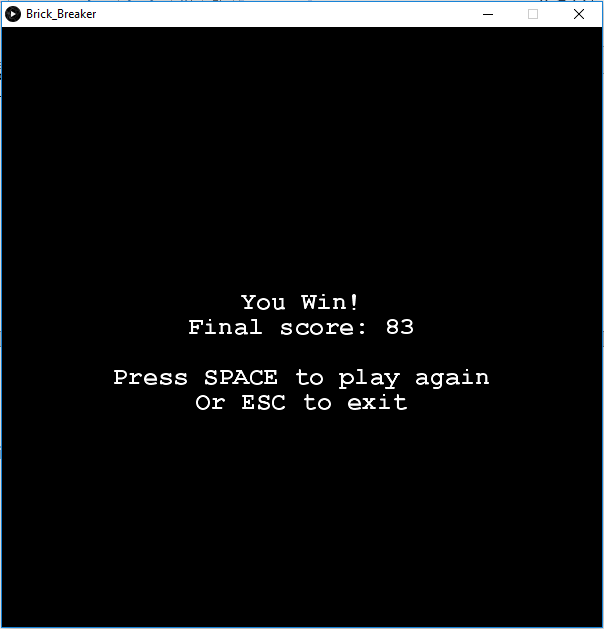


The game I chose to create is Brick Breaker. As you can see, the interface is pretty simple to understand. There are five rows of bricks, each of which employs one of three states:

* Green: Takes one hit to destroy it.
* Yellow: Takes two hits to destroy it.
* Red: Takes three hits to destroy it.

The other main elements of the game are the ball and the paddle. The ball bounces off the paddle, bricks and walls. The paddle is moved using the mouse, i.e. move the mouse where the player wants the paddle to go. If the paddle misses the ball and comes into contact with the bottom of the screen, a life is lost. The number of lives that the player has is displayed in the bottom right corner of the screen in the form of red hearts. Every time a life is lost, one of these hearts disappear. The score is displayed in the bottom left corner of the screen. Every time the ball hits one of the bricks, the player gains one point. If a brick is destroyed, on occasion, the size of the paddle may change. It is unknown until it happens as to whether the size of the paddle increases, decreases or stays the same. Every time a life is lost, the size of the paddle is reset.

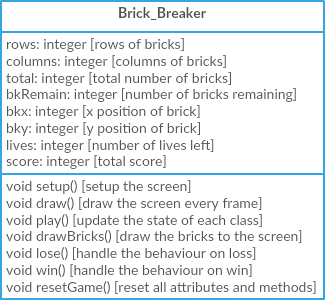
The end of the game is determined when either all lives have been lost or all the bricks have been destroyed. At this point, the user can either choose to play again or exit the game using the ‘space’ and ‘esc’ key respectively.

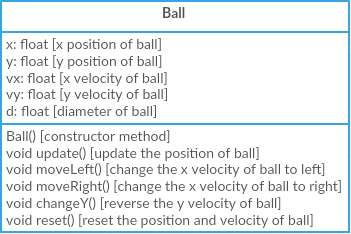
# Architecture & Structure

Below are class diagrams of each class. Each diagram contains the class name, attributes specified and methods implemented.

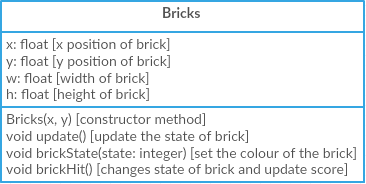
## Main Class

The main class is where all the methods are called to so that the game can be played.

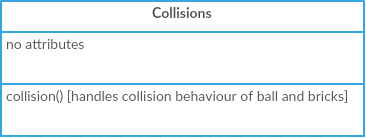
## Ball Class

The ball class handles the behaviour of the ball. ‘update()’ is called every frame in the main class. ‘moveLeft()’, ‘moveRight()’ and ‘changeY()’ are called every time the ball hits any other object. ‘reset()’ is called whenever the ball hits the bottom wall (indicating that a life is lost).

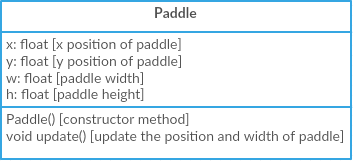
## Bricks Class

The bricks class handles the behaviour of each brick. ‘update()’ is called every frame in the main class. ‘brickState()’ is called at the start to set the initial state of the brick, as well as throughout the game whenever the ball comes into contact with a brick. The latter is called through ‘brickHit()’.

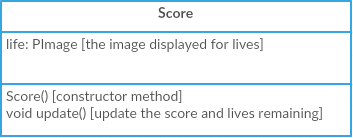
## Collisions Class

The collisions class handles the behaviour of the ball and bricks every time they come into contact with each other or another object such as a wall or the paddle.

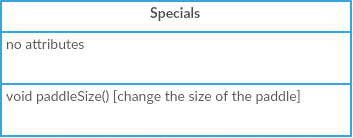
## Paddle Class

The paddle class handles the behaviour of the paddle. The x position of the paddle is determined by the mouse x minus half the width of the paddle. This ensures that the centre of the paddle is at the x position of the mouse. ‘update()’ is called every frame in the main class.

## Score Class

The score class handles the score and number of lives. Every time a brick is hit, the score is increased and every time the paddle misses the ball, the lives are decreased. ‘update()’ is called every frame in the main class.

## Specials Class

The specials class handles the behaviour every time a brick is destroyed and it is determined that a special ability will be called. It is random on every time of calling, as to whether the size will grow, shrink or stay the same.