Raul Aguilar

Professor Paulding

CS 220 2148

November 29, 2020

Homework 9: High Level Language

Square.jack

```
class Square {
    field int x, y; // Screen location of the square's top-left cornor
    field int size; // Length of square, in pixel

    constructor Square new(int xV, int yV, int sizeV) {
        let x = xV;
        let y = yV;
        let size = sizeV;
        do draw();
        return this;
    }

    // Disposes current square object
    method void dispose() {
        do Memory.deAlloc(this);
        return;
    }

    // Draws the square on the screen
    method void draw() {
        do Screen.setColor(true);
        do Screen.drawRectangle(x, y, x+size, y+size);
        return;
    }

    // Erases square from screen
    method void erase() {
        do Screen.setColor(false);
        do Screen.drawRectangle(x, y, x+size, y+size);
        return;
    }

    // Increments the square size by 2 pixels
    method void incSize() {
        if(((y+size) < 254) & ((x+size) < 510)) {
            do erase();
```

```
            let size = size + 2;
            do draw();
        }
        return;
    }

    // Decrements the square size by 2 pixels
    method void decSize() {
        if(size > 2) {
            do erase();
            let size = size - 2;
            do draw();
        }
        return;
    }

    // Moves the square up 2 pixels
    method void moveUp() {
        if(y > 1) {
            do Screen.setColor(false);
            do Screen.drawRectangle(x, (y+size)-1, x+size, y+size);
            let y = y - 2;
            do Screen.setColor(true);
            do Screen.drawRectangle(x, y, x+size, y+1);
        }
        return;
    }


    // Move square down 2 pixels
    method void moveDown() {
        if(y+size < 254) {
            do Screen.setColor(false);
            do Screen.drawRectangle(x, y, x+size, y+1);
            let y = y + 2;
            do Screen.setColor(true);
            do Screen.drawRectangle(x, (y+size)-1, x+size, y+size);
        }
        return;
    }

    // Move square down 2 pixels
    method void moveLeft() {
        if(x > 1) {
            do Screen.setColor(false);
```

```
            do Screen.drawRectangle((x+size)-1, y, x+size, y+size);
            let x = x - 2;
            do Screen.setColor(true);
            do Screen.drawRectangle(x, y, x+size, y+size);
        }
        return;
    }

        // Move square down 2 pixels
    method void moveRight() {
        if(x+size < 510) {
            do Screen.setColor(false);
            do Screen.drawRectangle(x, y, x+1, y+size);
            let x = x + 2;
            do Screen.setColor(true);
            do Screen.drawRectangle(x, y, x+size, y+size);
        }
        return;
    }
}
```

SquareGame.jack

```
class SquareGame {
    field Square square;
    field int direction;

    constructor SquareGame new() {
        let square = Square.new(0, 0, 30);
        let direction = 0;
        return this;
    }

    // Dispose of game
    method void dispose() {
        do square.dispose();
        do Memory.deAlloc(this);
        return;
    }

    // Move the square
    method void moveSquare() {
        if(direction = 1) { do square.moveUp(); }
        if(direction = 2) { do square.moveDown(); }
        if(direction = 3) { do square.moveLeft(); }
        if(direction = 4) { do square.moveRight(); }

        do Sys.wait(5);
        return;
    }

    // Run the game
    method void run() {
        var char key;
        var boolen exit;
        let exit = false;

        while(~exit) {
            // Waits for a key to be pressed
            while(key = 0) {
                let key = Keyboard.keyPressed();
                do moveSquare();
            }
            if(key = 81) { let exit = true; }       // q key
            if(key = 90) { do square.decSize(); }   // z key
            if(key = 88) { do square.incSize(); }   // x key
```

```
            if(key = 131) { let direction = 1; }    // up arrow
            if(key = 133) { let direction = 2; }    // down arrow
            if(key = 130) { let direction = 3; }    // left arrow
            if(key = 132) { let direction = 4; }    // right arrow

            // Waits for key to be released
            while(~(key = 0)) {
                let key = Keyboard.keyPressed();
                do moveSquare();
            }
        }
        return;
    }
}
```

Main.jack

```
class Main {
    function void main() {
        var SquareGame game;
        let game = SquareGame.new();
        do game.run();
        do game.dispose();
        return;
    }
}
```