IT PAT Technical Document

Lost

Kian Anderson

Table of Contents

1. Persistent Storage

User Array Constructor

The User Array Constructor reads all of the users from the Users.txt text file and creates an array from the users to be used later.

```
public UserArray() throws FileNotFoundException
{

      Scanner sc = new Scanner(new File("Users.txt"));
      while (sc.hasNextLine())
      {
            String[] aArr = (sc.nextLine()).split("\\|");
            userArr[size] = new User(aArr[0], aArr[1], aArr[2],
Integer.parseInt(aArr[3]));

            size++;
      }
}
```

(code using the userArray class and catching error)

```
private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
      try {
            UserArray m = new UserArray();
            if (m.login(lblUsernameExist.getText(), lblPassExist.getText())) {
                  new MainScreen().setVisible(true);
                  this.dispose();
            } else {
                  lblError.setText("Password or Username is incorrect");
                  lblError.setBackground(Color.red);
            }
      } catch (FileNotFoundException ex) {
            lblError.setText("File not found");
      }
}
```

Updating the Text File

When the array is changed in any way the update method allows for the text file to be updated from the array by writing the array back into the text file.

```
public void update() throws IOException
{
      BufferedWriter writer = null;
      writer = new BufferedWriter(new FileWriter(".\\Users.txt"));
      writer.write(toString());
      writer.close();

}
```

(Catching the error)

```
try {
      user.adduser();
      update();
}
catch (IOException ex)
{
      lblError.setText(“File not found”);
}
```

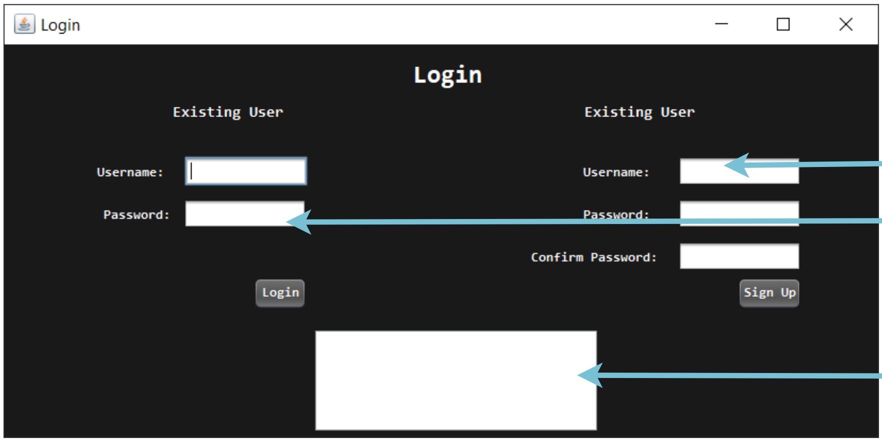
Reading the Help Topics

The Help Topics are read from the Help.txt text file and stored in an array to be used to display the help topics on the Help Screen.

```
public String[] helpArr = new String[10];
public int size = 0;
public int pos = 0;

public HelpScreen() {
      initComponents();

      try
      {
            Scanner sc = new Scanner(new File("Help.txt"));

            while(sc.hasNext())
            {
                  helpArr[size] = sc.nextLine();
                  size++;
            }

            } catch (FileNotFoundException ex)
            {
                  lblError.setText(“Help File not found”);
            }
      }
```

2. Defensive Programming

Sign Up



Two separate sections for Log In and Sign Up for simplicity and to limit user error.

Error messages appear here to inform users on errors.

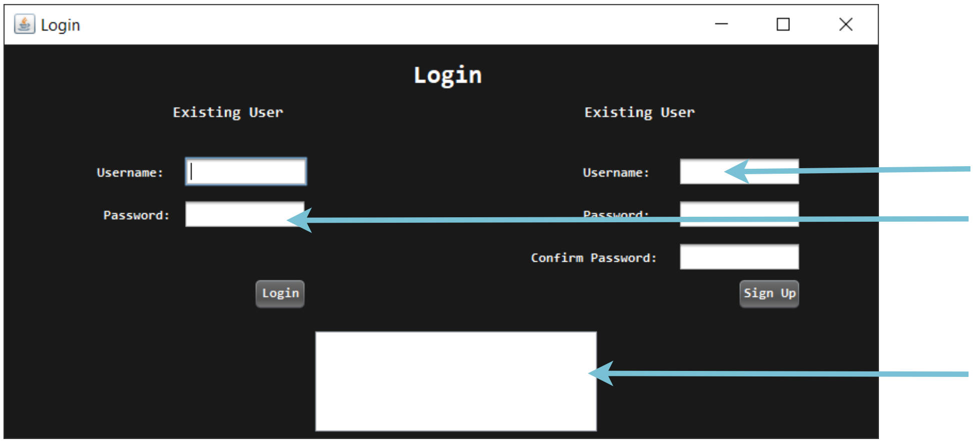
- In the sign up screen, when the user presses the sign in button, the two passwords are checked to see if they both match to ensure that the user has entered the desired password and if that is that case the user will be alerted by an error message in the bottom of the screen.

- When the user presses the sign up button, the username is checked to see if it already is in the database, if so the user will be asked to log in through an error message.

(code for sign up button)

```
private void btnSignUpActionPerformed(java.awt.event.ActionEvent evt)
{
      UserArray m = new UserArray();
      if (m.getUser(txtUsernameNew.getText()) != -1)
      {
            if ((txtPassNew1.getText()).equals(txtPassNew2.getText()))
            {
                  m.addUser(txtUsernameNew.getText(), txtPassNew1.getText(), "Not
                              completed#Not completed#Not completed", 1);

                  new MainScreen().setVisible(true);
                  this.dispose();
            } else
            {
                  lblArea.setText("Passwords do not match");
                  lblArea.setBackground(Color.red);
            }
      } else
      {
            lblArea.setText("User already exists");
            lblArea.setBackground(Color.red);
      }
}
```

Login



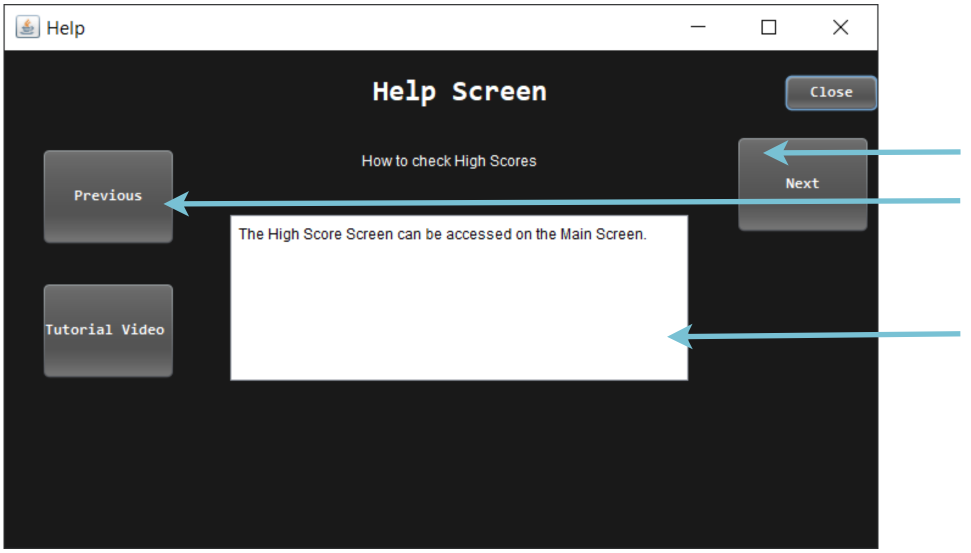
Two separate sections for Log In and Sign Up for simplicity and to limit user error.

Error messages appear here to inform users on errors.

- Checks if the user exists on the text file and whether the username and password match to ensure the user exists.

```
public Boolean loginUser(String user, String password)
{
      boolean b = false;
      if (getPassword(user).equals(password)) {
            b = true;
            makeCurrent(user);
      }

      return b;
}
```

Help Screen



The High Score Screen can be accessed on the Main Screen.

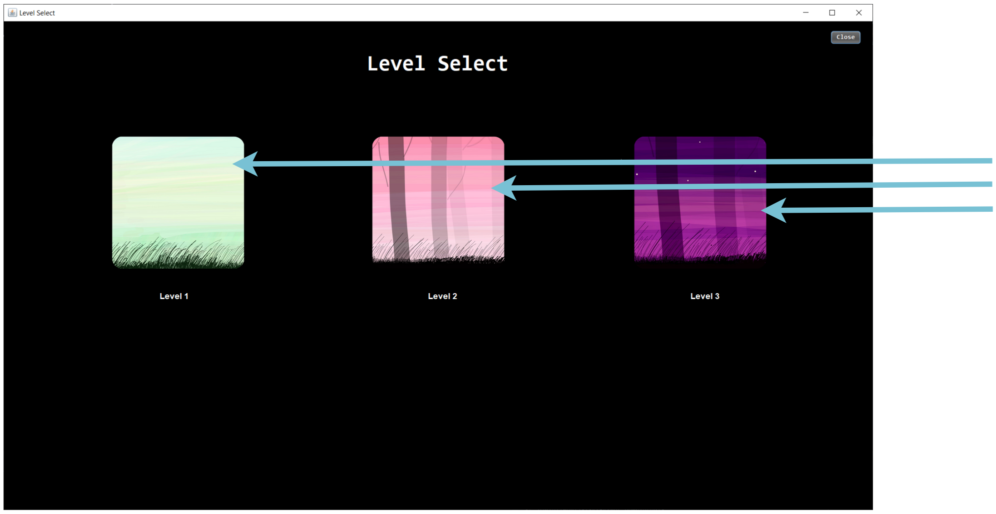
Two simple buttons to cycle help topics to limit user input.

Displays the relevant help information the user has selected.

- The Help Screen is comprised of two buttons to cycle endlessly through the help topics, this is to limit the user input and avoid user error.

```
private void btnCycle(java.awt.event.ActionEvent evt)
{
      int i = Integer.parseInt(evt.getActionCommand());
      pos += i;

      if (pos > 9) pos = 0;
      if (pos < 0) pos = 9;

      String[] s = helpArr[pos].split("#");

      lblTitle.setText(s[0]);
      lblMessage.setText(s[1]);
}
```

Level Select Screen



Three buttons for each level to limit user input.

- The Level Select Screen is comprised of 3 buttons for each level the user would like to go, but the user cannot select the level unless the have finished the previous level.

```
private void btnLvlActionPerformed(java.awt.event.ActionEvent evt)
{
      String lvl = evt.getActionCommand();

      if (!user.getBestTimes(0, Integer.parseInt(lvl) - 1).equals("Not completed"))
      {
            switch (lvl)
            {
                  case "1":
                        Level1Screen m = new Level1Screen();
                        break;

                  case "2":
                        Level2Screen m = new Level2Screen();
                        break;
```

```
              case "3":
                    Level3Screen m = new Level3Screen();
                    break;
        }

        Thread t = new Thread(m);
        t.start();
        m.setVisible(true);
        this.dispose();
    }
}
```

3. Externally sourced code

Video Player

Playing a video, used for a tutorial.

```
Desktop.getDesktop().open(new File("Tutorial.MOV"));
```

4. Critical Algorithms

Key Listener

The key listener takes user input from the keyboard and translates it into character movement.

```
keyListener
    keyPressed
        if (right key is pressed)
            character direction ⟵ 5
            character icon ⟵ right character image
        end if


        if (left key is pressed)
            character direction ⟵ -5
            character icon ⟵ left character image
        end if


        if (up Key is pressed and player jumping is false)
            jumping ⟵ true
            timer for jumping ⟵ 350 milliseconds
            jumping ⟵ false
            falling ⟵ true
            if (character collides with floor)
                falling ⟵ false
            end if
        end if
    end keyPressed


    keyReleased
        if (right key is released or Left Key is released)
            player stop
        end if
    end keyReleased

end keyListener
```

Animations

The animations change the image icon of the character depending on which direction the character is moving.

```
if (player direction is left)
    player icon ⟵ left image
end if


if (player direction is right)
    player icon ⟵ right player image
end if
```

Character movement

The

```
if (foreground x position ⟵ initial position)
    if (character direction is right)
        character move right
    end if

    if (character direction if left)
        character move left
    end if
end if
```

World Move

```
if (character position > reference location)
    for (amount of objects)
        move all objects direction of character
    end for
end if
```

Character Jumping

```
if (character is jumping)
    character jump
end if

if (character collides with floor)
    player falling ⟵ false
```

```
end if
```

Collide with Finish Line

```
if (character collides with finish)
    display "Level Finished"
    add 100 currency
    close game
    close screen
end if
```

Colliding with Obstacle

```
if (character collides with obstacle)
    kill character
    character location ⟵ start of level
end if
```

Colliding with Chest

```
if (character collides with chest)
    add 100 currency
    dispose chest
end if
```

5. Advanced Techniques

Object Collision

This code returns a Boolean when two objects intersect, which is used for detecting if the player has collided with any obstacles to make them die, and also used for the floor in order to make the character stop falling.

