**Flappy Spongebob**

Aastha and Woongbeen

June 20th, 2018

**Table of Contents:**

**Program Specifications:**

Initial Specifications…………………………………………………………………….pg 3

Release Schedule………………………………………………………………………pg 4-5

Work in Progress Report……………………………………………………………….pg 6-8

**Program Code:**

Scratch Background……………………………………………………………………pg 9

Scratch Jumping……………………………………………………………………….pg 10-11

Scratch JellyFish……………………………………………………………………….pg 12-13

Release 1.0……………………………………………………………………………..pg 14

Release 2.0……………………………………………………………………………..pg 15-16

Release 2.1……………………………………………………………………………..pg 17

Release 2.2……………………………………………………………………………..pg 18-19

Release 2.3……………………………………………………………………………..pg 20

Main Game……………………………………………………………………………..pg 21-24

**Program Overview and Introduction**

Overview and Intro……………………………………………………………………..pg 25

**Disclaimer of Bugs**

Disclaimer of bugs……………………………………………………………………...pg25

**User Manual**

Manual…………………………………………………………………………………..pg 26-31

**Journal**

Journal………………………………………………………………………………….pg 32

**Sources**

Sources………………………………………………………………………………...pg 33

**Note to Future Programmers:**

Note to future programmers………………………………………………………….pg 34

**Extra Lessons that helped**

Favourite Lesson……………………………………………………………………...pg 35

**Program Overview:**

The program will emulate flappy Spongebob. The score will increase by 10 every second that Spongebob is still alive. The goal of this game is to achieve a higher score than the previous tries.

**Description of User Input:**

If the user chooses single - player game, the user will use the spacebar to move the Spongebob up. Every time the spacebar is hit the Spongebob moves up, and if the spacebar is not hit the Spongebob will start falling to the ground. If the uses chooses two- player, one of the players will control a Spongebob using the spacebar key and the other player will control the Patrick with the key W.

**Description of Program Output:**

The program will provide a choice between a single - player or a two - player. When the Spongebob touches the jellyfish, the Spongebob will fall to the ground and the program will stop recording the score. The program will show the highest score by the user and the score from the last game.

**Release Schedule**

|  |  |
| --- | --- |
| 1.0 | The background - blue (under water for Spongebob) |
| 2.0 | Creation of Spongebob - Rectangle shape |
| 2.1 | Spongebob moves up when spacebar pressed |
| 2.2 | Spongebob dropping to the ground when the spacebar not pressed |
| 2.3 | Adjusted speed of drop |
| 3.0 | Addition of the jellyfish tubes (rectangle boxes with the jellyfish image) |
| 3.1 | Stopping the game when Spongebob touches the jellyfish |
| 4.0 | Addition of score bar with current score |
| 4.1 | Making the score increase by 10 per second |
| 4.2 | Saving the previous game score |
| 4.3 | Saving the score by comparing scores between the score from the previous game and the highest score |
| 5.0 | Addition of choice between single - player or two - player |
| 5.1 | Creation of Patrick - Rectangle shape |
| 5.2 | Patrick moves up when the key W is pressed |
| 5.3 | Patrick dropping to the ground when the spacebar not pressed |
| 5.4 | Adjusted speed of drop |
| 6.0 | Stopping the game when Patrick touches the jellyfish |
| 7.0 | Addition of score bar with current score for Patrick |
| 7.1 | Saving Patrick’s previous game score and Spongebob’s score |
| 7.2 | Print out the winner of the game |

**Work In Progress Report**

·          Setting up scratches and release

·         Creating our first release

·         Learned how to create the background ( Scratches – background)

·         First Release of the background(Version1.0)

Code:

TextureRegion trBackground;

trBackground = new TextureRegion(new Texture("background.png"), 0, 0, 700, 525);

}

batch.draw(trBackground, 0, 0);

batch.draw(trBackground, 0, Gdx.*graphics*.getHeight());

·         Learned how to create spongebob (Piskel)

·         Getting the image of bad logic to go up when the spacebar is pressed and down when not pressed (Scratches – Jumping)

Code:

float fGrav, fVelo, fX, fY;

fX = Gdx.*graphics*.getWidth() / 4 - sprite.getWidth() / 2;

fY = Gdx.*graphics*.getHeight() / 2 - sprite.getHeight();

sprite.setPosition(fX, fY);

sprite.setScale(0.5f);

fGrav = -0.09f;

fVelo = 1f;

sprite.setY(fY);

fVelo += fGrav;

fY += fVelo;

if (Gdx.*input*.isKeyJustPressed(Input.Keys.*SPACE*)) {

      fVelo += 6;

  }

  System.*out*.println(fVelo);

  fVelo = MathUtils.*clamp*(fVelo, -5, 4);

}

Released 2.1 - Spongebob moves up when spacebar is pressed:

Code:

TextureRegion trBackground;

float fX, fY, fVelo;

trBackground = new TextureRegion(new Texture("background.png"), 0, 0, 700, 525);

  img = new Texture("Spongebob.png");

  sprite = new Sprite(img);

  fX = Gdx.*graphics*.getWidth() / 4 - sprite.getWidth() / 2;

  fY = Gdx.*graphics*.getHeight() / 2 - sprite.getHeight();

  sprite.setPosition(fX, fY);

  sprite.setScale(0.7f);

}

batch.draw(trBackground, 0, 0);

sprite.setY(fY);

fY += fVelo;

if (Gdx.*input*.isKeyJustPressed(Input.Keys.*SPACE*)) {

  fVelo += 6;

}

**Major Challenges/setbacks( reference specific code please):**

·         Getting the jumping to work - We were struggling to getting the jumping to work,

we tested

Using ‘sprite.translateY();’ did not work

·         Getting the background (Scratches – Background)

We tried using ‘.jpeg’ but that did not work, so we used

* Trying to get jellyfish to move side to side (Scratches-JellyFish)

  We tried using the java.awt, but it kept crashing, we eventually figured it out.

**Any modifications to your specifications/release schedule:**

-          Deleted release 1.1

**Description of your scratch/test program:**

**Describe the generic concept you needed to test out:**

The first scratch we had was called background and we used the starting code that LIBGDX gave us, to try and change the colours of the background and figure out how we were gonna import an image for the background into the scratch.

The second scratch we did was called Jumping. We needed to test out how to get the image to move up when the spacebar is pressed and how to get the image to move down when the space bar is not pressed.

