Odd/Even/Compact* Semester (year)

BINUS UNIVERSITY
BINUS INTERNATIONAL

Assignment Cover Letter (Individual/Group* Work)

Student Information: **Surname Given Names Student ID Number**

1. Srinivasan Shravan Srinivasan 24400042872

Course Code : COMP6699

Course Name : Object Oriented Programming

Class : L2AC

Name of Lecturer(s) : 1. Mr Jude 2. Mr Raveltan

Major : Computer Science

Title of Assignment : Object Oriented Programming Project Report
(if any)

Type of Assignment : Making a Game using Java with IDE Greenfoot

Submission Pattern

Due Date : 22/7/2021

Submission Date : 7/7/2021

The assignment should meet the below requirements.
1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer's instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

Plagiarism/Cheating
BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

Declaration of Originality
By signing this assignment, I/we* understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I/we* declare that the work contained in this assignment is my/our* own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student: (Name of Student)

1.

etc. *) Delete the inappropriate option

“Golden Ju”

Object Oriented Programming

Project Report

Prepared by:

Shravan Srinivasan

NIM:24400042872

BINA NUSANTARA UNIVERSITY
2021

“Golden Ju”

Object Oriented Programming

Project Report

Shravan Srinivasan NIM:24400042872

Signature

BINA NUSANTARA UNIVERSITY

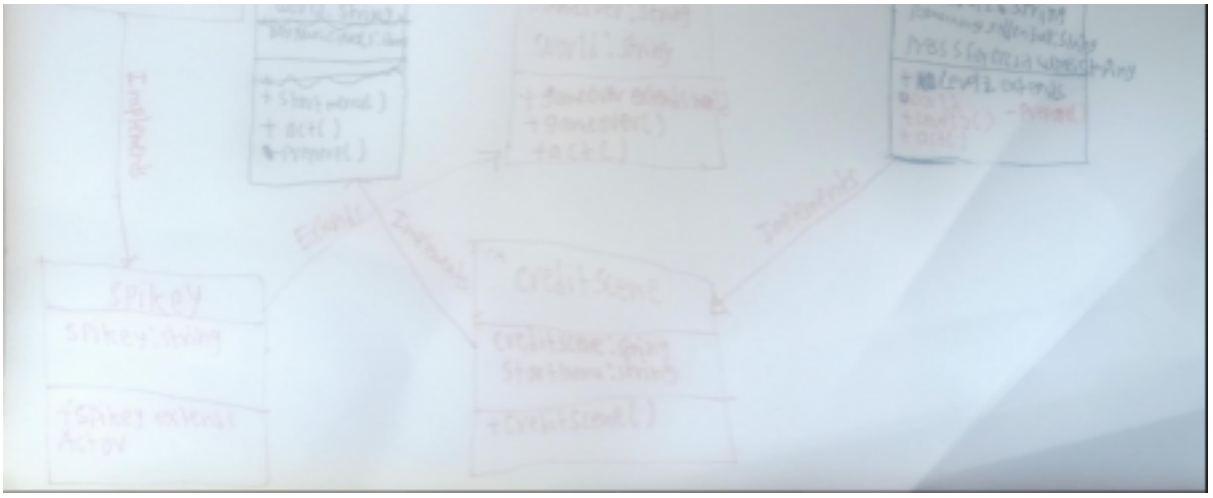
2021

Project Specification

The project is all about making a simple game called Golden Ju. The project was created by using an application called Green foot. Golden Ju is an educational game where you are portrayed as a business man that is trying to learn what it is like living in the business world. You have to do so by collecting the golden ball named Ju. To start the game you are taken to the start screen where it is blank and when you click play the text will automatically appear as known below the evidence and screenshots. You will start from the middle on the two levels. You have two levels to do which are MyWorld with score and level2 which is also known as real life without score. The first is given a score so that you are given a self-fulfillment task to do. While the second level is there to bring your overconfidence up by

stating collect the “remaining golden ball”. This is intended not because of making the fun part of the game disappear but it instead teaches you a lesson that when you become a businessman you are to be able to cope with such trick words. And also, when you are hit by Spikey it will take you to the game over scene and you have to press q.