(check collision method)

```
public boolean checkCollision(JLabel labelOne, JLabel labelTwo)
{
      Area areaOne = new Area(labelOne.getBounds());
      Area areaTwo = new Area(labelTwo.getBounds());
      return areaOne.intersects(areaTwo.getBounds2D());
}
```

(example of collision method in use)

```
if (c.checkCollision(sprite, finish))
{
      finishTime = System.currentTimeMillis();
      UserArray u = new UserArray();
      u.editTimes(0, 0, "" + (finishTime - startTime)/1000);
      lblUserMessage.setText("Level 1 Complete");
      currency += 100;
      lblCurrency.setText("Currency: " + currency);
      gameClosed = true;
      new Level1Screen().dispose();
}
```

Animations

This code makes every object that needs to be animated, cycle through a set of images that make the object appear that it is moving, for example the obstacles are saws, cycling through and array of 3 images these appear to be rotating, the same applies to the character, he has a walk animation.

(array in animations class of image icons for a right facing character)

```
public ImageIcon playerRightImageArr(int i)
{
      ImageIcon[] playerLeft = new ImageIcon[]{
(new javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite1R.png"))),
(new javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite2R.png"))),
(new javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite3R.png"))),
(new javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite4R.png"))),
(new javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite5R.png"))),
(new javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite6R.png")))};

      return playerLeft[i];
}
```

(example of it being used)

```
if (player.getDirection() == 5 && !player.isJumping() && !player.isFalling())
{
      sprite.setIcon(animation.playerRightImageArr(e));
}
```

Thread

The movement of the character is a thread, this allows for the character to move on every ‘tick’ of the game (approximately 30 times every second), this allows the character to have smooth movement and for it to appear to be moving and all the character movement takes place in this thread.

(example of thread sleeping)

```
try
{
      //amount of time in milliseconds for thread to sleep
      Thread.sleep(1000);
}
catch (InterruptedException ex)
{
      lblError.setText(Thread not found);
}
```

6. Test Plan and Results

User Test 1:

Tested by Matt Oosthuizen (9 June 2020)

Log in Screen	
The user is presented with a text field where their username is put in and another text field where the user must enter their password	Yes
Once the fields are filled out the user presses the login button and the username and password will be checked against the database to see if they are legitimate.	Yes
If it is the users first time logging in, there is a Sign Up button that the user must press that will allow the user to enter a username and password with a confirm password field that will add the user to the database	Yes
Once one of these tasks have been completed the user will be taken to the Main Screen.	Yes
Help Screen	
The user will be able to access this screen from the Main Screen	Yes
This screen will have different help tips that the user can select, depending on what the user has selected	Yes
The information will be pulled from the database and displayed on the screen for the user to read the relevant information	Yes
The screen will have a close button that will close the screen and return the player to the level select screen.	Yes
Best Time Screen	
This screen will be accessible through the Main Screen by use of a button	Yes
This screen will display the players best time achieved on every level in a table format that can be easily read by the user	Yes
The data will be pulled from the from the user class which is pulled from the database.	Yes
The screen will have a close button that will close the screen and return the player to the level select screen.	Yes
The screen will display the best time that has been achieved on the whole database by any user and compare it to the current player.	Yes
Game Screen	
The Game Screen will consist of the player the is centered in the middle of the screen with the background and the chosen level foreground and floor.	Yes
It will consist of a pause button which will bring up a lead button to quit the level, as well as pausing the game	Yes
The pause menu will also consist of a volume scroller to adjust the audio level and a mute button.	Yes
There will be a floor designed specifically for each level that will utilize collision detection, this will slow for the character to not fall below the cave floor	Yes
The character will spawn at the beginning of the level in a cave when they first start.	Yes
If the player touches an obstacle, they will die.	Yes
When a player collides with a platform the player jumping and falling will be set to false and the character will appear to walk in the platform.	Yes
To finish the level the player must reach the finish line.	Yes

If the player reaches a chest during the level 50 currency will be added to the database and if the player reaches the finish line 100 currency will be added to their database under the user name.	Yes
If the player dies there will be a timer of 1 second after the player touches the obstacle which will add to the user experience and allow the player to recognize that they have died.	Yes
A notification will be displayed "You died" when the player touches and obstacle.	Yes
The user will have to complete the previous level to unlock the next level, when the user logs in, their level progress will be saved.	Yes
A timer will begin as the player makes their first movement, this will only end when the player finishes the level.	Yes
Depending on the level the user selects, the positioning of the obstacles and platforms will change to make the level more challenging.	Yes
Level Select Screen	
This screen will allow the player to select a level, but the user must have completed the previous level to access the level	Yes
Otherwise it will be displayed as locked and the player will no be able to select it.	Yes
When a level is selected a runnable thread is initialised in that level and the level screen is set to visible.	Yes
Main Screen	
This screen will consist of a button to allow the user to go to the level select screen	Yes
A button to get to the Level Select Screen	Yes
A button to get to the Help Screen to display help.	Yes
A button that will open a screen that will display the player with their best times achieved on every level and the best time that has been achieved on the whole database by any user.	Yes

User Test 2:

Tested by Luke Payne (18 June 2020)

Log in Screen	
The user is presented with a text field where their username is put in and another text field where the user must enter their password	Yes
Once the fields are filled out the user presses the login button and the username and password will be checked against the database to see if they are legitimate.	Yes
If it is the users first time logging in, there is a Sign Up button that the user must press that will allow the user to enter a username and password with a confirm password field that will add the user to the database	Yes
Once one of these tasks have been completed the user will be taken to the Level Select Screen.	Yes
Help Screen	
The user will be able to access this screen from the Main Screen	Yes
This screen will have different help tips that the user can select, depending on what the user has selected	Yes
The information will be pulled from the database and displayed on the screen for the user to read the relevant information	Yes
The screen will have a close button that will close the screen and return the player to the level select screen.	Yes
Best Time Screen	
This screen will be accessible through the Main Screen by use of a button	Yes
This screen will display the players best time achieved on every level in a table format that can be easily read by the user	Yes
The data will be pulled from the from the user class which is pulled from the database.	Yes
The screen will have a close button that will close the screen and return the player to the level select screen.	Yes
The screen will display the best time that has been achieved on the whole database by any user and compare it to the current player.	Yes
Game Screen	
The Game Screen will consist of the player the is centered in the middle of the screen with the background and the chosen level foreground and floor.	Yes
It will consist of a pause button which will bring up a lead button to quit the level, as well as pausing the game	Yes
The pause menu will also consist of a volume scroller to adjust the audio level and a mute button.	Yes
There will be a floor designed specifically for each level that will utilize collision detection, this will slow for the character to not fall below the cave floor	Yes
The character will spawn at the beginning of the level in a cave when they first start.	Yes
If the player touches an obstacle, they will die.	Yes
When a player collides with a platform the player jumping and falling will be set to false and the character will appear to walk in the platform.	Yes
To finish the level the player must reach the finish line.	Yes
If the player reaches a chest during the level 50 currency will be added to the database and if the player reaches the finish line 100 currency will be added to their database under the user name.	Yes

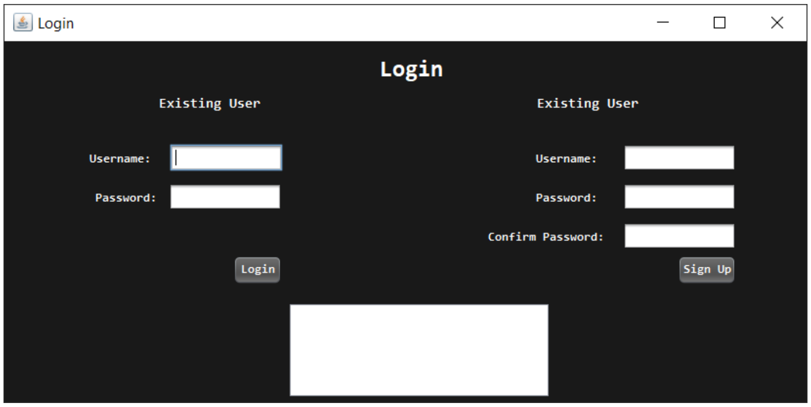
If the player dies there will be a timer of 1 second after the player touches the obstacle which will add to the user experience and allow the player to recognize that they have died.	Yes
A notification will be displayed "You died" when the player touches and obstacle.	Yes
The user will have to complete the previous level to unlock the next level, when the user logs in, their level progress will be saved.	Yes
A timer will begin as the player makes their first movement, this will only end when the player finishes the level.	Yes
Depending on the level the user selects, the positioning of the obstacles and platforms will change to make the level more challenging.	Yes
Level Select Screen	
This screen will allow the player to select a level, but the user must have completed the previous level to access the level	Yes
Otherwise it will be displayed as locked and the player will no be able to select it.	Yes
When a level is selected a runnable thread is initialised in that level and the level screen is set to visible.	Yes
Main Screen	
This screen will consist of a button to allow the user to go to the level select screen	Yes
A button to get to the Level Select Screen	Yes
A button to get to the Help Screen to display help.	Yes
A button that will open a screen that will display the player with their best times achieved on every level and the best time that has been achieved on the whole database by any user.	Yes

Data Validation

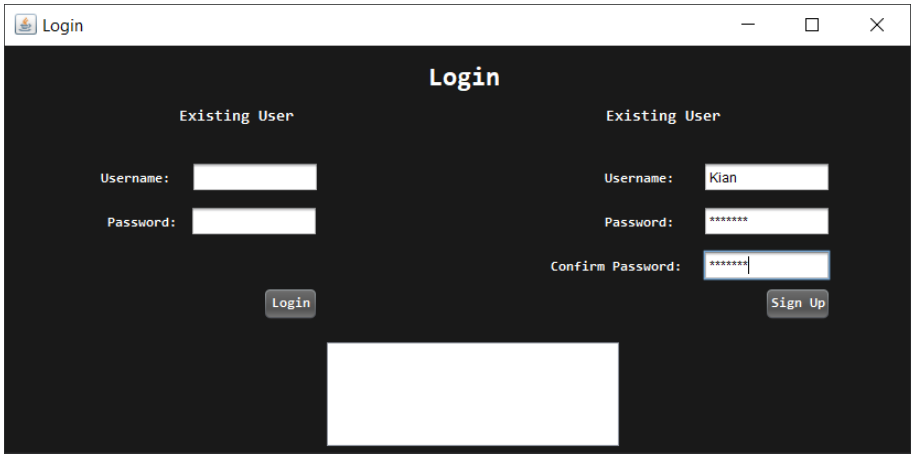
Sign Up:

Field	Standard Data	Extreme Data	Abnormal Data
Username	Kian	ffffffff	No username entered
Password	KianAnd	ddddd	No password entered
Confirm Password	KianAnd	ddddd	No password entered

Screen before data is entered:

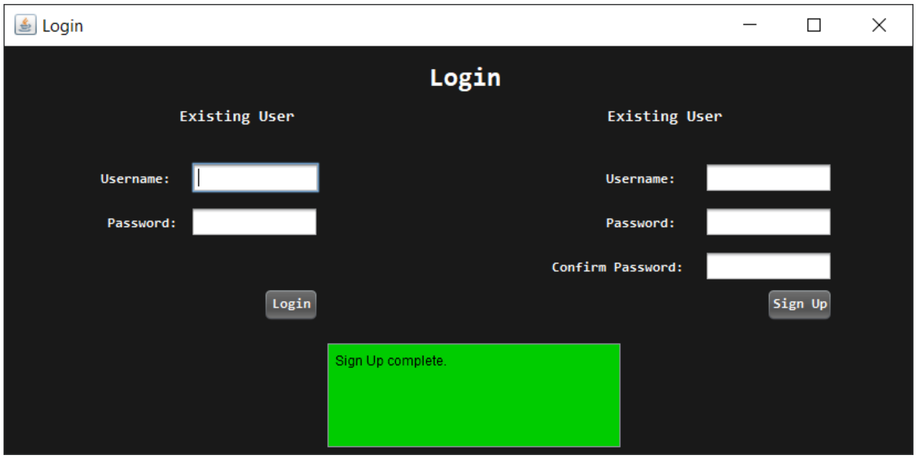


Standard:
Before:



Username: Kian
Password: *******
Confirm Password: *******

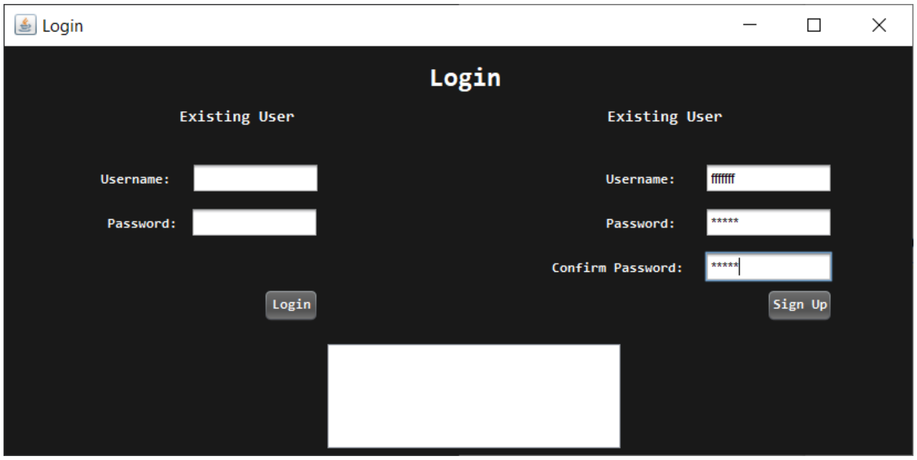
After:



Sign Up complete.

Extreme

Before:



Username:

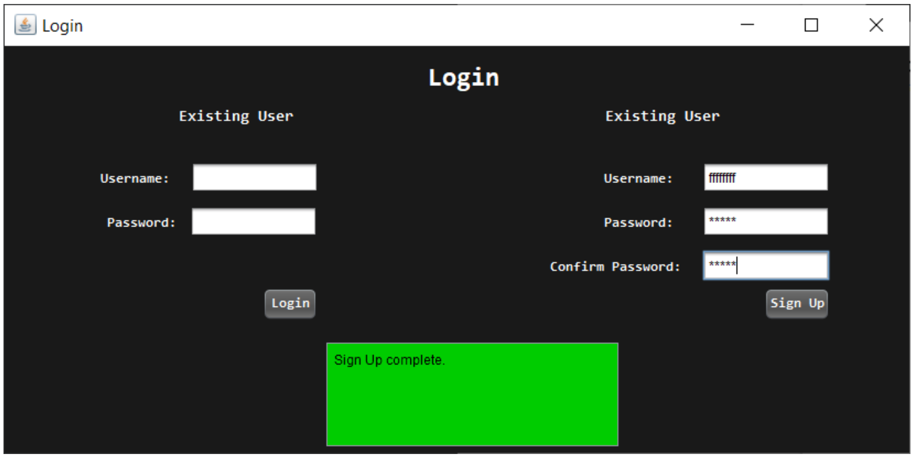
Password:

Username: ffffffff

Password: *****

Confirm Password: *****

After:



Username:

Password:

Username: ffffffff

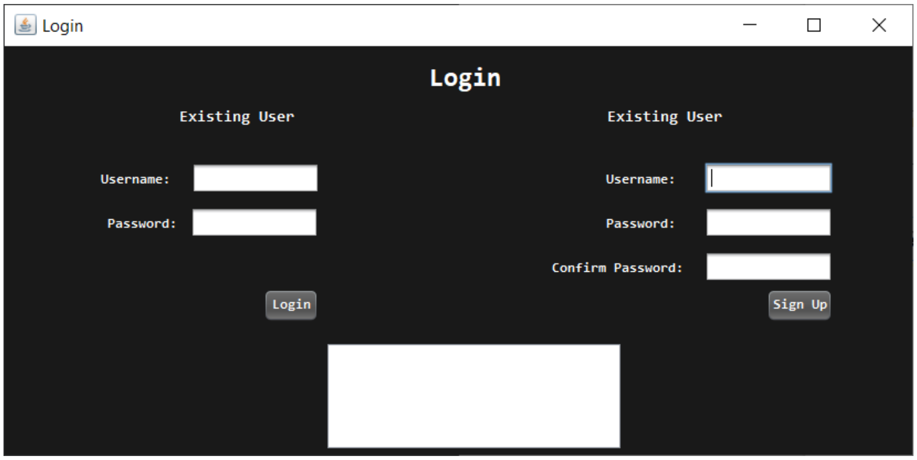
Password: *****

Confirm Password: *****

Sign Up complete.

Abnormal:

Before:



Username:

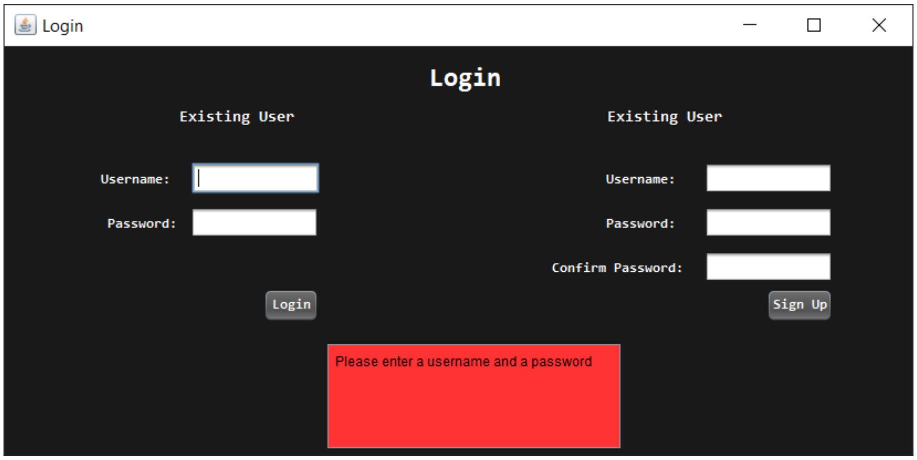
Password:

Username:

Password:

Confirm Password:

After:



Username:

Password:

Username:

Password:

Confirm Password:

Please enter a username and a password

7. Code Dump

Animations Class

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package anderson;

import javax.swing.ImageIcon;
import javax.swing.JLabel;

/**
 *
 * @author Kian
 */
public class Animations {

    //image array of saws
    public ImageIcon sawImageArr(int i)
    {
        ImageIcon[] saws = new ImageIcon[]{(new
javax.swing.ImageIcon(getClass().getResource("/anderson/Images/saw1.PNG"))),
                            (new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw2.PNG"))),
                            (new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw3.PNG")))};
        return saws[i];
    }

    //image array of player right
    public ImageIcon playerRightImageArr(int i)
    {
        ImageIcon[] playerRight = new ImageIcon[]{(new
javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite1.PNG"))),
                                                   (new
javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite2.PNG"))),
                                                   (new
javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite3.PNG"))),
                                                   (new
javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite4.PNG"))),
                                                   (new
javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite5.PNG"))),
                                                   (new
javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite6.PNG")))};
        return playerRight[i];
    }
    //image array of player left
    public ImageIcon playerLeftImageArr(int i)
    {
        ImageIcon[] playerLeft = new ImageIcon[]{(new 
javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite1R.png"))),
                                                    (new 
javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite2R.png"))),
                                                    (new 
javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite3R.png"))),
                                                    (new 
javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite4R.png"))),
                                                    (new 
javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite5R.png"))),
                                                    (new 
javax.swing.ImageIcon(getClass().getResource("/anderson/Images/sprite6R.png")))};
        return playerLeft[i];
    }


}
```

Character Class

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package anderson;

/**
 *
 * @author Kian
 */
public class Character {

    private int x;
    private int y;
    private int direction;
    private boolean jumping;
    private boolean falling;

    //constructor
    public Character(int x, int y, int direction) {
        this.x = x;
        this.y = y;
        this.direction = direction;
        jumping = false;
        falling = false;
    }

    //getters
    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }

    public void setY(int i) {
        x = i;
    }

    public void setX(int i) {
        y = i;
    }

    public int getDirection() {
        return direction;
    }
    //setters
    public void setLeft() {
        direction = 5;
    }

    public void setRight() {
        direction = -5;
    }

    //makes charcter stop
    public void stop() {
        direction = 0;
    }

    //controls character horizontal movement
    public void move() {
        x += direction;
    }

    //controls character vertical movement
    public void moveVert() {
        if (jumping) {
            y -= 5;
        } else if (falling) {
            y += 5;
        }
    }

    //jumping and falling getters
    public boolean isJumping() {
        return jumping;
    }

    public boolean isFalling() {
        return falling;
    }

    //jumping and falling setters
    public void setJumping(boolean j) {
        jumping = j;
    }

    public void setFalling(boolean f) {
        falling = f;
    }

}
```

Collision Detection Class

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package anderson;

import java.awt.geom.Area;
import javax.swing.JLabel;


/**
 *
 * @author Kian
 */
public class CollisionDetection
{

    Character player = new Character(0, 800, 0);

    //check collision between two labels, returns boolean
    public boolean checkCollision(JLabel labelOne, JLabel labelTwo)
    {
        Area areaOne = new Area(labelOne.getBounds());
        Area areaTwo = new Area(labelTwo.getBounds());
        return areaOne.intersects(areaTwo.getBounds2D());
    }


}
```

User Class

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package anderson;

/**
 * Handles the user information for a specific user.
 * @author Kian
 */
public class User
{
    private String username;
    private String password;
    private String bestTimes;
    private int level;

    /**
     * Constructor for the user class.
     * @param user The username of the user.
     * @param password The password of the user.
     * @param bestTimes The times of the user.
     * @param level The level that the user is currently busy with.
     */
    public User(String user, String password, String bestTimes, int level)
    {
        this.username = user;
        this.password = password;
        this.bestTimes = bestTimes;
        this.level = level;
    }

    /**
     * Gets the username of the user.
     * @return The username of the user.
     */
    public String getUsername()
    {
        return username;
    }

    /**
     * Gets the password of the user.
     * @return The password of the user.
     */
    public String getPassword()
    {
        return password;
    }

    /**
     * Gets the best times of the user.
     * @return The best times of the user.
     */
    public String getBestTimes()
    {
        return bestTimes;
    }

    /**
     * Gets the level that the user is currently on.
     * @return The level that the user is currently on.
     */
    public int getLevel()
    {
        return level;
    }

    /**
     * Sets the username of the user.
     * @param user The desired new username.
     */
    public void setUsername(String user)
    {
        this.username = user;
    }

    /**
     * Sets the password of the user.
     * @param password The desired password of the user.
     */
    public void setPassword(String password)
    {
        this.password = password;
    }

    /**
     * Sets the best times of the user.
     * @param bestTimes
     */
    public void setBestTimes(String bestTimes)
    {
        this.bestTimes = bestTimes;
    }

    /**
     * Sets the level of the user.
     * @param level The desired level of the user.
     */
    public void setLevel(int level)
    {
        this.level = level;
    }

    public String toString()
    {
        return username + "|" +password + "|" + bestTimes + "|" + level;
    }

    /**
     * Sets all the variables of the user in one method.
     * @param a The username of the user.
     * @param b The password of the user.
     * @param c The best times of the user.
     * @param d The level of the user.
     */
    public void setUser(String a,String b,String c,int d)
    {
        username = a;
        password = b;
        bestTimes = c;
        level = d;

    }

}
```