The third scratch we did was called JellyFish and we used it to test out how to get the JellyFish

Source any web site/book that helped you with that concept:

The tutorial for the LIBGDX helped us for the background - <https://www.youtube.com/watch?v=IYjnOuxjdKQ&list=PLS9MbmO_ssyCZ9Tjfay2tOQoaOVoG59Iy&index=5>

<https://www.youtube.com/watch?v=cXgA1d_E-jY>  - helped us get started on the jumping

4. <https://www.youtube.com/watch?v=UyNm3n1WNAA> - helped to get the jellyfish moving a bit

**Describe the code and the lesson that you learned from it:**

We learned how to import images into our code, and set it according to the size of the screen.

We learned how to get the spacebar to control the image. We also learned how to get the image to go up, when the spacebar is pressed, and also how to get the image to speed up and come down when the spacebar is not pressed.

We learned how to move the image from one side to another, without using the arrow keys

**Describe any challenges that you enjoyed in integrating this scratch code into your major project:**

 Getting the jumping to work - We were struggling to getting the jumping to work,

we tested

Using ‘sprite.translateY();’ did not work

·         Getting the background (Scratches – Background)

We tried using ‘.jpeg’ but that did not work, so we used

The jellyFish scratch to work, we were unsuccessful, on getting the code to  work. We tried a lot of things, but we couldn’t get it to move from one side to the other.

**Scratch Background:**

package gdx.game;

import com.badlogic.gdx.ApplicationAdapter;

import com.badlogic.gdx.Gdx;

import com.badlogic.gdx.graphics.GL20;

import com.badlogic.gdx.graphics.Texture;

import com.badlogic.gdx.graphics.g2d.SpriteBatch;

import com.badlogic.gdx.graphics.g2d.Sprite;

import com.badlogic.gdx.graphics.g2d.TextureRegion;

public class Main extends ApplicationAdapter {

   SpriteBatch batch;

   TextureRegion trBackground;

   @Override

   public void create() {

       batch = new SpriteBatch();

       trBackground = new TextureRegion(new Texture("background.png"), 0, 0, 700, 525);

   }

   @Override

   public void render() {

       Gdx.gl.glClearColor(45 / 255f, 233 / 255f, 255 / 255f, 1); //colour

       Gdx.gl.glClear(GL20.GL\_COLOR\_BUFFER\_BIT);

       batch.begin();

       batch.draw(trBackground, 0, 0);

       batch.draw(trBackground, 0, Gdx.graphics.getHeight());

       batch.end();

   }

   @Override

   public void dispose() {

       batch.dispose();

   }

}

In this scratch we tested the background. Our goal for the scratch was to put in an image in the background and use that as our background for the game. Instead of drawing an image in pixel, we just took a picture from google. We were able to successfully import an image to use for our background.

**Scratch Jumping:**

package gdx.game;

import com.badlogic.gdx.ApplicationAdapter;

import com.badlogic.gdx.Gdx;

import com.badlogic.gdx.Input;

import com.badlogic.gdx.graphics.GL20;

import com.badlogic.gdx.graphics.Texture;

import com.badlogic.gdx.graphics.g2d.Sprite;

import com.badlogic.gdx.graphics.g2d.SpriteBatch;

import com.badlogic.gdx.math.MathUtils;

public class Main extends ApplicationAdapter {

   SpriteBatch batch;

   Texture img;

   Sprite sprite;

   float fGrav, fVelo, fX, fY;

   @Override

   public void create() {

       batch = new SpriteBatch();

       img = new Texture("Spongebob.png");

       sprite = new Sprite(img);

       fX = Gdx.graphics.getWidth() / 4 - sprite.getWidth() / 2;

       fY = Gdx.graphics.getHeight() / 2 - sprite.getHeight();

       sprite.setPosition(fX, fY);

       sprite.setScale(0.5f);

       fGrav = -0.09f;

       fVelo = 1f;

   }

   @Override

   public void render() {

       Gdx.gl.glClearColor(1, 0, 0, 1);

       Gdx.gl.glClear(GL20.GL\_COLOR\_BUFFER\_BIT);

       sprite.setY(fY);

       fVelo += fGrav;

       fY += fVelo;

       batch.begin();

       sprite.draw(batch);

       batch.end();

       if (Gdx.input.isKeyJustPressed(Input.Keys.SPACE)) {

           fVelo += 6;

       }

       System.out.println(fVelo);

       fVelo = MathUtils.clamp(fVelo, -5, 4);

   }

   @Override

   public void dispose() {

       batch.dispose();

       img.dispose();

   }

}

In this scratch we tested out getting spongebob to jump. We imported an image of spongebob that we drew from pixel, we used the if structure where if the spacebar is pressed then spongebob goes up, but when it is not pressed it drops down. We were successfully able to do that, and the only thing left was for us to adjust the drop of speed.

**Scratch JellyFish**

package gdx.game;

import com.badlogic.gdx.ApplicationAdapter;

import com.badlogic.gdx.Gdx;

import com.badlogic.gdx.Input;

import com.badlogic.gdx.graphics.GL20;

import com.badlogic.gdx.graphics.Texture;

import com.badlogic.gdx.graphics.g2d.Sprite;

import com.badlogic.gdx.graphics.g2d.SpriteBatch;

import com.badlogic.gdx.math.MathUtils;

import javafx.animation.KeyFrame;

import javafx.animation.KeyValue;

import javafx.animation.Timeline;

import javafx.application.Application;

import javafx.geometry.VPos;

import javafx.scene.Scene;

import javafx.scene.layout.Pane;

import javafx.scene.text.Font;

import javafx.scene.text.Text;

import javafx.stage.Stage;

import javafx.util.Duration;

public class Main extends Application {

   public static void main(String[] args) {

       Application.launch(args);

   }

   @Override

   public void start(Stage stage) {

       Text msg = new Text("java");

       msg.setTextOrigin(VPos.TOP);

       msg.setFont(Font.font(24));

       Pane root = new Pane(msg);

       root.setPrefSize(500, 70);

       Scene scene = new Scene(root);

       stage.setScene(scene);

       stage.setTitle("Scrolling Text");

       stage.show();

       double sceneWidth = scene.getWidth();

       double msgWidth = msg.getLayoutBounds().getWidth();

       KeyValue initKeyValue = new KeyValue(msg.translateXProperty(), sceneWidth);

       KeyFrame initFrame = new KeyFrame(Duration.ZERO, initKeyValue);

       KeyValue endKeyValue = new KeyValue(msg.translateXProperty(), -1.0

               \* msgWidth);

       KeyFrame endFrame = new KeyFrame(Duration.seconds(3), endKeyValue);

       Timeline timeline = new Timeline(initFrame, endFrame);

       timeline.setCycleCount(Timeline.INDEFINITE);

       timeline.play();

   }

}

In this scratch we tested out getting the image of jellyfish that we drew on pixel, imported in the scratch and trying to get it to move from one side to the other without pressing any arrow keys.  We were unsuccessful, on getting the code to work. We tried a lot of things, but we couldn’t get it to move from one side to the other.

