

Homework 10: Jack – Square Dance

Requirements:

Build the Square Dance application, in Jack, per the instructions and guidance covered in class.

Grading method:

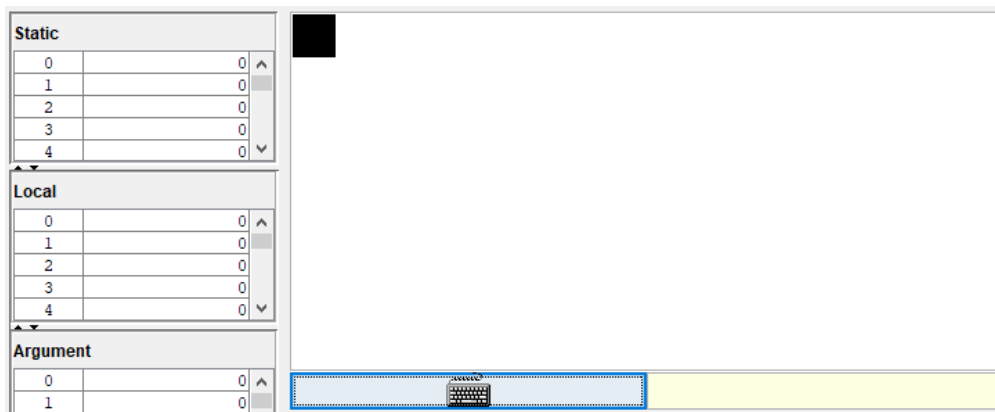
As usual with programming assignments, we look for elegance, clarity, reasonable documentation, and neatness.

Follow the instructions in lecture as far the classes and methods to build, as well as allowing command-line arguments as instructed. Document each method (description, precondition, postcondition) and add author information on each file. Provide an algorithm for your main method that drives the Square Dance application.

What do you turn in?

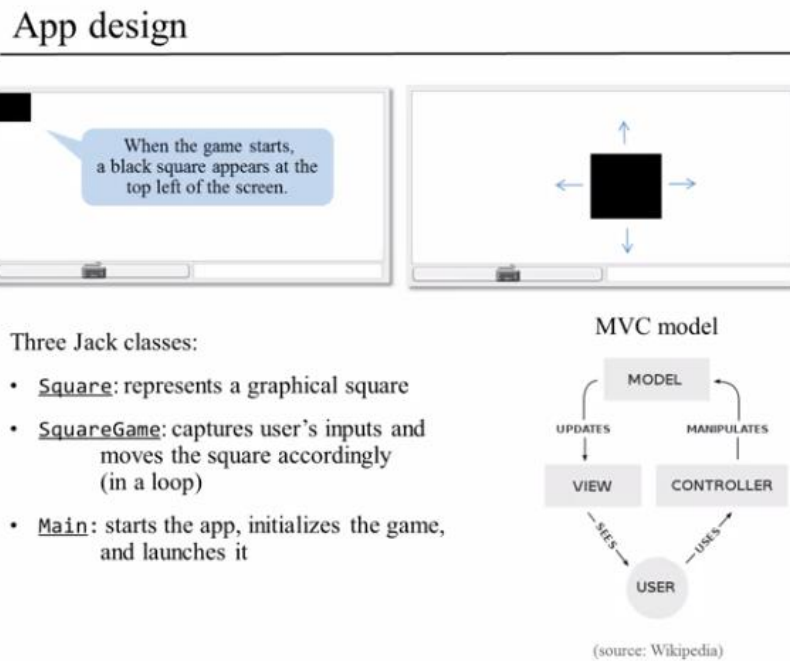
Create one Word document (or PDF) with the following in order:

1. The **Square.jack** source code (documented)
2. The **SquareGame.jack** source code (documented)
3. The **Main.jack** (with main function) source code (documented)
4. Create a screen shot of the application running in the VM Emulator (from the Nand2Tetris software package). The square should be somewhere other than the top-left ($x=0, y=0$) portion of the screen.



<i>Square Dance</i>	<i>Working?</i>
Working?	/ 60
Well built?	/ 30
Subtotal	/ 90
Documentation	/ 100

Below is the API documentation depicting the high-level details of the Square Dance application:



Square class API

Square API

```

/** Implements a graphical square. */
class Square {

    /** Constructs a new square with a given location and size */
    constructor Square new(int Ax, int Ay, int Asize)

    /** Disposes this square */
    method void dispose()

    /** Draws this square on the screen */
    method void draw()

    /** Erases this square from the screen */
    method void erase()

    /** Increments this square's size by 2 pixels */
    method void incSize()

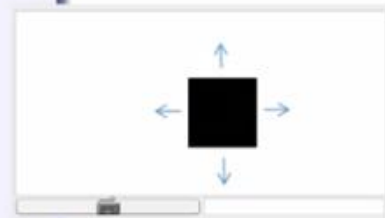
    /** Decrements this square's size by 2 pixels */
    method void decSize()

    /** Moves this square up by 2 pixels */
    method void moveUp()

    /** Moves this square down by 2 pixels */
    method void moveDown()

    /** Moves this square left by 2 pixels */
    method void moveLeft()

    /** Moves this square right by 2 pixels */
    method void moveRight()
}
  
```

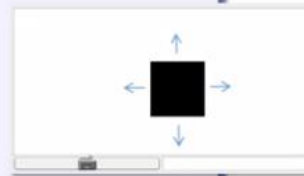


SquareGame class

SquareGame.jack

```
/** Implements a square game. */
class SquareGame {

    field Square square; // the square of this game
    field int direction; // the square's current direction:
                        // 0=none, 1=up, 2=down, 3=left, 4=right
}
```



The **SquareGame** class should have methods to:

SquareGame constructor - initializes the Square to $x=0$, $y=0$, $size=30$, $direction = 0$

void incSize() - increments the square size by 2 pixels

void decSize() - decrements the square size by 2 pixels

void run() - waits for a key to be pressed, then responds to the following:

- If key is 'q', exit the game
- If key is 'z', decrement the size of the square
- If key is 'x', increase the size of the square
- If key is 'up arrow', move up
- If key is 'down arrow', move down
- If key is 'left arrow', move left
- If key is 'right arrow', move right

void moveSquare() - moves the square according to its direction and delays for 5 ms.

void dispose() - disposes of the Square and de-allocates this.