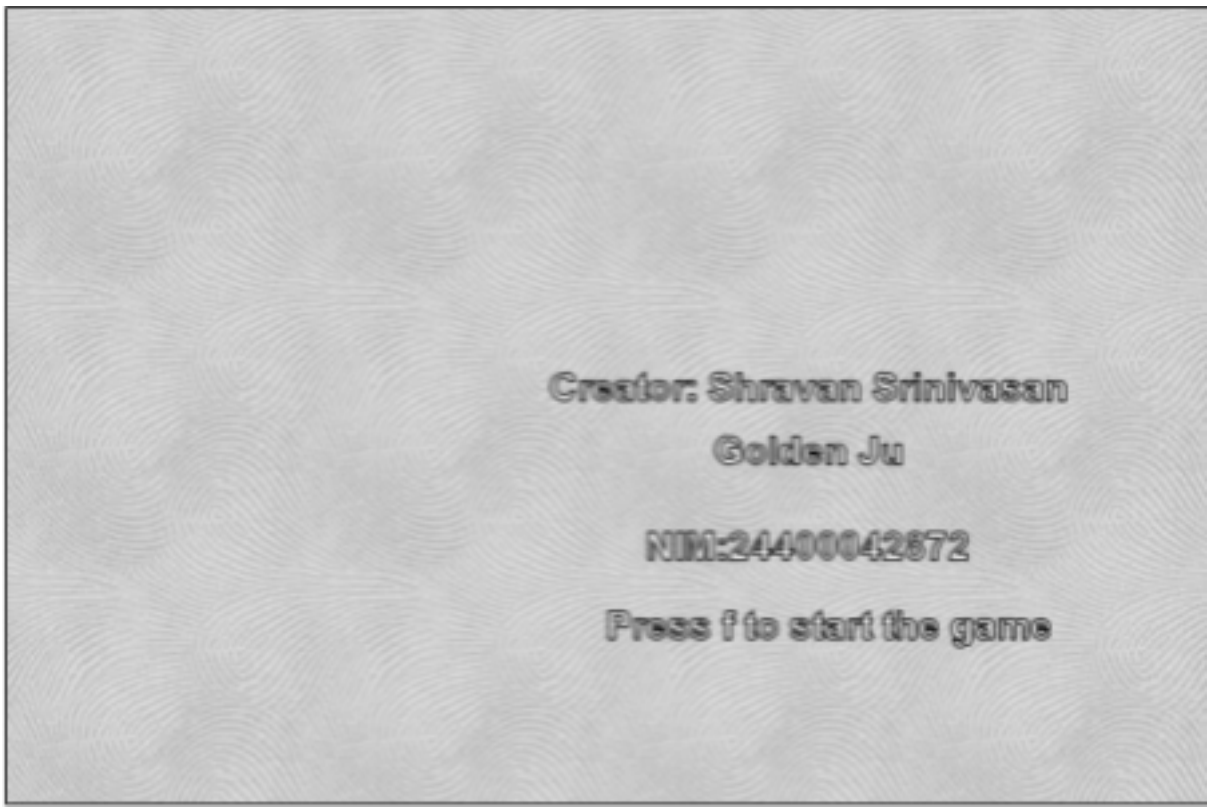
Solution Design



Algorithms, Solution Scheme, and Data Structure

There are two big classes which are World and Actor. They both are made for creating the whole package of the game. World class has five places which are Creditscene, MyWorld, gameover, level2, and startmenu. Whereas an Actor has four objects which are ball, platform, spikey, and sprite. The implementation of the solution scheme using Java programming language led to the simplification of the process and reliability of the results. Solution scheme focuses on the Java programming language and begins in the Greenfoot environment, then moves on to the Eclipse environment later on. Hence, by using all of the components, the game Golden Ju was made. Data structures that were directly used and implemented, like array, linked list, map, stack, and queue provided by Java API.