**Release 1.0**

package gdx.game;

import com.badlogic.gdx.ApplicationAdapter;

import com.badlogic.gdx.Gdx;

import com.badlogic.gdx.graphics.GL20;

import com.badlogic.gdx.graphics.Texture;

import com.badlogic.gdx.graphics.g2d.SpriteBatch;

import com.badlogic.gdx.graphics.g2d.Sprite;

import com.badlogic.gdx.graphics.g2d.TextureRegion;

public class Main extends ApplicationAdapter {

   SpriteBatch batch;

   TextureRegion trBackground;

   @Override

   public void create() {

       batch = new SpriteBatch();

       trBackground = new TextureRegion(new Texture("background.png"), 0, 0, 700, 525);

   }

   @Override

   public void render() {

       Gdx.gl.glClearColor(45 / 255f, 233 / 255f, 255 / 255f, 1); //colour

       Gdx.gl.glClear(GL20.GL\_COLOR\_BUFFER\_BIT);

       batch.begin();

       batch.draw(trBackground, 0, 0);

       batch.draw(trBackground, 0, Gdx.graphics.getHeight());

       batch.end();

   }

   @Override

   public void dispose() {

       batch.dispose();

   }

}

**Release 2.0**

package gdx.game;

import com.badlogic.gdx.ApplicationAdapter;

import com.badlogic.gdx.Gdx;

import com.badlogic.gdx.graphics.GL20;

import com.badlogic.gdx.graphics.Texture;

import com.badlogic.gdx.graphics.g2d.SpriteBatch;

import com.badlogic.gdx.graphics.g2d.Sprite;

import com.badlogic.gdx.graphics.g2d.TextureRegion;

public class Main extends ApplicationAdapter {

   SpriteBatch batch;

   Texture img;

   Sprite sprite;

   TextureRegion trBackground;

   @Override

   public void create() {

       batch = new SpriteBatch();

       trBackground = new TextureRegion(new Texture("background.png"), 0, 0, 700, 525);

       img = new Texture("Spongebob.png");

       sprite = new Sprite(img);

       sprite.setPosition(

               Gdx.graphics.getWidth() / 4 - sprite.getWidth() / 2,

               Gdx.graphics.getHeight() / 2 - sprite.getHeight());

   }

   @Override

   public void render() {

       Gdx.gl.glClearColor(45 / 255f, 233 / 255f, 255 / 255f, 1); //colour

       Gdx.gl.glClear(GL20.GL\_COLOR\_BUFFER\_BIT);

       batch.begin();

       batch.draw(trBackground, 0, 0);

       batch.draw(trBackground, 0, Gdx.graphics.getHeight());

       sprite.draw(batch);

       batch.end();

   }

   @Override

   public void dispose() {

       batch.dispose();

   }

}

**Release 2.1**

package gdx.game;

import com.badlogic.gdx.ApplicationAdapter;

import com.badlogic.gdx.Gdx;

import com.badlogic.gdx.Input;

import com.badlogic.gdx.graphics.GL20;

import com.badlogic.gdx.graphics.Texture;

import com.badlogic.gdx.graphics.g2d.SpriteBatch;

import com.badlogic.gdx.graphics.g2d.Sprite;

import com.badlogic.gdx.graphics.g2d.TextureRegion;

public class Main extends ApplicationAdapter {

   SpriteBatch batch;

   Texture img;

   Sprite sprite;

   TextureRegion trBackground;

   float fX, fY, fVelo;

   @Override

   public void create() {

       batch = new SpriteBatch();

       trBackground = new TextureRegion(new Texture("background.png"), 0, 0, 700, 525);

       img = new Texture("Spongebob.png");

       sprite = new Sprite(img);

       fX = Gdx.graphics.getWidth() / 4 - sprite.getWidth() / 2;

       fY = Gdx.graphics.getHeight() / 2 - sprite.getHeight();

       sprite.setPosition(fX, fY);

       sprite.setScale(0.7f);

   }

   @Override

   public void render() {

       Gdx.gl.glClearColor(45 / 255f, 233 / 255f, 255 / 255f, 1); //colour

       Gdx.gl.glClear(GL20.GL\_COLOR\_BUFFER\_BIT);

       batch.begin();

       batch.draw(trBackground, 0, 0);

       sprite.setY(fY);

       fY += fVelo;

       if (Gdx.input.isKeyJustPressed(Input.Keys.SPACE)) {

           fVelo += 6;

       }

       sprite.draw(batch);

       batch.end();

   }

   @Override

   public void dispose() {

       batch.dispose();

   }

}

**Release 2.2**

package gdx.game;

import com.badlogic.gdx.ApplicationAdapter;

import com.badlogic.gdx.Gdx;

import com.badlogic.gdx.Input;

import com.badlogic.gdx.graphics.GL20;

import com.badlogic.gdx.graphics.Texture;

import com.badlogic.gdx.graphics.g2d.SpriteBatch;

import com.badlogic.gdx.graphics.g2d.Sprite;

import com.badlogic.gdx.graphics.g2d.TextureRegion;

import com.badlogic.gdx.math.MathUtils;

public class Main extends ApplicationAdapter {

   SpriteBatch batch;

   Texture img;

   Sprite sprite;

   TextureRegion trBackground;

   float fX, fY, fVelo, fGrav;

   @Override

   public void create() {

       batch = new SpriteBatch();

       trBackground = new TextureRegion(new Texture("background.png"), 0, 0, 700, 525);

       img = new Texture("Spongebob.png");

       sprite = new Sprite(img);

       fX = Gdx.graphics.getWidth() / 4 - sprite.getWidth() / 2;

       fY = Gdx.graphics.getHeight() / 2 - sprite.getHeight();

       sprite.setPosition(fX, fY);

       sprite.setScale(0.7f);

       fGrav = -0.09f;

       fVelo = 1f;

   }

   @Override

   public void render() {

       Gdx.gl.glClearColor(45 / 255f, 233 / 255f, 255 / 255f, 1); //colour

       Gdx.gl.glClear(GL20.GL\_COLOR\_BUFFER\_BIT);

       batch.begin();

       batch.draw(trBackground, 0, 0);

       sprite.draw(batch);

       sprite.setY(fY);

       batch.end();

       fVelo += fGrav;

       fY += fVelo;

       if (Gdx.input.isKeyJustPressed(Input.Keys.SPACE)) {

           fVelo += 6;

