Mario Bros

ft. handsome squidward

Krish S, Nancy K, Tammy M

THE TEAM

Nancy Kama

12th Grade

Krish Shah

12th Grade

Tammy Mathews

11th Grade

Timeline of Code Development

March 29–31

Created Trello Board and Designed Characters on PixilArt

April 4–6

Completed Main Character Class (Updates, Location, and Movement)

April 7–9

Completed 2 Obstacle Classes (Falling + Stationary)

AP Prep

May 12

Implemented Inheritance with Stationary Obstacle 2 & Finished all Character classes

Timeline of Code Development Cont.

May 16–18

Implemented Scoring with 2D array Color Change and Collision

May 19–20

Imported Music class and added sounds under calls to methods with key codes

May 22–24

Finished Coin Class and Used Randomization and Booleans to Hide and Show Coins with Collision

May 25–27

Used Booleans to create Intro and Exit Screens for our Game with Different Text and Elements on Each

Member 1: NANCY

- Project Manager - Trello Updates
- Conducted Frame.java class code to compile game
- Added to character class functions to update
- Imported Java Classes like Random and Music Classes
- Implemented 2D arrays

Member 1: NANCY

snippets from code

```
47          private void update() {
48                  x += vx;
49                  vx = ax;
50                  y += vy;
51                  vy = ay;
52
53                  if (x > 790) {
54                          x = 10;
55                          vx = 0;
56                  }
57
58                  if (x < 10) {
59                          x = 10;
60                          vx = 0;
61                  }
62
63                  if (y > 590) {
64                          y = 10;
65                          vy = 0;
66                  }
67
68                  if (y < 10) {
69                          y = 10;
70                          vy = 10;
71                  }
72
73                  tx.setToTranslation(x, y);
74                  tx.scale(0.8, 0.8);
75          }
```

```
1   import java.io.File;
2   import java.io.IOException;
3
4   import javax.sound.sampled.AudioFormat;
5   import javax.sound.sampled.AudioInputStream;
6   import javax.sound.sampled.AudioSystem;
7   import javax.sound.sampled.Clip;
8   import javax.sound.sampled.DataLine;
9   import javax.sound.sampled.LineEvent;
10  import javax.sound.sampled.LineListener;
11  import javax.sound.sampled.UnsupportedAudioFileException;
12
13  public class Music  implements Runnable  {
```

```
199         @Override
200         public void keyPressed(KeyEvent arg0) {
201                 // TODO Auto-generated method stub
202                         System.out.println(arg0.getKeyCode());
203
204                         squiddy.fall();
205                         scoreColor = scoreColors[rnd.nextInt(scoreColors.length)][rnd.nextInt(scoreColors[0].length)];
206
207                         //intro screen diss. when enter is pressed
208                         if (arg0.getKeyCode() == 10) {
209                                 gameStart = true;
210
211                         }
212
213                         if (arg0.getKeyCode() == 48) {
214                                 gameStart = false;
215                         }
216
217         }
218
219         @Override
220         public void keyReleased(KeyEvent arg0) {
221                 // TODO Auto-generated method stub
222                         squiddy.jump();
223                         soundJump.play();
224         }
```

```
18  public class Frame extends JPanel implements ActionListener, MouseListener, KeyListener {
19
20          //jelly rand vars
21          int x0 = (int)(Math.random() * 601) + 100;
22          int x01 = (int)(Math.random() * 601) + 100;
23          int x02 = (int)(Math.random() * 601) + 100;
24          int x03 = (int)(Math.random() * 601) + 100;
25
26          //CREATE THE OBJECTS (STEP 1)
27          Background      bg      = new Background(0, 0);
28          Squiddy squiddy = new Squiddy (50, 380);
29          Jellyfish jelly1 = new Jellyfish (x0, 50);
30          Jellyfish jelly2 = new Jellyfish (x01, 50);
31          Patrick patty1 = new Patrick (150, 450);
32          Patrick patty2 = new Patrick (500, 450);
33          Spongebob spongy = new Spongebob (300, 450);
34          KrabbyPatty coin1 = new KrabbyPatty(50, 20);
35          KrabbyPatty coin2 = new KrabbyPatty(600, 20);
36          KrabbyPatty coin = new KrabbyPatty(x02, 0);
37          KrabbyPatty coinn = new KrabbyPatty(x03, 0);
38
39          //color 2D array
40          Color scoreColors[][] = {{Color.RED, Color.ORANGE}, {Color.YELLOW, Color.GREEN}, {Color.BLUE, Color.MAGENTA}, {Color.LIGHT_GRAY, Color.PINK}};
41          Color scoreColor = Color.WHITE;
42          Random rnd = new Random();
43
44          //MUSIC
45          Music soundJump = new Music("Mario-jump-sound.wav", false);
46          Music soundDead = new Music("Pacman-death-sound_(1).wav", false);
47
48          //intro screen
49          boolean gameStart = false;
50
51          //coin disappearance
52          boolean coinCollect = false;
```

```
65          public void paint(Graphics g) {
66                  super.paintComponent(g);
67
68                  bg.paint(g);
69
70                  //bg if game end
71
72                  if(gameEnd) {
73                          g.setColor(scoreColors[rnd.nextInt(scoreColors.length)][rnd.nextInt(scoreColors[0].length)]);
74                          g.setFont(f2);
75                          g.drawString("Score: " + score, 340, 200);
76                          g.drawString("Slay's Acquired: " + coinCount, 290, 250);
77                          g.drawString("BooHoo You Lost", 290, 300);
78                          g.drawString("ReRun the Game to Retry", 245, 350);
79                  return;
80                  }
81
82                  //bg before game begins
83
84                  bg.paint(g);
85                  coin1.paint(g);
86                  coin2.paint(g);
87
88                  if(!gameStart) {
89                          g.setColor(Color.WHITE);
90                          g.setFont(f2);
91                          g.drawString("welcome. help material squiddy win", 187, 250);
92                          g.drawString("press enter to commence your pineapple under-the-sea adventure", 20, 300);
93                  return;
94                  }
95
96                  bg.paint(g);
97                  squiddy.paint(g);
98                  jelly1.paint(g);
99                  patty1.paint(g);
100                 patty2.paint(g);
101                 jelly2.paint(g);
102                 coin.paint(g);
103                 coinn.paint(g);
```