Start Menu Image

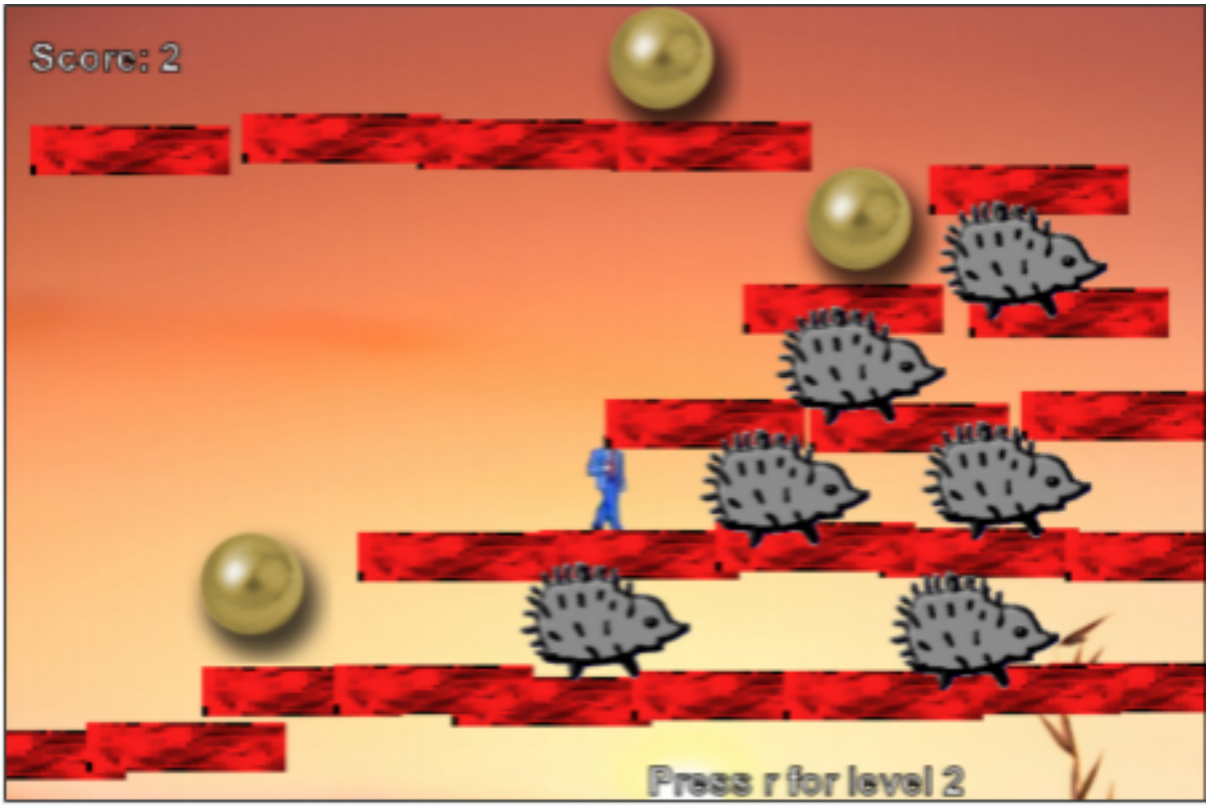


Start Menu Code

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

//uses of custom-built classes and methods along with polymorphism, inheritance, exception handling, and java collection

public class startmenu extends World

{

   GreenfootSound myMusic= new GreenfootSound("red.wav");//add musical background for all levels including start and credit scene.

  /**

   * Constructor for objects of class startmenu.

   *

   */

   public startmenu()

  {
      // Create a new world with 600x400 cells with a cell size of 1x1 pixels.

      super(600, 400, 1);
```

```
  }

   public void act() //make an act class when it runs the texts will show up //also add key bind
F to start.

  {

      showText("Press f to start the game",410, 310);

      showText("Golden Ju", 400,222);

      showText("Creator: Shravan Srinivasan", 400,190);

      showText("NIM:24400042872", 400,270);

       if(Greenfoot.isKeyDown("f"))
       Greenfoot.setWorld(new MyWorld());

       myMusic.play();//call out myMusic function

  }

   private void prepare() // make a class prepare as an object

  {
      sprite sprite= new sprite();// so a sprite can be used later on in the

      game. addObject(sprite, 600, 400);

      platform platform = new platform();// so a platform can be used later on in

      game. addObject(platform, 450, 400);}

}
```

MyWorld Image



MyWorld Code

```
import greenfoot.*; // (World, Actor, Images, Music and ScenarioInfo)//use methods of greenfoot.

//uses of custom-built classes and methods, along with exception handling, primitive data types, and java collections

//apply decency uses of class, polymorphism, methods, and

inheritance. public class MyWorld extends World

{ // use static link to set score to always be 0 which in other words is exception handling.

   public static int score= 0;

   public MyWorld() {
```

```
        // certain size and features on what my world will look like

        super(600, 400, 1);

        addObject(new ball(), 300, 150); // so that you can add balls

        addObject(new sprite(),300, 150); // so that you can add

        sprites

       prepare();// call out the prepare class using the method prepare();

    }

   public void act()

    { // use exception handling to add up the scores and
bind certain keys.

        showText("Score: " + score, 50, 25);

        showText("Press r for level 2", 400,390);

        if(Greenfoot.isKeyDown("q"))

        Greenfoot.setWorld(new startmenu());

        if(Greenfoot.isKeyDown("r"))
        Greenfoot.setWorld(new level2());



    }
   private void prepare()
```

```
    { // uses a variety of java collections, instance
variables, and objects.