User Array Class

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package anderson;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 * Handles the all the users information in an array.
 *
 * @author Kian
 */
public class UserArray
{

    private User[] userArr = new User[50];
    private int size;

    /**
     * Constructor for the UserArray class, reads the text file into an array.
     */
    public UserArray()
    {
        try
        {
            Scanner sc = new Scanner(new File("Users.txt"));
            while (sc.hasNextLine())
            {

                String[] aArr = (sc.nextLine()).split("\\|");
                userArr[size] = new User(aArr[0], aArr[1], aArr[2],
Integer.parseInt(aArr[3]));

                size++;
            }
        } catch (FileNotFoundException ex)
        {
            Logger.getLogger(UserArray.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public String toString()
    {
        String s = "";
        for (int i = 0; i < size; i++)
        {
            s += userArr[i].toString() + "\n";

        }
        return s;
    }

    /**
     * Updates the text file from the array.
     */
    public void update()
    {
        BufferedWriter writer = null;
        try
        {
            writer = new BufferedWriter(new FileWriter(".\\Users.txt"));
            writer.write(toString());
            writer.close();
        } catch (IOException ex)
        {
            Logger.getLogger(UserArray.class.getName()).log(Level.SEVERE, null, ex);
        }

    }

    /**
     * Gets the users position from the username.
     *
     * @param name The user to be found.
     * @return The position of the user.
     */
    public int getUser(String name)
    {
        int i = 0;
        boolean flag = false;
        int pos = -1;
        while (i < size && flag == false)
        {
            if (name.equals(userArr[i].getUsername()))
            {
                flag = true;
                pos = i;
            }
            i++;
        }
        return pos;
    }

    /**
     * Deletes a user from the database.
     *
     * @param me The username of the user to be deleted.
     */
    public void delete(String me)
    {
        int temp = getUser(me);

        if (temp > 0)
        {
            shiftLeft(temp);
        }

    }

    /**
     * Gets the username of a user from a position in the array.
     *
     * @param pos The position of user in the array.
     * @return The username of the user.
     */
    public String getName(int pos)
    {
        return userArr[pos].getUsername();
    }

    /**
     * Gets the password of the user from a position in the array.
     *
     * @param pos The position of the user in the array.
     * @return The password of the user.
     */
    public String getPassword(int pos)
    {
        return userArr[pos].getPassword();
    }

    /**
     * Gets the best time for a user on a specific level.
     *
     * @param pos The position of the user in the array.
     * @param level The level for which the best time is being requested.
     * @return The best time for the user of that level.
     */
    public String getBestTimes(int pos, int level)
    {
        String[] sArr = (userArr[pos].getBestTimes()).split("\\#");
        return sArr[level - 1];
    }
    /**
     * Gets the level a user is currently on.
     *
     * @param pos The position of the user in the array.
     * @return The level the user is currently on.
     */
    public int getLevel(int pos)
    {
        return userArr[pos].getLevel();
    }

    /**
     * Edits the times of the user if the new times is better than the existing
     * time.
     *
     * @param pos The position of the user in the array.
     * @param level The level which has just been completed.
     * @param time The time taken by the user to complete the level.
     */
    public void editTimes(int pos, int level, String time)
    {
        String[] sArr = (userArr[pos].getBestTimes()).split("\\#");
        String times = userArr[pos].getBestTimes();

        if (sArr[level].equals("Not completed") || Integer.parseInt(time) <
Integer.parseInt(sArr[level]))
        {

            switch (level)
            {
                case 0:
                    times = time + "#" + sArr[1] + "#" + sArr[2];
                    break;

                case 1:
                    times = sArr[0] + "#" + time + "#" + sArr[2];
                    break;

                case 2:
                    times = sArr[0] + "#" + sArr[1] + "#" + time;
                    break;
            }
        }

        userArr[pos].setBestTimes(times);
        update();
    }

    /**
     * Returns a boolean if the user exists on the database, and if the username
     * and password match. Also makes the user the current user.
     *
     * @param user The username the user has entered.
     * @param password The password the user has entered.
     * @return A boolean whether the entered credentials are valid.
     */
    public boolean login(String user, String password)
    {
        if (userArr[getUser(user)].getPassword().equals(password))
        {
            makeCurrent(user);
            return true;

        } else
        {
            return false;
        }

    }

    /**
     * Shifts the array right from a desired position.
     *
     * @param position The position from which to shift from.
     */
    public void shiftRight(int position)
    {
        System.out.println(size);
        userArr[size] = new User(userArr[size - 1].getUsername(), userArr[size -
1].getPassword(), userArr[size - 1].getBestTimes(), userArr[size - 1].getLevel());
        for (int i = size - 1; i >= position; i--)
        {
            userArr[i].setUser(userArr[i - 1].getUsername(), userArr[i -
1].getPassword(), userArr[i - 1].getBestTimes(), userArr[i - 1].getLevel());
        }
        size++;
    }

    /**
     * Shifts the array left from a desired position.
     *
     * @param position The position from which to shift from.
     */
    public void shiftLeft(int position)
    {
        for (int i = position; i < size - 1; i++)
        {
            userArr[i] = userArr[i + 1];
        }
        size--;
    }

    /**
```

```
     * Adds a user to the text file and array.
     *
     * @param x The username of the user.
     * @param y The password of the user.
     * @param z The times of the user.
     * @param a The level that the user is currently on.
     */
    public void addUser(String x, String y, String z, int a)
    {
        shiftRight(1);
        userArr[0].setUser(x, y, z, a);
        update();
    }

    public void makeCurrent(String user)
    {
        String tempName = userArr[getUser(user)].getUsername();
        String tempPassword = userArr[getUser(user)].getPassword();
        String tempBestTime = userArr[getUser(user)].getBestTimes();
        int tempLevel = userArr[getUser(user)].getLevel();
        delete(user);
        addUser(tempName, tempPassword, tempBestTime, tempLevel);
        update();
    }

}
```

Best Time Screen

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package anderson;


public class BestTimeScreen extends javax.swing.JFrame
{

    public BestTimeScreen()
    {
        initComponents();
        UserArray m = new UserArray();
        txtArea.setText("Level 1: " + m.getBestTimes(0, 1) + "\nLevel 2: " +
m.getBestTimes(0, 2) + "\nLevel 3: " + m.getBestTimes(0, 3));

    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        pnlBackground = new javax.swing.JPanel();
        lblBestTimes = new javax.swing.JLabel();
        pnlScroll = new javax.swing.JScrollPane();
        txtArea = new javax.swing.JTextArea();
        btnClose = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Best Time");
        getContentPane().setLayout(null);

        pnlBackground.setBackground(new java.awt.Color(25, 25, 25));

        lblBestTimes.setFont(new java.awt.Font("Consolas", 1, 36)); // NOI18N
        lblBestTimes.setForeground(new java.awt.Color(255, 255, 255));
        lblBestTimes.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblBestTimes.setText("Best Times");

        pnlScroll.setBackground(new java.awt.Color(51, 51, 51));
        pnlScroll.setForeground(new java.awt.Color(255, 255, 255));
        txtArea.setBackground(new java.awt.Color(51, 51, 51));
        txtArea.setColumns(20);
        txtArea.setFont(new java.awt.Font("Consolas", 0, 24)); // NOI18N
        txtArea.setForeground(new java.awt.Color(255, 255, 255));
        txtArea.setRows(5);
        txtArea.setText("\n");
        pnlScroll.setViewportView(txtArea);

        btnClose.setBackground(new java.awt.Color(51, 51, 51));
        btnClose.setFont(new java.awt.Font("Consolas", 1, 14)); // NOI18N
        btnClose.setForeground(new java.awt.Color(255, 255, 255));
        btnClose.setText("Close");
        btnClose.setBorder(null);
        btnClose.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnCloseActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout pnlBackgroundLayout = new
javax.swing.GroupLayout(pnlBackground);
        pnlBackground.setLayout(pnlBackgroundLayout);
        pnlBackgroundLayout.setHorizontalGroup(

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlBackgroundLayout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(lblBestTimes, javax.swing.GroupLayout.PREFERRED_SIZE, 552,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(1372, Short.MAX_VALUE))
            .addGroup(pnlBackgroundLayout.createSequentialGroup()
                .addGap(28, 28, 28)
                .addComponent(pnlScroll, javax.swing.GroupLayout.PREFERRED_SIZE, 507,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addGroup(pnlBackgroundLayout.createSequentialGroup()
                .addGap(476, 476, 476)
                .addComponent(btnClose, javax.swing.GroupLayout.PREFERRED_SIZE, 59,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
        pnlBackgroundLayout.setVerticalGroup(

pnlBackgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlBackgroundLayout.createSequentialGroup()
                .addGap(32, 32, 32)
                .addComponent(btnClose, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(50, 50, 50)
                .addComponent(lblBestTimes, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(71, 71, 71)
                .addComponent(pnlScroll, javax.swing.GroupLayout.PREFERRED_SIZE, 431,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

        getContentPane().add(pnlBackground);
        pnlBackground.setBounds(-4, -14, 1930, 660);

        setSize(new java.awt.Dimension(574, 673));
        setLocationRelativeTo(null);
    }// </editor-fold>

    private void btnCloseActionPerformed(java.awt.event.ActionEvent evt)
    {
        new MainScreen().setVisible(true);
        dispose();
    }


    public static void main(String args[])
    {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default
look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/
lookandfeel/plaf.html
         */
        try
        {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels())
            {
                if ("Nimbus".equals(info.getName()))
                {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex)
        {

java.util.logging.Logger.getLogger(BestTimeScreen.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
        } catch (InstantiationException ex)
        {
```

```
java.util.logging.Logger.getLogger(BestTimeScreen.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex)
        {

java.util.logging.Logger.getLogger(BestTimeScreen.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex)
        {

java.util.logging.Logger.getLogger(BestTimeScreen.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
        }
        //</editor-fold>
        //</editor-fold>
        //</editor-fold>
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                new BestTimeScreen().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton btnClose;
    private javax.swing.JLabel lblBestTimes;
    private javax.swing.JPanel pnlBackground;
    private javax.swing.JScrollPane pnlScroll;
    private javax.swing.JTextArea txtArea;
    // End of variables declaration
}
```

Help Screen

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package anderson;

import java.awt.Desktop;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;


import java.util.Scanner;


/**
 *
 * @author 10027
 */
public class HelpScreen extends javax.swing.JFrame {



    public String[] helpArr = new String[10];
    public int size = 0;
    public int pos = 0;

    public HelpScreen() {
        initComponents();

        try
        {
            Scanner sc = new Scanner(new File("Help.txt"));

            while(sc.hasNext())
            {
                helpArr[size] = sc.nextLine();
                size++;
            }


        } catch (FileNotFoundException ex)
        {
            System.out.println("Fle not found");
        }