       }

       fVelo = MathUtils.clamp(fVelo, -5, 4);

   }

   @Override

   public void dispose() {

       batch.dispose();

   }

}

**Release 2.3**

package gdx.game;

import com.badlogic.gdx.Game;

import com.badlogic.gdx.graphics.g2d.SpriteBatch;

public class GamMain extends Game {

   SpriteBatch batch;

   ScrPlay scrPlay;

   ScrMenu scrMenu;

   int nScreen;

   public void updateScreen(int \_nScreen) {

       nScreen = \_nScreen;

       if (nScreen == 0) setScreen(scrMenu);

       else if (nScreen == 1) setScreen(scrPlay);

   }

   @Override

   public void create() {

       nScreen = 0;

       batch = new SpriteBatch();

       scrMenu = new ScrMenu(this);

       scrPlay = new ScrPlay(this);

       updateScreen(0);

   }

   @Override

   public void render() {

       super.render();

   }

   @Override

   public void dispose() {

       super.dispose();

}

**Main Game**

package gdx.game;

import com.badlogic.gdx.Gdx;

import com.badlogic.gdx.Input;

import com.badlogic.gdx.InputProcessor;

import com.badlogic.gdx.Screen;

import com.badlogic.gdx.graphics.GL20;

import com.badlogic.gdx.graphics.Texture;

import com.badlogic.gdx.graphics.g2d.Sprite;

import com.badlogic.gdx.graphics.g2d.SpriteBatch;

import com.badlogic.gdx.graphics.g2d.TextureRegion;

import com.badlogic.gdx.math.MathUtils;

public class ScrPlay implements Screen, InputProcessor {

   GamMain game;

   SpriteBatch batch;

   Texture txSB, txInstr;

   Sprite sprSB, sprInstr;

   TextureRegion trBackground;

   float fX, fY, fVelo, fGrav;

   boolean bStart;

   public ScrPlay(GamMain game) {

       this.game = game;

       this.batch = game.batch;

       trBackground = new TextureRegion(new Texture("background.png"), 0, 0, 700, 525);

       txSB = new Texture("Spongebob.png");

       txInstr = new Texture("Instruction.png");

       sprSB = new Sprite(txSB);

       sprInstr = new Sprite(txInstr);

       fX = Gdx.graphics.getWidth() / 4 - sprSB.getWidth() / 2;

       fY = Gdx.graphics.getHeight() / 2 - sprSB.getHeight();

       bStart = false;

       sprSB.setPosition(fX, fY);

       sprSB.setScale(0.7f);

       sprInstr.setScale(0.5f);

       sprInstr.setPosition(Gdx.graphics.getWidth() / 2 - sprInstr.getWidth() / 2,

               Gdx.graphics.getHeight() / 2 - sprInstr.getHeight() / 2);

       fGrav = -0.09f;

       fVelo = 0;

       Gdx.input.setInputProcessor(this);

   }

   @Override

   public void show() {

   }

   @Override

   public void render(float delta) {

       if (!bStart) {

           batch.begin();

           batch.draw(trBackground, 0, 0);

           sprInstr.draw(batch);

           batch.end();

           if (Gdx.input.isKeyJustPressed(Input.Keys.SPACE)) {

               bStart = true;

               fVelo = 1;

           }

       }

       if (bStart) {

           Gdx.gl.glClearColor(45 / 255f, 233 / 255f, 255 / 255f, 1); //colour

           Gdx.gl.glClear(GL20.GL\_COLOR\_BUFFER\_BIT);

           batch.begin();

           batch.draw(trBackground, 0, 0);

           sprSB.draw(batch);

           batch.end();

           sprSB.setY(fY);

           fVelo += fGrav;

           fY += fVelo;

           if (Gdx.input.isKeyJustPressed(Input.Keys.SPACE)) {

               fVelo += 7;

           }

           fVelo = MathUtils.clamp(fVelo, -7, 4);

       }

   }

   @Override

   public void resize(int width, int height) {

   }

   @Override

   public void pause() {

   }

   @Override

   public void resume() {

   }

   @Override

   public void hide() {

   }

   @Override

   public void dispose() {

       txSB.dispose();

       txInstr.dispose();

       batch.dispose();

   }

   @Override

   public boolean keyDown(int keycode) {

       return false;

   }

   @Override

   public boolean keyUp(int keycode) {

       return false;

   }

   @Override

   public boolean keyTyped(char character) {

       return false;

   }

   @Override

   public boolean touchDown(int screenX, int screenY, int pointer, int button) {

       return false;

   }

   @Override

   public boolean touchUp(int screenX, int screenY, int pointer, int button) {

       return false;

   }

   @Override

   public boolean touchDragged(int screenX, int screenY, int pointer) {

       return false;

   }

   @Override

   public boolean mouseMoved(int screenX, int screenY) {

       return false;

   }

   @Override

   public boolean scrolled(int amount) {

       return false;

   }

}

**Program Overview:**

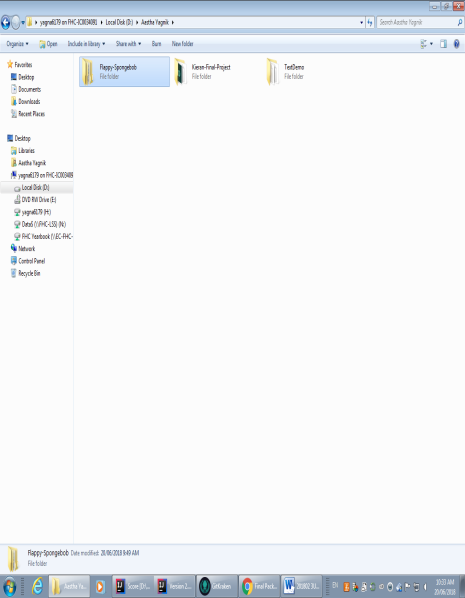
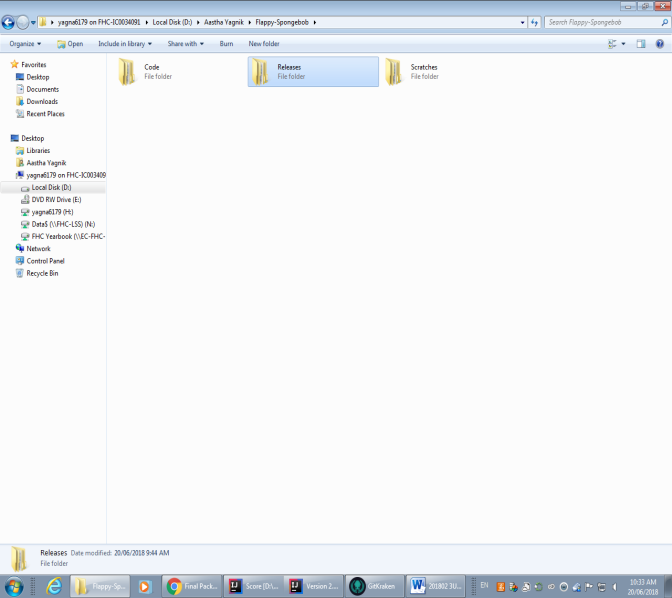
In this program we will emulate flappy Spongebob. The score will increase by 10 every second that Spongebob is still alive. The goal of this game is to achieve a higher score than the previous tries. The user will use the spacebar to move spongebob up and down, and try to avoid getting stung by the jellyfish.