        score=0;
        sprite sprite= new sprite();


        addObject(sprite, -50, -25);


       platform platform = new platform();


        addObject(platform, 450, 400);


        sprite.setLocation(-50,-25);


        sprite.setLocation(-50,-25);


       platform.setLocation(412,246);


       platform.setLocation(415,246);


       platform.setLocation(416,235);


       platform platform2 = new platform();


        addObject(platform2,485,248);
       platform2.setLocation(502,245);


       platform2.setLocation(503,246);


       platform platform3 = new platform();
        addObject(platform3,578,241);


       platform3.setLocation(591,250);
```

```
platform platform4 = new platform();

addObject(platform4,57,341);

platform4.setLocation(48,346);

platform platform5 = new platform();

addObject(platform5,148,344);

platform5.setLocation(148,345);

platform platform6 = new platform();

addObject(platform6,239,351);

platform6.setLocation(300,282);

platform4.setLocation(74,391);

platform platform7 = new platform();

addObject(platform7,382,280);

platform7.setLocation(404,285);

platform2.setLocation(516,289);

platform6.setLocation(317,276);

platform.setLocation(273,350);

platform platform8 = new platform();
```

```
addObject(platform8,323,354);

platform8.setLocation(349,344);

platform8.setLocation(347,345);

platform3.setLocation(412,342);

platform4.setLocation(50,390);

platform4.setLocation(10,377);

platform4.setLocation(10,377);

platform4.setLocation(10,377);

platform3.setLocation(469,339);

platform8.setLocation(573,343);

platform platform9 = new platform();

addObject(platform9,364,344);

platform9.setLocation(364,343);

platform9.setLocation(363,347);

platform platform10 = new

platform();

addObject(platform10,208,343);
```

```
platform10.setLocation(210,350);

platform7.setLocation(439,266);

platform2.setLocation(487,275);

platform platform11 = new platform();

addObject(platform11,580,277);

platform11.setLocation(580,277);

platform8.setLocation(553,353); ball

ball = new ball();



addObject(ball,497,144);

ball.setLocation(502,158);

removeObject(ball);
removeObject(sprite);

platform8.setLocation(508,344);

platform3.setLocation(572,343);

platform10.setLocation(214,344);

platform7.setLocation(391,270);

platform platform12 = new platform();
```

```
addObject(platform12,239,271);

platform12.setLocation(226,277);

platform12.setLocation(226,277);

platform platform13 = new platform();

addObject(platform13,520,197);

platform13.setLocation(558,206);

platform platform14 = new platform();

addObject(platform14,437,204);

platform14.setLocation(452,212);

platform platform15 = new platform();

addObject(platform15,360,207);

platform15.setLocation(350,210);

platform platform16 = new platform();

addObject(platform16,379,141);

platform16.setLocation(406,128);

platform platform17 = new platform();

addObject(platform17,494,142);
```

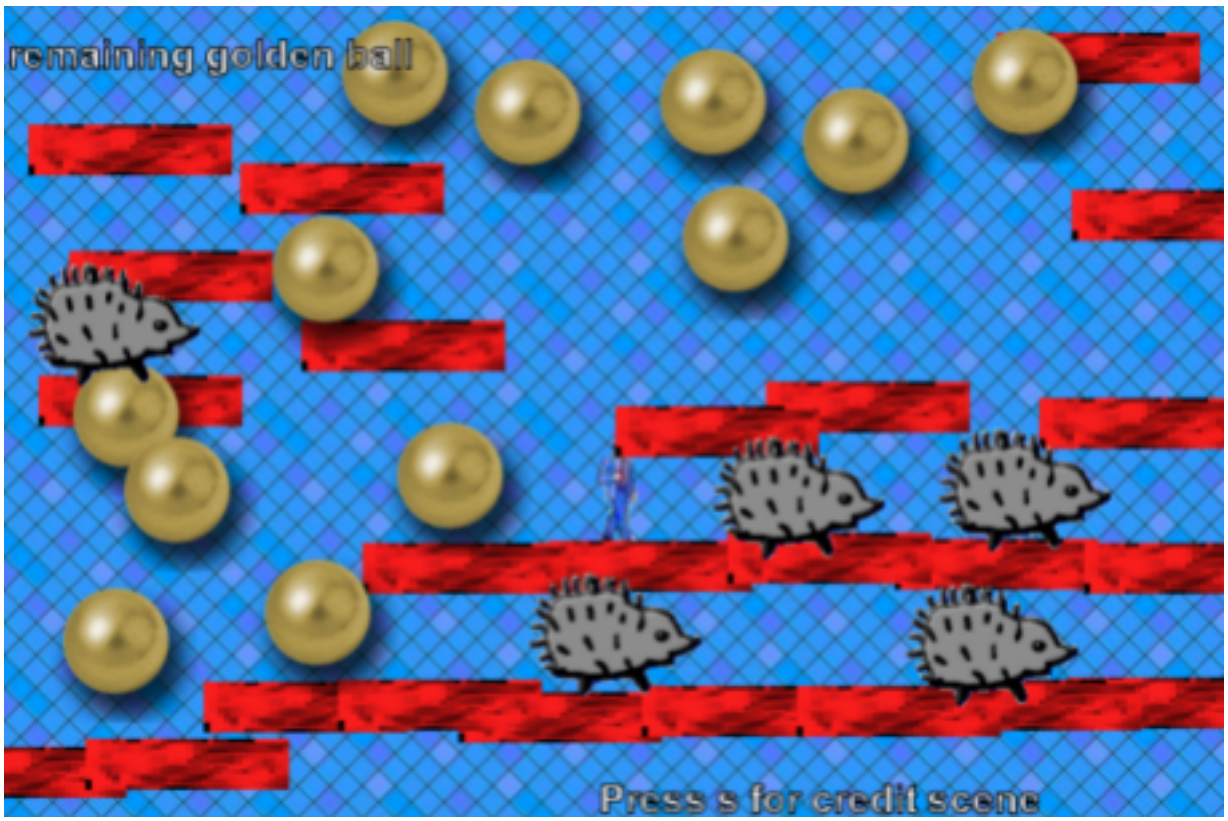
```
platform17.setLocation(507,134);