    }
    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        btnClose = new javax.swing.JButton();
        lblHelpScreen = new javax.swing.JLabel();
        btnNext = new javax.swing.JButton();
        btnTutorial = new javax.swing.JButton();
        btnPrevious = new javax.swing.JButton();
        txtTitle = new javax.swing.JLabel();
        pnlMessage = new javax.swing.JScrollPane();
        txtArea = new javax.swing.JTextArea();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Help");
        getContentPane().setLayout(null);

        jPanel1.setBackground(new java.awt.Color(25, 25, 25));
        jPanel1.setLayout(null);

        btnClose.setBackground(new java.awt.Color(51, 51, 51));
        btnClose.setFont(new java.awt.Font("Consolas", 1, 12)); // NOI18N
        btnClose.setForeground(new java.awt.Color(255, 255, 255));
        btnClose.setText("Close");
        btnClose.setBorder(null);
        btnClose.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnCloseActionPerformed(evt);
            }
        });
        jPanel1.add(btnClose);
        btnClose.setBounds(660, 30, 77, 30);

        lblHelpScreen.setFont(new java.awt.Font("Consolas", 1, 24)); // NOI18N
        lblHelpScreen.setForeground(new java.awt.Color(255, 255, 255));
        lblHelpScreen.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblHelpScreen.setText("Help Screen");
        jPanel1.add(lblHelpScreen);
        lblHelpScreen.setBounds(210, 30, 366, 30);

        btnNext.setBackground(new java.awt.Color(51, 51, 51));
        btnNext.setFont(new java.awt.Font("Consolas", 1, 12)); // NOI18N
        btnNext.setForeground(new java.awt.Color(255, 255, 255));
        btnNext.setText("Next");
        btnNext.setActionCommand("1");
        btnNext.setBorder(null);
        btnNext.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnCycle(evt);
            }
        });
        jPanel1.add(btnNext);
        btnNext.setBounds(620, 80, 110, 80);

        btnTutorial.setBackground(new java.awt.Color(51, 51, 51));
        btnTutorial.setFont(new java.awt.Font("Consolas", 1, 12)); // NOI18N
        btnTutorial.setForeground(new java.awt.Color(255, 255, 255));
        btnTutorial.setText("Tutorial Video ");
        btnTutorial.setBorder(null);
        btnTutorial.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnTutorialActionPerformed(evt);
            }
        });
        jPanel1.add(btnTutorial);
        btnTutorial.setBounds(50, 200, 110, 80);

        btnPrevious.setBackground(new java.awt.Color(51, 51, 51));
        btnPrevious.setFont(new java.awt.Font("Consolas", 1, 12)); // NOI18N
        btnPrevious.setForeground(new java.awt.Color(255, 255, 255));
        btnPrevious.setText("Previous");
        btnPrevious.setActionCommand("-1");
        btnPrevious.setBorder(null);
        btnPrevious.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnCycle(evt);
            }
        });
        jPanel1.add(btnPrevious);
        btnPrevious.setBounds(50, 90, 110, 80);

        txtTitle.setForeground(new java.awt.Color(255, 255, 255));
        txtTitle.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        txtTitle.setText("How to check High Scores");
        jPanel1.add(txtTitle);
        txtTitle.setBounds(230, 80, 310, 40);

        txtArea.setColumns(20);
        txtArea.setLineWrap(true);
        txtArea.setRows(5);
        txtArea.setText("The High Score Screen can be accessed on the Main Screen.");
        pnlMessage.setViewportView(txtArea);

        jPanel1.add(pnlMessage);
        pnlMessage.setBounds(203, 143, 380, 140);

        getContentPane().add(jPanel1);
        jPanel1.setBounds(-20, -10, 780, 450);
        setSize(new java.awt.Dimension(734, 455));
        setLocationRelativeTo(null);
    }// </editor-fold>

    private void btnCloseActionPerformed(java.awt.event.ActionEvent evt)
    {
        new MainScreen().setVisible(true);
        this.dispose();

    }

    private void btnTutorialActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            Desktop.getDesktop().open(new File("Tutorial.mov"));

        } catch (IOException ex) {
            System.out.println("File not found");
        }

    }

    private void btnCycle(java.awt.event.ActionEvent evt)
    {
        int i = Integer.parseInt(evt.getActionCommand());
        pos += i;

        if (pos > 9) pos = 0;
        if (pos < 0) pos = 9;

        String[] s = helpArr[pos].split("#");

        txtTitle.setText(s[0]);
        txtArea.setText(s[1]);
    }


    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code 
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default 
look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/
lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : 
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(HelpScreen.class.getName()).log(java.util.logging.Leve
l.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(HelpScreen.class.getName()).log(java.util.logging.Leve
l.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(HelpScreen.class.getName()).log(java.util.logging.Leve
l.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(HelpScreen.class.getName()).log(java.util.logging.Leve
l.SEVERE, null, ex);
        }
        //</editor-fold>
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new HelpScreen().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton btnClose;
    private javax.swing.JButton btnNext;
    private javax.swing.JButton btnPrevious;
    private javax.swing.JButton btnTutorial;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JLabel lblHelpScreen;
    private javax.swing.JScrollPane pnlMessage;
    private javax.swing.JTextArea txtArea;
    private javax.swing.JLabel txtTitle;
    // End of variables declaration
}
```

Level 1 Screen

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package anderson;

import java.awt.event.*;
import java.util.TimerTask;
import java.util.Timer;
import javax.swing.ImageIcon;
import javax.swing.JLabel;

/**
 *
 * @author Kian
 */
class Level1Screen extends javax.swing.JFrame implements Runnable
{

    Character player = new Character(540, 800, 0);
    boolean gameClosed = false;
    double initialPosX;
    double lastPosX;
    int currency = 0;
    boolean chestCollided = false;
    long startTime, finishTime;

    /**
     * Creates new form GameScreen
     */
    public Level1Screen()
    {
        initComponents();


        //setting up key listener
        KeyListener listener = new Level1Screen.MyKeyListener();
        addKeyListener(listener);
        setFocusable(true);

        //sets the initial location of the sprite
        sprite.setLocation(player.getX(), player.getY());
        startTime = System.currentTimeMillis();

        //runs when the window is closed, used to stop the thread
        WindowListener exitListener = new WindowAdapter()
        {
            @Override
            public void windowClosing(WindowEvent e)
            {
                gameClosed = true;
                System.exit(0);
            }
        };

        this.addWindowListener(exitListener);
        player.setFalling(true);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        lblUserMessage = new javax.swing.JLabel();
        lblCurrency2 = new javax.swing.JLabel();
        lblCurrency1 = new javax.swing.JLabel();
        lblCurrency = new javax.swing.JLabel();
        platformCollide = new javax.swing.JLabel();
        platformCollide1 = new javax.swing.JLabel();
        platformCollide2 = new javax.swing.JLabel();
        platformCollide3 = new javax.swing.JLabel();
        ground = new javax.swing.JLabel();
        sprite = new javax.swing.JLabel();
        obstacleCollide = new javax.swing.JLabel();
        obstacleCollide1 = new javax.swing.JLabel();
        obstacleCollide2 = new javax.swing.JLabel();
        obstacleCollide3 = new javax.swing.JLabel();
        obstacleCollide4 = new javax.swing.JLabel();
        platform = new javax.swing.JLabel();
        platform2 = new javax.swing.JLabel();
        platform3 = new javax.swing.JLabel();
        foreground2 = new javax.swing.JLabel();
        foreground1 = new javax.swing.JLabel();
        foreground = new javax.swing.JLabel();
        finish = new javax.swing.JLabel();
        chest = new javax.swing.JLabel();
        platform1 = new javax.swing.JLabel();
        obstacle = new javax.swing.JLabel();
        obstacle1 = new javax.swing.JLabel();
        obstacle2 = new javax.swing.JLabel();
        obstacle3 = new javax.swing.JLabel();
        obstacle4 = new javax.swing.JLabel();
        background = new javax.swing.JLabel();
        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Game Screen");
        getContentPane().setLayout(null);

        lblUserMessage.setFont(new java.awt.Font("Consolas", 0, 36)); // NOI18N
        lblUserMessage.setForeground(new java.awt.Color(102, 102, 102));
        lblUserMessage.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        getContentPane().add(lblUserMessage);
        lblUserMessage.setBounds(370, 30, 1220, 70);

        lblCurrency2.setFont(new java.awt.Font("Consolas", 0, 36)); // NOI18N
        lblCurrency2.setForeground(new java.awt.Color(102, 102, 102));
        lblCurrency2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblCurrency2.setText("Level 1:");
        getContentPane().add(lblCurrency2);
        lblCurrency2.setBounds(-170, 10, 560, 50);

        lblCurrency1.setFont(new java.awt.Font("Consolas", 0, 36)); // NOI18N
        lblCurrency1.setForeground(new java.awt.Color(153, 153, 153));
        lblCurrency1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblCurrency1.setText("Get Chests for 50 Currency");
        getContentPane().add(lblCurrency1);
        lblCurrency1.setBounds(20, 770, 560, 50);

        lblCurrency.setFont(new java.awt.Font("Consolas", 0, 36)); // NOI18N
        lblCurrency.setForeground(new java.awt.Color(102, 102, 102));
        lblCurrency.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblCurrency.setText("Currency: 0");
        getContentPane().add(lblCurrency);
        lblCurrency.setBounds(1610, 30, 270, 40);
        getContentPane().add(platformCollide);
        platformCollide.setBounds(800, 790, 430, 10);
        getContentPane().add(platformCollide1);
        platformCollide1.setBounds(2510, 740, 430, 10);
        getContentPane().add(platformCollide2);
        platformCollide2.setBounds(1850, 420, 430, 10);
        getContentPane().add(platformCollide3);
        platformCollide3.setBounds(1320, 610, 430, 10);
        getContentPane().add(ground);
        ground.setBounds(10, 990, 3880, 50);

        sprite.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/sprite1.PNG"))); // NOI18N
        getContentPane().add(sprite);
        sprite.setBounds(70, 870, 130, 130);
        getContentPane().add(obstacleCollide);
        obstacleCollide.setBounds(1800, 900, 60, 50);
        getContentPane().add(obstacleCollide1);
        obstacleCollide1.setBounds(1280, 910, 60, 50);
        getContentPane().add(obstacleCollide2);
        obstacleCollide2.setBounds(2010, 340, 60, 50);
        getContentPane().add(obstacleCollide3);
        obstacleCollide3.setBounds(990, 710, 60, 50);
        getContentPane().add(obstacleCollide4);
        obstacleCollide4.setBounds(810, 910, 60, 50);

        platform.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/platform.PNG"))); // NOI18N
        getContentPane().add(platform);
        platform.setBounds(1810, 390, 500, 110);

        platform2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/platform.PNG"))); // NOI18N
        getContentPane().add(platform2);
        platform2.setBounds(1280, 580, 500, 110);

        platform3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/platform.PNG"))); // NOI18N
        getContentPane().add(platform3);
        platform3.setBounds(2470, 710, 500, 110);

        foreground2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/ground.png"))); // NOI18N
        getContentPane().add(foreground2);
        foreground2.setBounds(3840, -20, 1930, 1080);

        foreground1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/ground.png"))); // NOI18N
        getContentPane().add(foreground1);
        foreground1.setBounds(1920, -20, 1930, 1080);

        foreground.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/ground.png"))); // NOI18N
        getContentPane().add(foreground);
        foreground.setBounds(0, -10, 1960, 1080);

        finish.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/finish.png"))); // NOI18N
        getContentPane().add(finish);
        finish.setBounds(3590, 480, 100, 500);

        chest.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/Images/
chest.png"))); // NOI18N
        getContentPane().add(chest);
        chest.setBounds(1470, 490, 140, 120);

        platform1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/platform.PNG"))); // NOI18N
        getContentPane().add(platform1);
        platform1.setBounds(760, 760, 500, 110);

        obstacle.setBackground(new java.awt.Color(0, 0, 0));
        obstacle.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw1.PNG"))); // NOI18N
```

```
        getContentPane().add(obstacle);
        obstacle.setBounds(780, 860, 120, 140);

        obstacle1.setBackground(new java.awt.Color(0, 0, 0));
        obstacle1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw1.PNG"))); // NOI18N
        getContentPane().add(obstacle1);
        obstacle1.setBounds(1250, 870, 120, 140);

        obstacle2.setBackground(new java.awt.Color(0, 0, 0));
        obstacle2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw1.PNG"))); // NOI18N
        getContentPane().add(obstacle2);
        obstacle2.setBounds(1980, 300, 120, 140);

        obstacle3.setBackground(new java.awt.Color(0, 0, 0));
        obstacle3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw1.PNG"))); // NOI18N
        getContentPane().add(obstacle3);
        obstacle3.setBounds(1770, 860, 120, 140);

        obstacle4.setBackground(new java.awt.Color(0, 0, 0));
        obstacle4.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw1.PNG"))); // NOI18N
        getContentPane().add(obstacle4);
        obstacle4.setBounds(960, 670, 120, 140);

        background.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/level1Background.PNG"))); // NOI18N
        getContentPane().add(background);
        background.setBounds(-10, -150, 5820, 1370);

        setSize(new java.awt.Dimension(1927, 1103));
        setLocationRelativeTo(null);
    }// </editor-fold>

    /**
     * @param args the command line arguments
     */
    @Override
    public void run()
    {
        initialPosX = foreground.getLocation().getX();
        long time = System.currentTimeMillis();

        CollisionDetection c = new CollisionDetection();

        int e = 0, f = 0, y = 0;

        //first location of the character used to control the chracter movement
        double referenceLocation = sprite.getLocation().getX();
        Animations animation = new Animations();
        while (!gameClosed)
        {
            if (System.currentTimeMillis() - time > 3)
            {
                //loop runs approx 30 times per second

                //character move
                if (foreground.getLocation().getX() == initialPosX)
                {
                    if (player.getDirection() == 5)
                    {
                        player.move();
                        sprite.setLocation(player.getX(), player.getY());
                    } else if (player.getDirection() == -5)
                    {
                        player.move();
                        sprite.setLocation(player.getX(), player.getY());
                    }
                }

                //world move
                if (sprite.getLocation().getX() > referenceLocation)
                {
                    //refreshes locations every game tick
                    for (int i = 0; i < 24; i++)
                    {
                        refreshLocations(labelsArr(i));

                    }
                }

                //keeps player jumping and handles collisions with the floor
                if (player.isJumping() || player.isFalling())
                {
                    player.moveVert();

                    //grounds
                    if (c.checkCollision(sprite, ground) || c.checkCollision(sprite, 
platformCollide1)
                            || c.checkCollision(sprite, platformCollide) || 
c.checkCollision(sprite, platformCollide2)
                            || c.checkCollision(sprite, platformCollide3))
                    {
                        player.setFalling(false);
                    }

                    sprite.setLocation(player.getX(), player.getY());
                }

                //grounds
                if (c.checkCollision(sprite, ground) == false && c.checkCollision(sprite,
platformCollide) == false
                        && c.checkCollision(sprite, platformCollide1) == false &&
c.checkCollision(sprite, platformCollide2) == false
                        && c.checkCollision(sprite, platformCollide3) == false)
                {
                    player.setFalling(true);
                }

                //obstacles
                if (c.checkCollision(sprite, obstacleCollide1) ||
c.checkCollision(sprite, obstacleCollide)
                        || c.checkCollision(sprite, obstacleCollide2) ||
c.checkCollision(sprite, obstacleCollide3)
                        || c.checkCollision(sprite, obstacleCollide4))
                {
                    lblUserMessage.setText("You Died");

                    //thread sleeps for 1 second
                    try
                    {
                        //amount of time in milliseconds
                        Thread.sleep(1000);
                    } catch (InterruptedException ex)
                    {
                    }

                    //players positions is reset to beginning
                    player.setX(800);
                    player.setY(500);
                    lastPosX = foreground.getLocation().getX();

                    //all objects in level are reset to their initial position
                    for (int i = 0; i < 24; i++)
                    {
                        resetLocations(labelsArr(i));
                    }
                }

                //chest
                if (c.checkCollision(sprite, chest) && chestCollided == false)
                {
                    chest.setVisible(false);
                    currency += 50;
                    lblCurrency.setText("Currency: " + currency);
                    chestCollided = true;
                }

                //finish
                if (c.checkCollision(sprite, finish))
                {
                    finishTime = System.currentTimeMillis();
                    System.out.println((finishTime - startTime)/1000);
                    UserArray u = new UserArray();
                    u.editTimes(0, 0, "" + (finishTime - startTime)/1000);
                    lblUserMessage.setText("Level 1 Complete");
                    currency += 100;
                    lblCurrency.setText("Currency: " + currency);
                    gameClosed = true;
                    new Level1Screen().dispose();

                }

                //animatates saws every game tick
                for (int i = 0; i < 5; i++)
                {
                    animateSaws(sawArr(i), animation.sawImageArr(f));
                }

                //animates character every 6 game ticks
                if (y == 0)
                {
                    //uses left image set
                    if (player.getDirection() == -5 && !player.isJumping() && !
player.isFalling())
                    {
                        sprite.setIcon(animation.playerLeftImageArr(e));
                    }

                    //uses right image set
                    if (player.getDirection() == 5 && !player.isJumping() && !
player.isFalling())
                    {
                        sprite.setIcon(animation.playerRightImageArr(e));
                    }
                }

                //controls for animations
                f++;
                if (f == 3)
                {
                    f = 0;
                }

                e++;
                if (e == 6)
                {
                    e = 0;
                }

                y++;
                if (y == 7)
                {
                    y = 0;
                }

                //gets the current milli time of system
                time = System.currentTimeMillis();
            }

        }

    }

    public class MyKeyListener implements KeyListener
    {

        @Override
        public void keyTyped(KeyEvent e)
        {

        }

        @Override
        public void keyPressed(KeyEvent e)
        {
            lblCurrency1.setText("");
            //Left
            if (e.getKeyCode() == KeyEvent.VK_A && player.getDirection() != -5 && !
gameClosed)
            {
                lblUserMessage.setText("");
                sprite.setIcon(new javax.swing.ImageIcon(getClass().getResource("/
anderson/Images/sprite1R.png")));
                player.setRight();
            }

            //Right
            if (e.getKeyCode() == KeyEvent.VK_D && player.getDirection() != 5 && !
gameClosed)
            {
                lblUserMessage.setText("");
                sprite.setIcon(new javax.swing.ImageIcon(getClass().getResource("/
anderson/Images/sprite1.png")));
                player.setLeft();
            }

            //Jump
            if (e.getKeyCode() == KeyEvent.VK_SPACE && !player.isJumping() && !
player.isFalling())
            {
                player.setJumping(true);
                Timer jumpTimer = new Timer("Jumping");

                TimerTask stopJump = new TimerTask()
                {
                    public void run()
                    {
                        player.setJumping(false);
                        player.setFalling(true);

                    }
                };
                jumpTimer.schedule(stopJump, 350);
            }

            //game is closed
            if (gameClosed)
            {
                new LevelSelectScreen().setVisible(true);
                dispose();
            }

        }

        @Override
        public void keyReleased(KeyEvent e)
        {
            if (e.getKeyCode() == KeyEvent.VK_A && player.getDirection() == -5)
            {
                player.stop();
            }
            if (e.getKeyCode() == KeyEvent.VK_D && player.getDirection() == 5)
            {
                player.stop();
            }
        }
    }

    //refreshes locations of all labels on the screen
    public void refreshLocations(JLabel x)
    {
        int d = -1 * player.getDirection();

        x.setLocation((int) x.getLocation().getX() + d, (int) x.getLocation().getY());
    }

    //restets locations of all labels on the screen
    public void resetLocations(JLabel x)
    {
        int i = (int) (initialPosX - lastPosX);
        x.setLocation((int) x.getLocation().getX() + i, (int) x.getLocation().getY());
    }

    //array of all labels on the screen
    public JLabel labelsArr(int i)
    {
        JLabel[] labels = new JLabel[]
        {
            obstacle, obstacle1, obstacle2, obstacle3, obstacle4, obstacleCollide, 
obstacleCollide1, obstacleCollide2, obstacleCollide3, obstacleCollide4, 
            platformCollide1, platformCollide2, platformCollide3, platform, platform1, 
platform2, platform3, 
            foreground1, ground, platformCollide, foreground, chest, finish, foreground2
        };
        return labels[i];
    }

    //array of all saws on screen
    public JLabel sawArr(int i)
    {
        JLabel[] labels = new JLabel[]
        {
            obstacle, obstacle1, obstacle2, obstacle3, obstacle4
        };
        return labels[i];
    }

    //sets icon of every saw on the screen according to the parameters
    public void animateSaws(JLabel x, ImageIcon i)
    {
        x.setIcon(i);
    }

    public static void main(String args[])
    {

        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code 
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default 
look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/
lookandfeel/plaf.html 
         */
        try
        {
            for (javax.swing.UIManager.LookAndFeelInfo info : 
javax.swing.UIManager.getInstalledLookAndFeels())
            {
                if ("Nimbus".equals(info.getName()))
                {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex)
        {
            java.util.logging.Logger.getLogger(Level1Screen.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex)
        {
            java.util.logging.Logger.getLogger(Level1Screen.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex)
        {
            java.util.logging.Logger.getLogger(Level1Screen.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex)
        {
            java.util.logging.Logger.getLogger(Level1Screen.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>
        //</editor-fold>
        //</editor-fold>
        //</editor-fold>
    }

    // Variables declaration - do not modify
    private javax.swing.JLabel background;
    private javax.swing.JLabel chest;
    private javax.swing.JLabel finish;
    private javax.swing.JLabel foreground;
    private javax.swing.JLabel foreground1;
    private javax.swing.JLabel foreground2;
    private javax.swing.JLabel ground;
    private javax.swing.JLabel lblCurrency;
    private javax.swing.JLabel lblCurrency1;
    private javax.swing.JLabel lblCurrency2;
    private javax.swing.JLabel lblUserMessage;
    private javax.swing.JLabel obstacle;
    private javax.swing.JLabel obstacle1;
    private javax.swing.JLabel obstacle2;
    private javax.swing.JLabel obstacle3;
    private javax.swing.JLabel obstacle4;
    private javax.swing.JLabel obstacleCollide;
    private javax.swing.JLabel obstacleCollide1;
    private javax.swing.JLabel obstacleCollide2;
    private javax.swing.JLabel obstacleCollide3;
    private javax.swing.JLabel obstacleCollide4;
    private javax.swing.JLabel platform;
    private javax.swing.JLabel platform1;
    private javax.swing.JLabel platform2;
    private javax.swing.JLabel platform3;
    private javax.swing.JLabel platformCollide;
    private javax.swing.JLabel platformCollide1;
    private javax.swing.JLabel platformCollide2;
    private javax.swing.JLabel platformCollide3;
    private javax.swing.JLabel sprite;
    // End of variables declaration
}
```