**Disclaimer known of Bugs:**

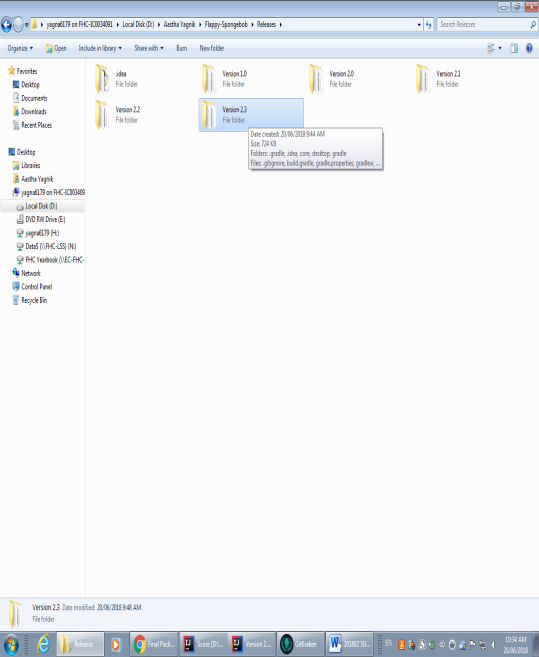
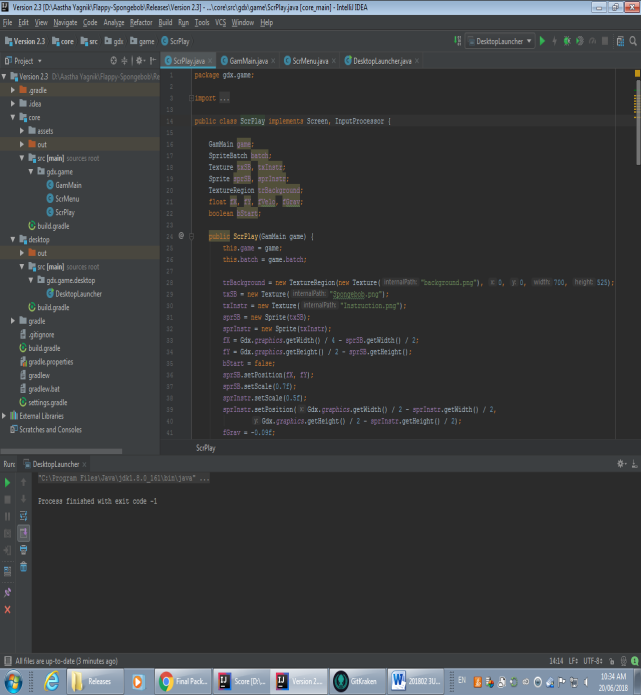
1. One of the problems we were having, was that when we were testing the jellyfish jumping scratch, I couldn't get the program to run. Everytime I try to run it it said “main” java.lang.NullPointerException error. The reason it didn’t work, was because we had to import a “import javafx.animation.KeyFrame;”, but once we did that, it didn’t work so we did it again.

**Manual**

1. Our game is in the folder Flappy Spongebob, then in folder release, then in folder release 2.3, and of the different screens it is the screen called SRCPLAY.JAVA.