platform platform18 = new platform();

addObject(platform18,572,130);

platform18.setLocation(583,140);

platform18.setLocation(582,138);

platform platform19 = new platform();

addObject(platform19,415,348);

platform19.setLocation(439,347);

platform platform20 = new platform();

addObject(platform20,353,70);

platform platform21 = new platform();

addObject(platform21,265,72);

platform21.setLocation(256,69);

platform platform22 = new platform();

addObject(platform22,142,68);

platform22.setLocation(168,66);

platform platform23 = new platform();
```

```
addObject(platform23,41,68);

platform23.setLocation(62,72);

platform platform24 = new platform();

addObject(platform24,80,374);

platform24.setLocation(93,381);

platform24.setLocation(90,373);

platform7.setLocation(405,272);

platform16.setLocation(419,152);

platform17.setLocation(485,96);

platform18.setLocation(532,154);

platform17.setLocation(512,92); ball

ball2 = new ball();

addObject(ball2,231,207); ball

ball3 = new ball();

addObject(ball3,427,107); ball

ball4 = new ball();

addObject(ball4,328,26);
```

```
ball ball5 = new ball();

addObject(ball5,123,291);

spikey spikey = new

spikey();

addObject(spikey,218,306);

spikey spikey2 = new spikey();

addObject(spikey2,334,242);

spikey.setLocation(201,308);

spikey.setLocation(300,310);

spikey2.setLocation(390,243);

spikey spikey3 = new spikey();

addObject(spikey3,501,239);

spikey spikey4 = new spikey();

addObject(spikey4,485,315);

spikey spikey5 = new spikey();

addObject(spikey5,509,128);
```

```
        spikey spikey6 = new spikey();

        addObject(spikey6,399,185);

        spikey6.setLocation(429,186);

        spikey6.setLocation(391,176);

        spikey6.setLocation(391,176);

        spikey6.setLocation(391,176);

        spikey6.setLocation(379,167);

        spikey6.setLocation(400,172);

        removeObject(spikey6);

        addObject(spikey6,428,181);

    ball2.setLocation(243,244); }

}
```

Level2 Image



Level2 Code

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)


// Create class of level2
  //uses of custom-built classes and methods along with polymorphism, inheritance, exception handling, primitive data types, and java collections
public class level2 extends World
{



    public level2() {


        super(600, 400, 1);
        addObject(new ball(), 300, 150);
        addObject(new sprite(),300, 150);
         prepare();
    }
    public void act()
    { // display text for the scenario
       showText("Collect the remaining golden ball", 50, 25);
       showText("Press s for credit scene", 400,390);
        // When you have done collecting the golden ball click s for credit scene.
         if(Greenfoot.isKeyDown("s"))
```