Member 2: TAMMY

- Contributed to Character Classes with Location Variables and Methods
- Imported files into src like music files and images
- Helped with collision between obstacles and characters

Member 2: TAMMY

snippets from code

```
public class Patrick {
        //add location attributes
                public int x2 , y2;
                //position
                private Image img;
                private AffineTransform tx;

                public Patrick(int x, int y) {
                        this.x2 = x;
                        this.y2 = y;
                        img = getImage("/imgs/emogworlpatty.png"); //load the image for Tree

                        tx = AffineTransform.getTranslateInstance(x, y );
                        init(x, y);                     //initialize the location of the image
                                                                        //use your variables
                }
```

```
import java.io.File;
import java.io.IOException;

import javax.sound.sampled.AudioFormat;
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.sound.sampled.DataLine;
import javax.sound.sampled.LineEvent;
import javax.sound.sampled.LineListener;
import javax.sound.sampled.UnsupportedAudioFileException;
```

```
114                 //set collision
115                         if (rsquiddy.intersects(rpatty1) || rsquiddy.intersects(rpatty2) || rsquiddy.intersects(rjelly) || rsquiddy.intersects(rjelly1)) {
116                                 score -= 3;
117                         }
118
119                         if (rsquiddy.intersects(rcoin) || rsquiddy.intersects(rcoinn)) {
120                                 score += 10;
121                                 coinCount ++;
122                                 coin.y3 = 0;
123                                 coinn.y3 = 10;
124                                 coin.x3 = rnd.nextInt(700 - 10 + 1) + 10;
125                                 coinn.x3 = rnd.nextInt(700 - 10 + 1) + 10;
126                         }
```

```
10  public class Squiddy {
11
12          //add location attributes
13          public int x = 50, y = 300;
14          //position of peppa
15          private Image img;
16          private int vx = 1;
17          private int ax = 3;
18          private int vy = 1;
19          private int ay = 3;
20          private AffineTransform tx;
21
22          public Squiddy(int x, int y) {
23                  this.x = x;
24                  this.y = y;
25                  img = getImage("/imgs/squiddy.png"); //load the image for Tree
26
27                  tx = AffineTransform.getTranslateInstance(x, y );
28                  init(x, y);                                 //initialize the location of the image
29                                                                          //use your variables
30          }
```

Member 3: KRISH

- Overall Character Design and Game Design
- Implemented Inheritance with two similar obstacles
- Also contributed to finding sounds and special design aspects of our game
- Worked on some character classes

Member 3: KRISH

snippets from code

- bgMaterial.png
- emogworlpatty.png
- emogworlpatty_150x150.png
- ezgif.com-gif-maker (1).gif
- fully yassified spongy.png
- fully_yassified_spongy_2_20
- jelly-removebg-preview_1_12

- Mario-jump-sound.wav
- Pacman-death-sound_(1).wav

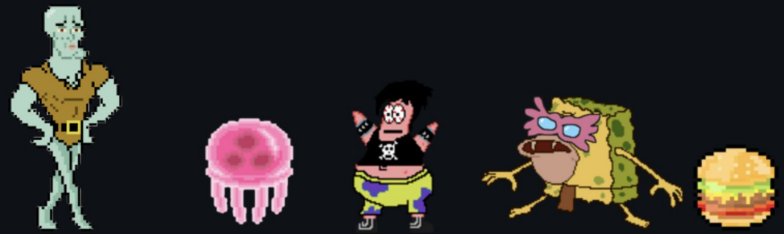
```
/* update the picture variable location */
private void update() {
        y1+= bvy1;

        if (y1 > 600) {
                y1 = 0;
        }

        tx.setToTranslation(x1, y1);
       tx.scale(.2, .2);
}

private void init(double a, double b) {
       tx.setToTranslation(a, b);
       tx.scale(0.4, 0.4);
}
```

Meet the Characters:



```
8   public class Spongebob extends Patrick {
9           //add location attributes
10                  private int x2 , y2;
11                  //position
12                  private Image img;
13                  private AffineTransform tx;
14
15                  public Spongebob (int x, int y) {
16                          super(x, y);
17                          img = getImage("/imgs/fully yassified spongy.png"); //load the image
18                          //use your variables
19                  }
```

Most Challenging Aspects

01. 2D Array Implementation

When trying to implement a 2D array, we came across a lot of issues because we weren't using basic Java objects, rather other classes.

02. Inheritance

When we implemented inheritance, the character we painted that was inherited from another class kept causing unknown errors.

03. General Issues/Bugs

There were some discouraging moments when issues occured because it was hard to stay motivated

04. Showing/Hiding Elements

It was tricky to figure out how to show and hide elements without using obj.splice methods from other imported classes.

Most Rewarding Aspects

01. Game Design

It was rewarding to be able to have creative freedom over our game. When we saw our characters painted, it was nice to see our vision.

02. Github Commits

The green on the contribution chart is motivating because it makes us feel like real programmers ☺️

03. Runnability of Program

When the program runs as intended with no issues, it is motivating especially after being stuck on one component for a long time.

04. End Result

Creating a game from scratch utilizing a lot of different Java elements paid off when we saw the game in action.

THANK YOU FOR LISTENING!

VIDEO