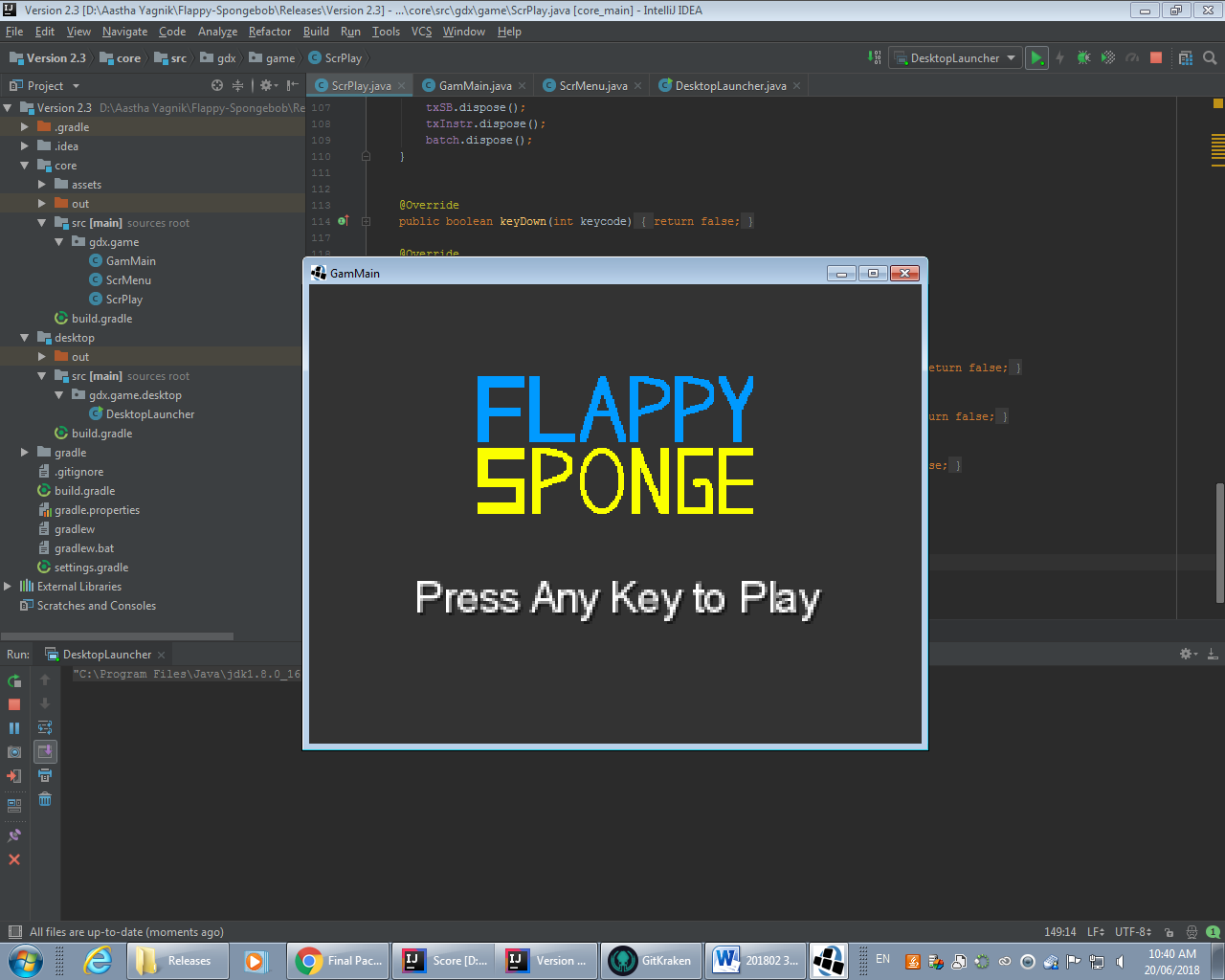
 

1. 2.

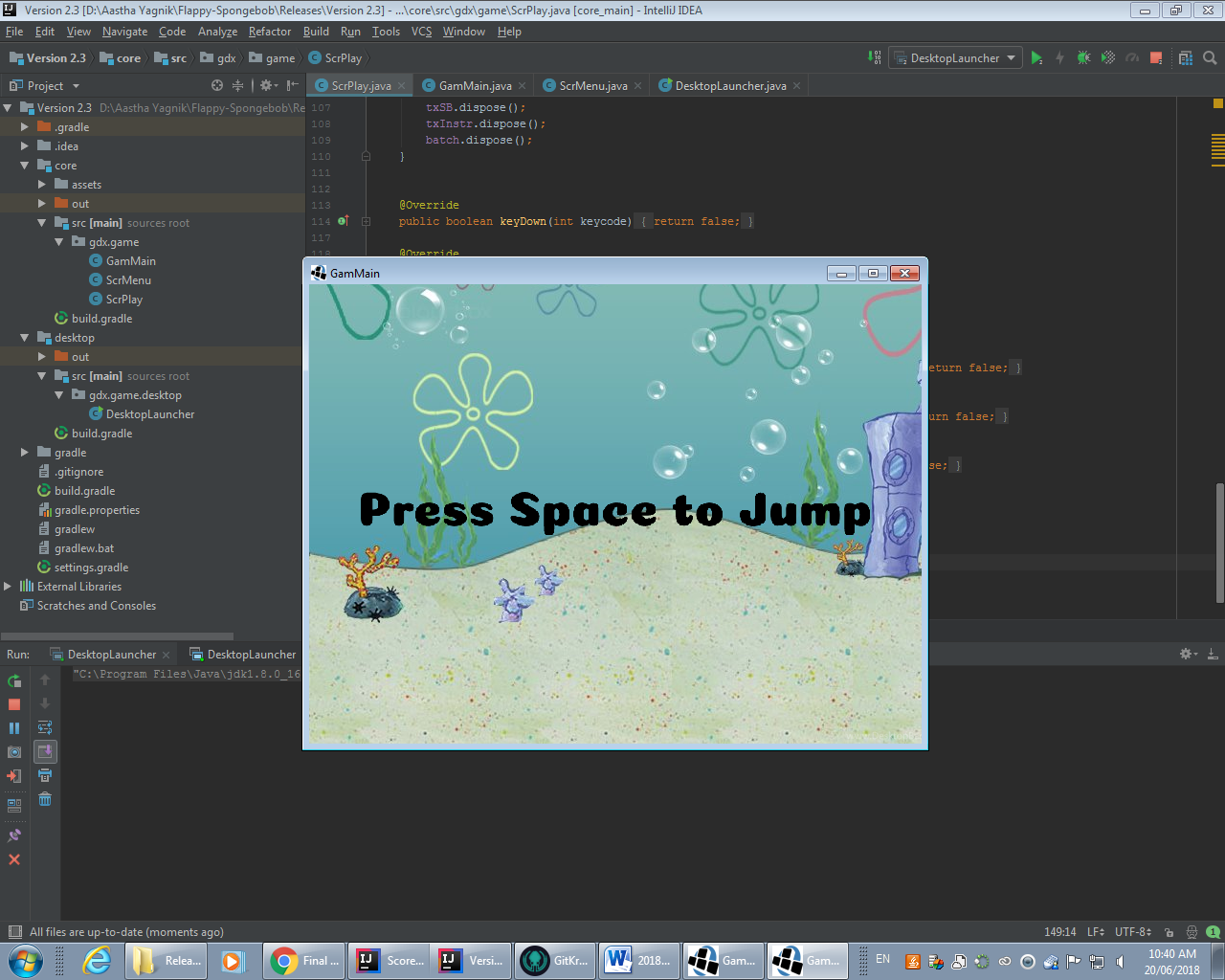
 

3.                                                               4.