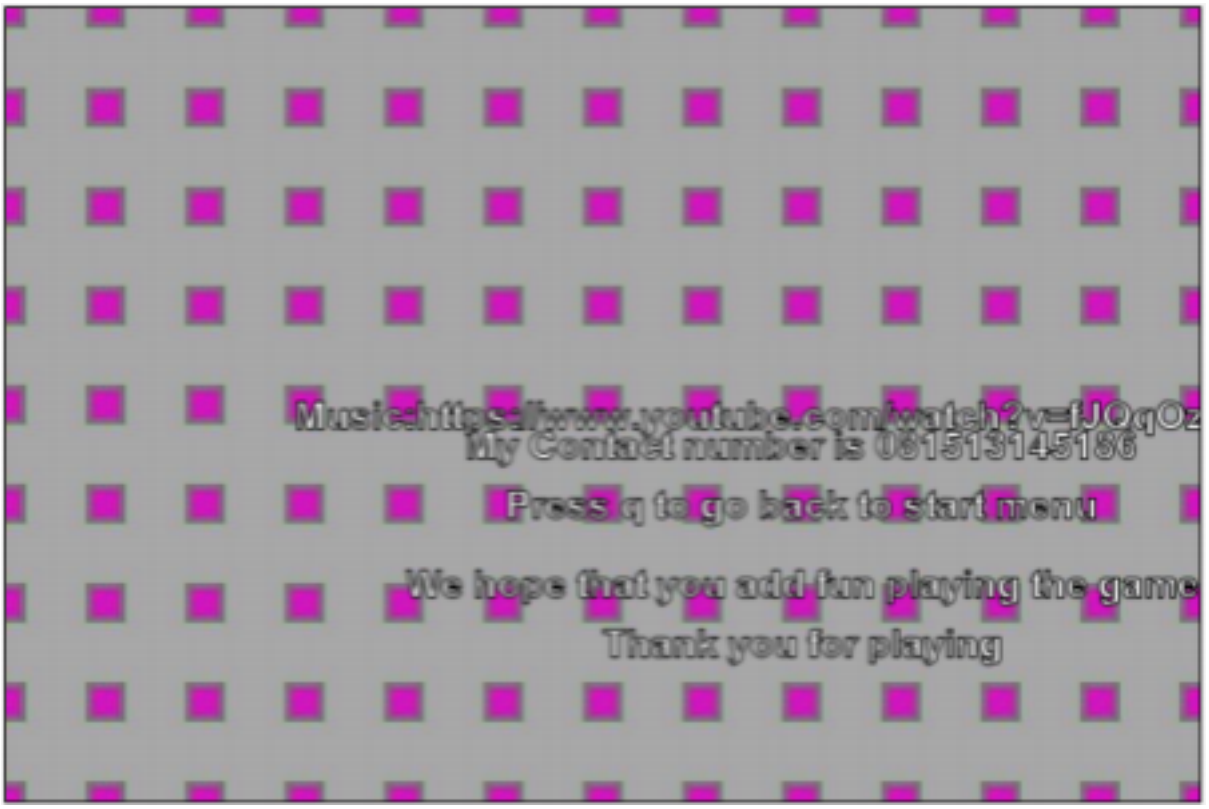
```
    Greenfoot.setWorld(new Creditscene());
}
// everything that is needed for the level2 background game
 private void prepare()
{ // set a sprite on a location and also the platform
    sprite sprite= new sprite();
    addObject(sprite, 600, 400);
     platform platform = new
     platform();
    addObject(platform, 450, 400);


    sprite.setLocation(600,400);
    sprite.setLocation(600,400);
    platform.setLocation(412,246);
    platform.setLocation(415,246);
    platform.setLocation(416,235);
    platform platform2 = new platform();
    addObject(platform2,485,248);
    platform2.setLocation(502,245);
    platform2.setLocation(503,246);
    platform platform3 = new platform();
    addObject(platform3,578,241);
    platform3.setLocation(591,250);
    platform platform4 = new platform();
    addObject(platform4,57,341);
    platform4.setLocation(48,346);
    platform platform5 = new platform();
    addObject(platform5,148,344);
    platform5.setLocation(148,345);
    platform platform6 = new platform();
    addObject(platform6,239,351);
    platform6.setLocation(300,282);
    platform4.setLocation(74,391);
    platform platform7 = new platform();
    addObject(platform7,382,280);
    platform7.setLocation(404,285);
    platform2.setLocation(516,289);
    platform6.setLocation(317,276);
    platform.setLocation(273,350);
    platform platform8 = new platform();
    addObject(platform8,323,354);
    platform8.setLocation(349,344);
    platform8.setLocation(347,345);
    platform3.setLocation(412,342);
    platform4.setLocation(50,390);
    platform4.setLocation(10,377);
    platform4.setLocation(10,377);
    platform4.setLocation(10,377);
    platform3.setLocation(469,339);
    platform8.setLocation(573,343);
    platform platform9 = new platform();
    addObject(platform9,364,344);
    platform9.setLocation(364,343);
    platform9.setLocation(363,347);
    platform platform10 = new platform();
    addObject(platform10,208,343);
```

```
platform10.setLocation(210,350);
platform7.setLocation(439,266);
platform2.setLocation(487,275);
platform platform11 = new platform();
addObject(platform11,580,277);
platform11.setLocation(580,277);
platform8.setLocation(553,353); ball
ball = new ball();
addObject(ball,497,144);
ball.setLocation(502,158);
removeObject(ball);
removeObject(sprite);
platform8.setLocation(508,344);
platform3.setLocation(572,343);
platform10.setLocation(214,344);
platform7.setLocation(391,270);
platform platform12 = new platform();
addObject(platform12,239,271);
platform12.setLocation(226,277);
platform12.setLocation(226,277);
platform platform13 = new platform();
addObject(platform13,520,197);
platform13.setLocation(558,206);
platform platform14 = new platform();
addObject(platform14,437,204);
platform14.setLocation(452,212);