Level 2 Screen

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package anderson;

import java.awt.event.*;
import java.util.TimerTask;
import java.util.Timer;
import javax.swing.ImageIcon;
import javax.swing.JLabel;

/**
 *
 * @author Kian
 */
class Level2Screen extends javax.swing.JFrame implements Runnable
{

    Character player = new Character(540, 800, 0);
    boolean gameClosed = false;
    double initialPosX;
    double lastPosX;
    int currency = 0;
    boolean chestCollided = false;
    long startTime, finishTime;

    /**
     * Creates new form GameScreen
     */
    public Level2Screen()
    {
        initComponents();

        //setting up key listener
        KeyListener listener = new Level2Screen.MyKeyListener();
        addKeyListener(listener);
        setFocusable(true);

        //sets the initial location of the sprite
        sprite.setLocation(player.getX(), player.getY());
        startTime = System.currentTimeMillis();

        //runs when the window is closed, used to stop the thread
        WindowListener exitListener = new WindowAdapter()
        {
            @Override
            public void windowClosing(WindowEvent e)
            {
                gameClosed = true;
                System.exit(0);
            }
        };

        this.addWindowListener(exitListener);
        player.setFalling(true);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents()
    {

        userMessage = new javax.swing.JLabel();
        lblCurrency2 = new javax.swing.JLabel();
        lblCurrency1 = new javax.swing.JLabel();
        lblCurrency = new javax.swing.JLabel();
        platformCollide = new javax.swing.JLabel();
        platformCollide1 = new javax.swing.JLabel();
        platformCollide2 = new javax.swing.JLabel();
        platformCollide3 = new javax.swing.JLabel();
        ground = new javax.swing.JLabel();
        sprite = new javax.swing.JLabel();
        obstacleCollide = new javax.swing.JLabel();
        obstacleCollide1 = new javax.swing.JLabel();
        obstacleCollide2 = new javax.swing.JLabel();
        obstacleCollide3 = new javax.swing.JLabel();
        obstacleCollide4 = new javax.swing.JLabel();
        platform = new javax.swing.JLabel();
        platform2 = new javax.swing.JLabel();
        platform3 = new javax.swing.JLabel();
        foreground2 = new javax.swing.JLabel();
        foreground1 = new javax.swing.JLabel();
        foreground = new javax.swing.JLabel();
        finish = new javax.swing.JLabel();
        chest = new javax.swing.JLabel();
        platform1 = new javax.swing.JLabel();
        obstacle = new javax.swing.JLabel();
        obstacle1 = new javax.swing.JLabel();
        obstacle2 = new javax.swing.JLabel();
        obstacle3 = new javax.swing.JLabel();
        obstacle4 = new javax.swing.JLabel();
        background = new javax.swing.JLabel();
        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Game Screen");
        getContentPane().setLayout(null);

        userMessage.setFont(new java.awt.Font("Lucida Grande", 0, 36)); // NOI18N
        userMessage.setForeground(new java.awt.Color(255, 255, 255));
        userMessage.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        getContentPane().add(userMessage);
        userMessage.setBounds(370, 30, 1330, 70);