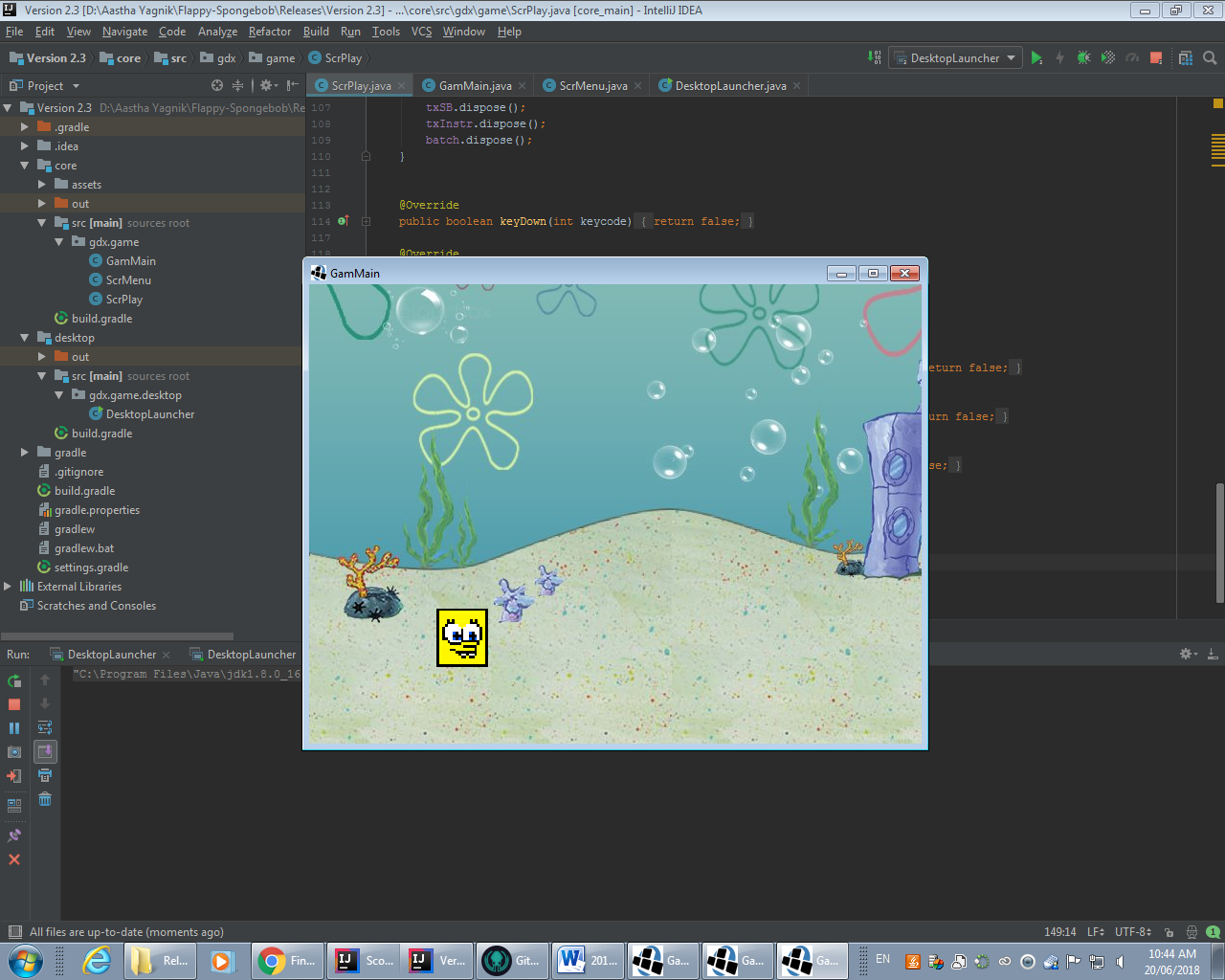
Next you run the program and it says press any key to play.



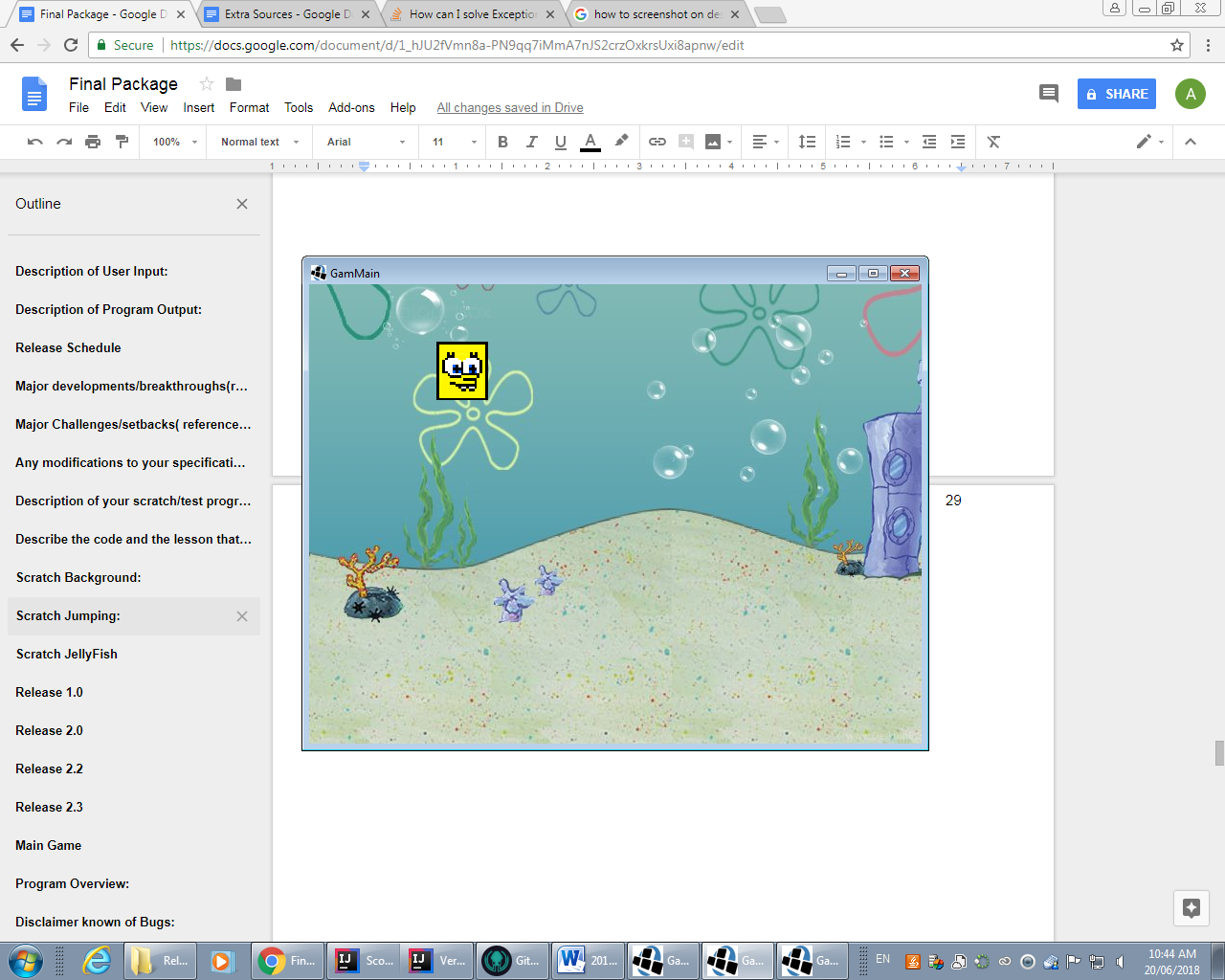
Next, when you press any key to play the background will come up, with the instructions that says press spacebar to jump.