platform platform15 = new platform();
addObject(platform15,360,207);
platform15.setLocation(350,210);
platform platform16 = new platform();
addObject(platform16,379,141);
platform16.setLocation(406,128);
platform platform17 = new platform();
addObject(platform17,494,142);
platform17.setLocation(507,134);
platform platform18 = new platform();
addObject(platform18,572,130);
platform18.setLocation(583,140);
platform18.setLocation(582,138);
platform platform19 = new platform();
addObject(platform19,415,348);
platform19.setLocation(439,347);
platform platform20 = new platform();
addObject(platform20,353,70);
platform platform21 = new platform();
addObject(platform21,265,72);
platform21.setLocation(256,69);
platform platform22 = new platform();
addObject(platform22,142,68);
platform22.setLocation(168,66);
platform platform23 = new platform();
addObject(platform23,41,68);
platform23.setLocation(62,72);
platform platform24 = new platform();
addObject(platform24,80,374);
platform24.setLocation(93,381);
```

```
platform24.setLocation(90,373);
platform7.setLocation(405,272);
platform16.setLocation(419,152);
platform17.setLocation(485,96);
platform18.setLocation(532,154);
platform17.setLocation(512,92); ball
ball2 = new ball();
addObject(ball2,231,207);
ball ball3 = new ball();
addObject(ball3,427,107);
ball ball4 = new ball();
addObject(ball4,328,26);
ball ball5 = new ball();
addObject(ball5,123,291);
spikey spikey = new
spikey();
addObject(spikey,218,306);
spikey spikey2 = new spikey();
addObject(spikey2,334,242);
spikey.setLocation(201,308);
spikey.setLocation(300,310);
spikey2.setLocation(390,243);
spikey spikey3 = new spikey();
addObject(spikey3,501,239);
spikey spikey4 = new spikey();
addObject(spikey4,485,315);
platform14.setLocation(424,198)
;
platform17.setLocation(554,108
) ;
platform20.setLocation(212,138
) ;
platform21.setLocation(76,142)
;
platform22.setLocation(166,91)
; ball4.setLocation(27,17);
ball3.setLocation(418,108);
ball3.setLocation(192,35);
ball2.setLocation(76,94);
platform21.setLocation(82,134);
platform20.setLocation(196,168
) ;
platform16.setLocation(412,160
) ; ball5.setLocation(418,68);
ball ball6 = new ball();
addObject(ball6,501,39);
ball ball7 = new ball();
addObject(ball7,348,49);
ball ball8 = new ball();
addObject(ball8,359,116);
ball ball9 = new ball();
addObject(ball9,257,54);
ball ball10 = new ball();
addObject(ball10,85,239);
ball ball11 = new ball();
addObject(ball11,55,313);
```

```
        ball ball12 = new ball();
        addObject(ball12,219,232);
        ball ball13 = new ball();
        addObject(ball13,154,299);
        ball4.setLocation(60,202);
     platform16.setLocation(537,27
) ;
platform17.setLocation(574,104)
;
platform18.setLocation(67,195);
ball2.setLocation(158,131);
spikey spikey5 = new
spikey();
addObject(spikey5,54,158);

}
}
```