        lblCurrency2.setFont(new java.awt.Font("Consolas", 0, 36)); // NOI18N
        lblCurrency2.setForeground(new java.awt.Color(255, 255, 255));
        lblCurrency2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblCurrency2.setText("Level 2:");
        getContentPane().add(lblCurrency2);
        lblCurrency2.setBounds(-170, 10, 560, 50);

        lblCurrency1.setFont(new java.awt.Font("Consolas", 0, 36)); // NOI18N
        lblCurrency1.setForeground(new java.awt.Color(255, 255, 255));
        lblCurrency1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblCurrency1.setText("Get Chests for 50 Currency");
        getContentPane().add(lblCurrency1);
        lblCurrency1.setBounds(30, 720, 560, 50);

        lblCurrency.setFont(new java.awt.Font("Consolas", 0, 36)); // NOI18N
        lblCurrency.setForeground(new java.awt.Color(255, 255, 255));
        lblCurrency.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblCurrency.setText("Currency: 0");
        getContentPane().add(lblCurrency);
        lblCurrency.setBounds(1610, 20, 280, 30);
        getContentPane().add(platformCollide);
        platformCollide.setBounds(800, 790, 430, 10);
        getContentPane().add(platformCollide1);
        platformCollide1.setBounds(2510, 740, 430, 10);
        getContentPane().add(platformCollide2);
        platformCollide2.setBounds(1930, 590, 430, 10);
        getContentPane().add(platformCollide3);
        platformCollide3.setBounds(1570, 770, 430, 10);
        getContentPane().add(ground);
        ground.setBounds(10, 990, 3880, 50);

        sprite.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/sprite1.PNG"))); // NOI18N
        getContentPane().add(sprite);
        sprite.setBounds(70, 870, 130, 130);
        getContentPane().add(obstacleCollide);
        obstacleCollide.setBounds(1800, 900, 60, 50);
        getContentPane().add(obstacleCollide1);
        obstacleCollide1.setBounds(1010, 730, 60, 50);
        getContentPane().add(obstacleCollide2);
        obstacleCollide2.setBounds(2300, 540, 60, 50);
        getContentPane().add(obstacleCollide3);
        obstacleCollide3.setBounds(2490, 900, 60, 50);
        getContentPane().add(obstacleCollide4);
        obstacleCollide4.setBounds(1190, 910, 60, 50);

        platform.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/platform.PNG"))); // NOI18N
        getContentPane().add(platform);
        platform.setBounds(1900, 570, 500, 110);

        platform2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/platform.PNG"))); // NOI18N
        getContentPane().add(platform2);
        platform2.setBounds(1530, 740, 500, 110);

        platform3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/platform.PNG"))); // NOI18N
        getContentPane().add(platform3);
        platform3.setBounds(2470, 710, 500, 110);

        foreground2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/ground.png"))); // NOI18N
        getContentPane().add(foreground2);
        foreground2.setBounds(3840, -20, 1930, 1080);

        foreground1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/ground.png"))); // NOI18N
        getContentPane().add(foreground1);
        foreground1.setBounds(1920, -20, 1930, 1080);

        foreground.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/ground.png"))); // NOI18N
        getContentPane().add(foreground);
        foreground.setBounds(0, -10, 1960, 1080);

        finish.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/finish.png"))); // NOI18N
        getContentPane().add(finish);
        finish.setBounds(3590, 480, 100, 500);

        chest.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/Images/
chest.png"))); // NOI18N
        getContentPane().add(chest);
        chest.setBounds(1450, 880, 140, 120);

        platform1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/platform.PNG"))); // NOI18N
        getContentPane().add(platform1);
        platform1.setBounds(760, 760, 500, 110);

        obstacle.setBackground(new java.awt.Color(0, 0, 0));
        obstacle.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw1.PNG"))); // NOI18N
```

```
        getContentPane().add(obstacle);
        obstacle.setBounds(1160, 870, 120, 140);

        obstacle1.setBackground(new java.awt.Color(0, 0, 0));
        obstacle1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw1.PNG"))); // NOI18N
        getContentPane().add(obstacle1);
        obstacle1.setBounds(980, 690, 120, 140);

        obstacle2.setBackground(new java.awt.Color(0, 0, 0));
        obstacle2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw1.PNG"))); // NOI18N
        getContentPane().add(obstacle2);
        obstacle2.setBounds(2270, 490, 120, 140);

        obstacle3.setBackground(new java.awt.Color(0, 0, 0));
        obstacle3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw1.PNG"))); // NOI18N
        getContentPane().add(obstacle3);
        obstacle3.setBounds(1770, 860, 120, 140);

        obstacle4.setBackground(new java.awt.Color(0, 0, 0));
        obstacle4.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw1.PNG"))); // NOI18N
        getContentPane().add(obstacle4);
        obstacle4.setBounds(2460, 860, 120, 140);

        background.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/level2Background.PNG"))); // NOI18N
        getContentPane().add(background);
        background.setBounds(-10, -150, 5820, 1370);

        setSize(new java.awt.Dimension(1926, 1103));
        setLocationRelativeTo(null);
    }// </editor-fold>

    /**
     * @param args the command line arguments
     */
    @Override
    public void run()
    {
        initialPosX = foreground.getLocation().getX();
        long time = System.currentTimeMillis();
        CollisionDetection c = new CollisionDetection();

        int e = 0, f = 0, y = 0;

        //first location of the character used to control the chracter movement
        double referenceLocation = sprite.getLocation().getX();
        Animations animation = new Animations();
        while (!gameClosed)
        {
            if (System.currentTimeMillis() - time > 3)
            {
                //loop runs approx 30 times per second

                //character move
                if (foreground.getLocation().getX() == initialPosX)
                {
                    if (player.getDirection() == 5)
                    {
                        player.move();
                        sprite.setLocation(player.getX(), player.getY());
                    } else if (player.getDirection() == -5)
                    {
                        player.move();
                        sprite.setLocation(player.getX(), player.getY());
                    }
                }

                //world move
                if (sprite.getLocation().getX() > referenceLocation)
                {
                    //refreshes locations every game tick
                    for (int i = 0; i < 24; i++)
                    {
                        refreshLocations(labelsArr(i));
                    }
                }

                //keeps player jumping and handles collisions with the floor
                if (player.isJumping() || player.isFalling())
                {
                    player.moveVert();

                    //grounds
                    if (c.checkCollision(sprite, ground) || c.checkCollision(sprite, 
platformCollide1)
                            || c.checkCollision(sprite, platformCollide) || 
c.checkCollision(sprite, platformCollide2)
                            || c.checkCollision(sprite, platformCollide3))
                    {
                        player.setFalling(false);
                    }

                    sprite.setLocation(player.getX(), player.getY());
                }

                //grounds
                if (c.checkCollision(sprite, ground) == false && c.checkCollision(sprite, 
platformCollide) == false
```

```
                        && c.checkCollision(sprite, platformCollide1) == false &&
c.checkCollision(sprite, platformCollide2) == false
                        && c.checkCollision(sprite, platformCollide3) == false)
                {
                    player.setFalling(true);
                }

                //obstacles
                if (c.checkCollision(sprite, obstacleCollide1) ||
c.checkCollision(sprite, obstacleCollide)
                        || c.checkCollision(sprite, obstacleCollide2) ||
c.checkCollision(sprite, obstacleCollide3)
                        || c.checkCollision(sprite, obstacleCollide4))
                {

                    userMessage.setText("You Died");

                    //thread sleeps for 1 second
                    try
                    {
                        //amount of time in milliseconds
                        Thread.sleep(1000);
                    } catch (InterruptedException ex)
                    {
                    }

                    //players positions is reset to beginning
                    player.setX(800);
                    player.setY(500);
                    lastPosX = foreground.getLocation().getX();

                    //all objects in level are reset to their initial position
                    for (int i = 0; i < 24; i++)
                    {
                        resetLocations(labelsArr(i));
                    }
                }

                //chest
                if (c.checkCollision(sprite, chest) && chestCollided == false)
                {
                    chest.setVisible(false);
                    currency += 50;
                    lblCurrency.setText("Currency: " + currency);
                    chestCollided = true;
                }

                //finish
                if (c.checkCollision(sprite, finish))
                {
                    finishTime = System.currentTimeMillis();
                    UserArray u = new UserArray();
                    u.editTimes(0, 1, "" + (finishTime - startTime)/1000);
                    currency += 100;
                    lblCurrency.setText("Currency: " + currency);
                    gameClosed = true;
                    new Level2Screen().dispose();
                }

                //animatates saws every game tick
                for (int i = 0; i < 5; i++)
                {
                    animateSaws(sawArr(i), animation.sawImageArr(f));
                }

                //animates character every 6 game ticks
                if (y == 0)
                {
                    //uses left image set
                    if (player.getDirection() == -5 && !player.isJumping() && !
player.isFalling())
                    {
                        sprite.setIcon(animation.playerLeftImageArr(e));
                    }

                    //uses right image set
                    if (player.getDirection() == 5 && !player.isJumping() && !
player.isFalling())
                    {
                        sprite.setIcon(animation.playerRightImageArr(e));
                    }
                }

                //controls
                f++;
                if (f == 3)
                {
                    f = 0;
                }

                e++;
                if (e == 6)
                {
                    e = 0;
                }

                y++;
                if (y == 7)
                {
                    y = 0;
                }

                //gets the current milli time of system
                time = System.currentTimeMillis();
```

```
            }

        }

    }

    public class MyKeyListener implements KeyListener
    {

        @Override
        public void keyTyped(KeyEvent e)
        {

        }

        @Override
        public void keyPressed(KeyEvent e)
        {

            lblCurrency1.setText("");
            //Left
            if (e.getKeyCode() == KeyEvent.VK_A && player.getDirection() != -5 && !
gameClosed)
            {
                userMessage.setText("");
                sprite.setIcon(new javax.swing.ImageIcon(getClass().getResource("/
anderson/Images/sprite1R.png")));
                player.setRight();
            }

            //Right
            if (e.getKeyCode() == KeyEvent.VK_D && player.getDirection() != 5 && !
gameClosed)
            {
                userMessage.setText("");
                sprite.setIcon(new javax.swing.ImageIcon(getClass().getResource("/
anderson/Images/sprite1.png")));
                player.setLeft();
            }

            //Jump
            if (e.getKeyCode() == KeyEvent.VK_SPACE && !player.isJumping() && !
player.isFalling())
            {
                player.setJumping(true);
                Timer jumpTimer = new Timer("Jumping");

                TimerTask stopJump = new TimerTask()
                {
                    public void run()
                    {
                        player.setJumping(false);
                        player.setFalling(true);

                    }
                };
                jumpTimer.schedule(stopJump, 350);
            }

            //game is closed
            if (gameClosed)
            {
                new LevelSelectScreen().setVisible(true);
                dispose();
            }

        }

        @Override
        public void keyReleased(KeyEvent e)
        {
            if (e.getKeyCode() == KeyEvent.VK_A && player.getDirection() == -5)
            {
                player.stop();
            }
            if (e.getKeyCode() == KeyEvent.VK_D && player.getDirection() == 5)
            {
                player.stop();
            }

        }
    }

    //refreshes locations of all labels on the screen
    public void refreshLocations(JLabel x)
    {
        int d = -1 * player.getDirection();

        x.setLocation((int) x.getLocation().getX() + d, (int) x.getLocation().getY());
    }

    //restets locations of all labels on the screen
    public void resetLocations(JLabel x)
    {
        int i = (int) (initialPosX - lastPosX);
        x.setLocation((int) x.getLocation().getX() + i, (int) x.getLocation().getY());
    }

    //array of all labels on the screen
    public JLabel labelsArr(int i)
    {
        JLabel[] labels = new JLabel[]
        {
            obstacle, obstacle1, obstacle2, obstacle3, obstacle4, obstacleCollide,
obstacleCollide1, obstacleCollide2, obstacleCollide3, obstacleCollide4,
            platformCollide1, platformCollide2, platformCollide3, platform, platform1,
platform2, platform3,
            foreground1, ground, platformCollide, foreground, chest, finish, foreground2
        };
        return labels[i];
    }

    //array of all saws on screen
    public JLabel sawArr(int i)
    {
        JLabel[] labels = new JLabel[]
        {
            obstacle, obstacle1, obstacle2, obstacle3, obstacle4
        };
        return labels[i];
    }