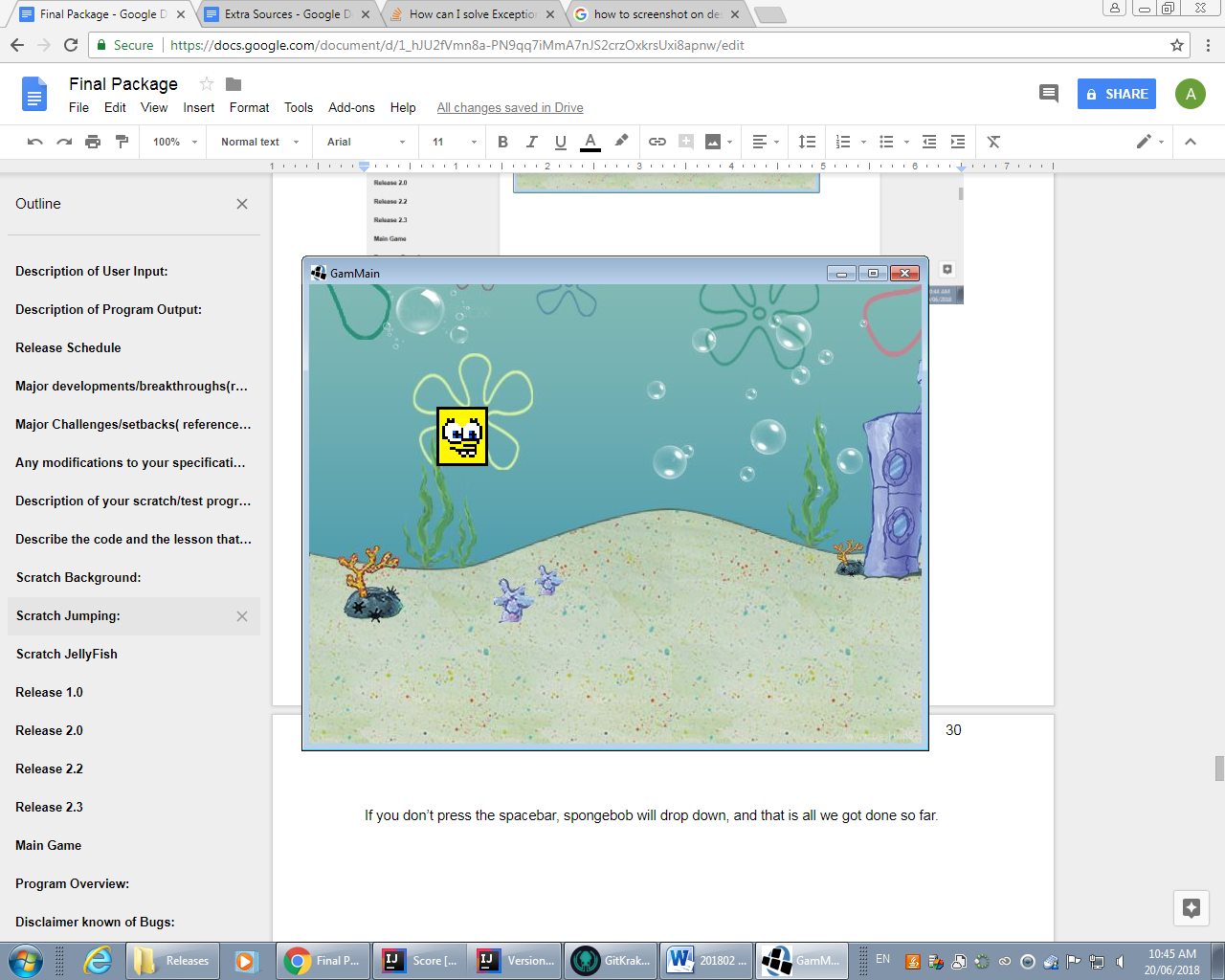
When you press the spacebar an image of spongebob comes on the screen with the background, and you use, the spacebar to control spongebob.



If you press the spacebar, the image will move up



If you don’t press the spacebar, spongebob will drop down, and that is all we got done so far.



**Journal**

Day 1 - We created a folder called Floppy Spongebob, and worked on creating scratches and releases. We created different scratches one called background, one called jumping and one called creating spongebob. I started to work on creating the spongebob, and Woongbeen started to trying to get spongebob to jump.

Day 2 - Woongbeen started to work on trying to get spongebob to move at first. He watched a YouTube video called ‘Coding Challenge #31: Flappy Bird’, that helped him get started. I started to work on the background, and started watching the video “LibGDX 2D Tutorial #15: Scrolling Star Background” but then Woongbeen needed the spongebob image so instead I went on piskel to draw the spongebob image. I drew the image, but I drew it 19x17 pixels which was small, so I started to draw it again.

Day 3 -Woongbeen got the jumping code. We deleted the scratches that we didn’t work on, or that we did not need. For ex. we deleted the background file, and recreated it, and added a basic scratch. I went on piskel and drew spongebob again, the right size this time, and pushed it.

Day 4  - I worked on the Work In Progress report and Woongbeen released 1.0, and 2.0. 1.0 is the background, and 2.0 is the release of the spongebob image that I drew.

Day 7 - Woongbeen released version 2.1, and 2.2,

Day 9 - Started to work on the scratch for JellyFish moving side to side. Started following a tutorial for trying to get the jellyfish to get it to move from side to side:

https://www.youtube.com/watch?v=UyNm3n1WNAA

Woongbeen created a JellyFish from pixel, and pushed it and I pulled it.

Day 11 : Woongbeen started to work on setting up screens and I was still working on trying to get the jellyfish to move but the video I used, didn’t help to solve it. I could get it moving to one side, but then it would just crash after it moved to one side.

Day 12 :  Woongbeen pushed the code with screens setup and we still didn’t get the jellyfish working, so I started to work on the final package and Woongbeen started to work on the code for the jellyfish.

**Extra Sources**

1.    https://www.youtube.com/watch?v=UyNm3n1WNAA - helped get started on the background

2.    https://www.youtube.com/watch?v=cXgA1d\_E-jY – helped getting started on the jumping of Flappy Bird

3.    https://www.piskelapp.com/ -  created spongebob on this website

4. https://www.youtube.com/watch?v=UyNm3n1WNAA - helped a bit to getting the jellyfish move from one side to the other(helped with the basic concept)