Credit Scene Image



Creditscene Code

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and
MouseInfo)// uses the method called greenfoot


// applying extend use of polymorphism, inheritance, and methods.
```

```
public class Creditscene extends World



 {
   /**

    * Constructor for objects of class Creditscene

    *

    */
   //make a class from an object called CreditScene

public Creditscene()

    {

        // Create a new world with 600x400 cells with a cell size of 1x1 pixels.

        // add certain texts to be display & the size of the world as a whole.

        super(600, 400, 1);



        showText("Thank you for playing",400, 320);

        showText("We hope that you add fun playing the game", 400,290);

        showText("Press q to go back to start menu", 400,250);



        showText("My Contact number is 081513145186", 400,220);

       showText("Music:https://www.youtube.com/watch?v=fJQqOzkcHjg", 400,205);
        //when binds the key q it will go automatically to the starting page.
But it is recommended to do so when you want to play some more games.
```


```
    }
}
```

Gameover Image

Gameover Code

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
/**
* // Make a class gameover.



*
```

```
* @author (Shravan Srinivasan)

*

*/

public class gameover extends World

 {

    /**

     * Constructor for objects of class gameover.

     *

     */

//Give the size and what the gameover looks like displayed on the image

   public gameover()



    {
        super(600, 400, 1);
    }

   public void act()

    {
        // Bind the key q to go to start. NOTE DO NOT CLICK RESET AS THE PURPOSE
OF THE GAME IS TO MAKE YOU LISTEN THE REPEATING SONGS UNTIL YOU CAN EITHER
FIGURE OUT HOW TO CHEAT OR TO GO TO THE CREDIT SCENE.

        if(Greenfoot.isKeyDown("q"))
```

```
            Greenfoot.setWorld(new startmenu());

        }

    }
```

Ball Image

Ball Code

import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and

MouseInfo) //The ball doesnt do much but stand in position waiting for the sprite

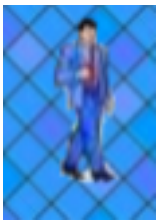
to touch it

public class ball extends Actor

{

}

Sprite Image



Sprite Code

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)




// Make the actor be able to stand on a platform.


//uses of custom-built classes and methods along with polymorphism, inheritance,
exception handling, primitive data types, and java collections.


public class sprite extends Actor


{ //call a out a method using GRAVITY=1


    private final int GRAVITY= 1;


    private int velocity;// set the velocity up


    public sprite() { // another class called sprite


        velocity=0;// set the velocity number equal to hero


    }
```

```
    MyWorld thisGame;//use this for acting out the world.
    public void act()

    { // implements all the movement the sprite needs to do.

        fall();

        if (Greenfoot.isKeyDown("space"))jump();//

        move();//call out move object class as the method from move();

        Actor gold= getOneIntersectingObject(ball.class);

        if(isTouching(spikey.class))

        {

            Greenfoot.setWorld(new gameover());// if sprite touches spikey it goes
to gameover.

            Greenfoot.playSound("Gameover.wav");

        }

        if(gold != null)

        {
```

```
            Greenfoot.playSound("Cha-Ching.wav");

            getWorld().removeObject(gold);// make the gold disappear when sprite
touches it.

            thisGame.score++;// add the score whenever the sprite eats up the golden
ball

    }

}

    public void fall() {

        setLocation(getX(), getY() + velocity);// make sure the the fall of sprite
touches the platform

        if (isOnSolidGround()) velocity=0;

        else velocity += GRAVITY;

    }

    public void jump() {

        velocity = -10;// when it jumps so that it wont have any errors.

    }

    public void move() {// start the movement with binding up, down, left, and right
key

        int y = getY();
```

```
        int x = getX();


        if(Greenfoot.isKeyDown("Up")) y-=3;


        if(Greenfoot.isKeyDown("Down")) y+=3;
        if(Greenfoot.isKeyDown("Left")) x-=3;


        if(Greenfoot.isKeyDown("Right"))x+=3;


        setLocation(x,y);


    }


    public boolean isOnSolidGround() {// call out a method boolean with function
isOnSolidGround as well as primitive data type.


        boolean isOnGround = false;







        if(getY() > getWorld().getHeight() - 50)isOnGround=true;// implements
exception handling







        int imageWidth = getImage().getWidth();


        int imageHeight = getImage().getHeight();


        if (getOneObjectAtOffset(imageWidth / -2, imageHeight /2, platform.class) !=
null ||// exception handling


            getOneObjectAtOffset(imageWidth / -2, imageHeight /2, platform.class) !=
null)


            isOnGround= true;
```

```
        return isOnGround;// return the class
    }



}
```

Spikey Image

Spikey Code

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/**



* Write a description of class spikey here.


*
* @author (your name)


* @version (a version number or a date)


*/
```

```
// this doesnt have to do much just like the golden ball.
public class spikey extends Actor
 {



 }
```

Platform Image

Platform Code

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and
MouseInfo) //declare class for object called platform

public class platform extends Actor
{
  //Give location of platform
   public platform() {
       this(100,25);
   }
  //gives the whole image sizes
   public platform(int width, int height) {
        GreenfootImage image = getImage();
        image.scale(width,height);
       setImage(image);
   }
}
```