    //sets icon of every saw on the screen according to the parameters
    public void animateSaws(JLabel x, ImageIcon i)
    {
        x.setIcon(i);
    }

    public static void main(String args[])
    {

        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default
look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/
lookandfeel/plaf.html
         */
        try
        {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels())
            {
                if ("Nimbus".equals(info.getName()))
                {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;

                }
            }
        } catch (ClassNotFoundException ex)
        {
            java.util.logging.Logger.getLogger(Level2Screen.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex)
        {
            java.util.logging.Logger.getLogger(Level2Screen.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex)
        {
            java.util.logging.Logger.getLogger(Level2Screen.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex)
        {
            java.util.logging.Logger.getLogger(Level2Screen.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>
        //</editor-fold>
        //</editor-fold>
        //</editor-fold>
    }

    // Variables declaration - do not modify
    private javax.swing.JLabel background;
    private javax.swing.JLabel chest;
    private javax.swing.JLabel finish;
    private javax.swing.JLabel foreground;
    private javax.swing.JLabel foreground1;
    private javax.swing.JLabel foreground2;
    private javax.swing.JLabel ground;
    private javax.swing.JLabel lblCurrency;
    private javax.swing.JLabel lblCurrency1;
    private javax.swing.JLabel lblCurrency2;
    private javax.swing.JLabel obstacle;
    private javax.swing.JLabel obstacle1;
    private javax.swing.JLabel obstacle2;
    private javax.swing.JLabel obstacle3;
    private javax.swing.JLabel obstacle4;
    private javax.swing.JLabel obstacleCollide;
    private javax.swing.JLabel obstacleCollide1;
    private javax.swing.JLabel obstacleCollide2;
    private javax.swing.JLabel obstacleCollide3;
    private javax.swing.JLabel obstacleCollide4;
    private javax.swing.JLabel platform;
    private javax.swing.JLabel platform1;
    private javax.swing.JLabel platform2;
    private javax.swing.JLabel platform3;
    private javax.swing.JLabel platformCollide;
    private javax.swing.JLabel platformCollide1;
    private javax.swing.JLabel platformCollide2;
    private javax.swing.JLabel platformCollide3;
    private javax.swing.JLabel sprite;
    private javax.swing.JLabel userMessage;
    // End of variables declaration
}
```

Level 3 Screen

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package anderson;

import java.awt.event.*;
import java.util.TimerTask;
import java.util.Timer;
import javax.swing.ImageIcon;
import javax.swing.JLabel;

/**
 *
 * @author Kian
 */
class Level3Screen extends javax.swing.JFrame implements Runnable
{

    Character player = new Character(540, 800, 0);
    boolean gameClosed = false;
    double initialPosX;
    double lastPosX;
    int currency = 0;
    boolean chestCollided = false;
    long startTime, finishTime;

    /**
     * Creates new form GameScreen
     */
    public Level3Screen()
    {
        initComponents();

        //setting up key listener
        KeyListener listener = new Level3Screen.MyKeyListener();
        addKeyListener(listener);
        setFocusable(true);

        //sets the initial location of the sprite
        sprite.setLocation(player.getX(), player.getY());
        startTime = System.currentTimeMillis();

        //runs when the window is closed, used to stop the thread
        WindowListener exitListener = new WindowAdapter()
        {
            @Override
            public void windowClosing(WindowEvent e)
            {
                gameClosed = true;
                System.exit(0);

            }
        };

        this.addWindowListener(exitListener);
        player.setFalling(true);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents()
    {

        userMessage = new javax.swing.JLabel();
        lblCurrency2 = new javax.swing.JLabel();
        lblCurrency1 = new javax.swing.JLabel();
        lblCurrency = new javax.swing.JLabel();
        platformCollide = new javax.swing.JLabel();
        platformCollide1 = new javax.swing.JLabel();
        platformCollide2 = new javax.swing.JLabel();
        platformCollide3 = new javax.swing.JLabel();
        ground = new javax.swing.JLabel();
        sprite = new javax.swing.JLabel();
        obstacleCollide = new javax.swing.JLabel();
        obstacleCollide1 = new javax.swing.JLabel();
        obstacleCollide2 = new javax.swing.JLabel();
        obstacleCollide3 = new javax.swing.JLabel();
        obstacleCollide4 = new javax.swing.JLabel();
        platform = new javax.swing.JLabel();
        platform2 = new javax.swing.JLabel();
        platform3 = new javax.swing.JLabel();
        foreground2 = new javax.swing.JLabel();
        foreground1 = new javax.swing.JLabel();
        foreground = new javax.swing.JLabel();
        finish = new javax.swing.JLabel();
        chest = new javax.swing.JLabel();
        platform1 = new javax.swing.JLabel();
        obstacle = new javax.swing.JLabel();
        obstacle1 = new javax.swing.JLabel();
        obstacle2 = new javax.swing.JLabel();
        obstacle3 = new javax.swing.JLabel();
        obstacle4 = new javax.swing.JLabel();
        background = new javax.swing.JLabel();
        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Game Screen");
        getContentPane().setLayout(null);

        userMessage.setFont(new java.awt.Font("Consolas", 0, 36)); // NOI18N
        userMessage.setForeground(new java.awt.Color(255, 255, 255));
        userMessage.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        getContentPane().add(userMessage);
        userMessage.setBounds(370, 30, 1160, 70);

        lblCurrency2.setFont(new java.awt.Font("Consolas", 0, 36)); // NOI18N
        lblCurrency2.setForeground(new java.awt.Color(255, 255, 255));
        lblCurrency2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblCurrency2.setText("Level 3:");
        getContentPane().add(lblCurrency2);
        lblCurrency2.setBounds(-180, 10, 560, 50);

        lblCurrency1.setFont(new java.awt.Font("Consolas", 0, 36)); // NOI18N
        lblCurrency1.setForeground(new java.awt.Color(255, 255, 255));
        lblCurrency1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblCurrency1.setText("Get Chests for 50 Currency");
        getContentPane().add(lblCurrency1);
        lblCurrency1.setBounds(30, 720, 560, 50);

        lblCurrency.setFont(new java.awt.Font("Consolas", 0, 36)); // NOI18N
        lblCurrency.setForeground(new java.awt.Color(255, 255, 255));
        lblCurrency.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblCurrency.setText("Currency: 0");
        getContentPane().add(lblCurrency);
        lblCurrency.setBounds(1640, 30, 240, 30);
        getContentPane().add(platformCollide);
        platformCollide.setBounds(800, 790, 430, 10);
        getContentPane().add(platformCollide1);
        platformCollide1.setBounds(2510, 740, 430, 10);
        getContentPane().add(platformCollide2);
        platformCollide2.setBounds(1930, 590, 430, 10);
        getContentPane().add(platformCollide3);
        platformCollide3.setBounds(1320, 610, 430, 10);
        getContentPane().add(ground);
        ground.setBounds(10, 990, 3880, 50);

        sprite.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/sprite1.PNG"))); // NOI18N
        getContentPane().add(sprite);
        sprite.setBounds(70, 870, 130, 130);
        getContentPane().add(obstacleCollide);
        obstacleCollide.setBounds(1800, 900, 60, 50);
        getContentPane().add(obstacleCollide1);
        obstacleCollide1.setBounds(1010, 730, 60, 50);
        getContentPane().add(obstacleCollide2);
        obstacleCollide2.setBounds(2300, 540, 60, 50);
        getContentPane().add(obstacleCollide3);
        obstacleCollide3.setBounds(1940, 550, 60, 50);
        getContentPane().add(obstacleCollide4);
        obstacleCollide4.setBounds(810, 910, 60, 50);

        platform.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/platform.PNG"))); // NOI18N
        getContentPane().add(platform);
        platform.setBounds(1900, 570, 500, 110);

        platform2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/platform.PNG"))); // NOI18N
        getContentPane().add(platform2);
        platform2.setBounds(1280, 580, 500, 110);

        platform3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/platform.PNG"))); // NOI18N
        getContentPane().add(platform3);
        platform3.setBounds(2470, 710, 500, 110);

        foreground2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/ground.png"))); // NOI18N
        getContentPane().add(foreground2);
        foreground2.setBounds(3840, -20, 1930, 1080);

        foreground1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/ground.png"))); // NOI18N
        getContentPane().add(foreground1);
        foreground1.setBounds(1920, -20, 1930, 1080);

        foreground.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/ground.png"))); // NOI18N
        getContentPane().add(foreground);
        foreground.setBounds(0, 0, 1960, 1080);

        finish.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/finish.png"))); // NOI18N
        getContentPane().add(finish);
        finish.setBounds(3590, 480, 100, 500);

        chest.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/Images/
chest.png"))); // NOI18N
        getContentPane().add(chest);
        chest.setBounds(1370, 860, 140, 120);

        platform1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/platform.PNG"))); // NOI18N
        getContentPane().add(platform1);
        platform1.setBounds(760, 760, 500, 110);

        obstacle.setBackground(new java.awt.Color(0, 0, 0));
        obstacle.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw1.PNG"))); // NOI18N
        getContentPane().add(obstacle);
        obstacle.setBounds(780, 860, 120, 140);

        obstacle1.setBackground(new java.awt.Color(0, 0, 0));
        obstacle1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw1.PNG"))); // NOI18N
        getContentPane().add(obstacle1);
        obstacle1.setBounds(980, 690, 120, 140);

        obstacle2.setBackground(new java.awt.Color(0, 0, 0));
        obstacle2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw1.PNG"))); // NOI18N
        getContentPane().add(obstacle2);
        obstacle2.setBounds(2270, 490, 120, 140);

        obstacle3.setBackground(new java.awt.Color(0, 0, 0));
        obstacle3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw1.PNG"))); // NOI18N
        getContentPane().add(obstacle3);
        obstacle3.setBounds(1770, 860, 120, 140);

        obstacle4.setBackground(new java.awt.Color(0, 0, 0));
        obstacle4.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/saw1.PNG"))); // NOI18N
        getContentPane().add(obstacle4);
        obstacle4.setBounds(1910, 500, 120, 140);

        background.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/level3Background.PNG"))); // NOI18N
        getContentPane().add(background);
        background.setBounds(-10, -150, 5820, 1370);

        setSize(new java.awt.Dimension(1924, 1103));
        setLocationRelativeTo(null);
    }// </editor-fold>

    /**
     * @param args the command line arguments
     */
    @Override
    public void run()
    {
        initialPosX = foreground.getLocation().getX();
        long time = System.currentTimeMillis();
        CollisionDetection c = new CollisionDetection();

        int e = 0, f = 0, y = 0;

        //first location of the character used to control the chracter movement
        double referenceLocation = sprite.getLocation().getX();
        Animations animation = new Animations();

        while (!gameClosed)
        {
            if (System.currentTimeMillis() - time > 3)
            {
                //loop runs approx 30 times per second

                //character move
                if (foreground.getLocation().getX() == initialPosX)
                {
                    if (player.getDirection() == 5)
                    {
                        player.move();
                        sprite.setLocation(player.getX(), player.getY());
                    } else if (player.getDirection() == -5)
                    {
                        player.move();
                        sprite.setLocation(player.getX(), player.getY());
                    }
                }

                //world move
                if (sprite.getLocation().getX() > referenceLocation)
                {
                    //refreshes locations every game tick
                    for (int i = 0; i < 24; i++)
                    {
                        refreshLocations(labelsArr(i));
                    }
                }

                //keeps player jumping and handles collisions with the floor
                if (player.isJumping() || player.isFalling())
                {
                    player.moveVert();
                    //grounds
                    if (c.checkCollision(sprite, ground) || c.checkCollision(sprite, 
platformCollide1)
                            || c.checkCollision(sprite, platformCollide) || 
c.checkCollision(sprite, platformCollide2)
                            || c.checkCollision(sprite, platformCollide3))
                    {
                        player.setFalling(false);
                    }

                    sprite.setLocation(player.getX(), player.getY());
                }

                //grounds
                if (c.checkCollision(sprite, ground) == false && c.checkCollision(sprite, 
platformCollide) == false
```

```
                        && c.checkCollision(sprite, platformCollide1) == false &&
c.checkCollision(sprite, platformCollide2) == false
                        && c.checkCollision(sprite, platformCollide3) == false)
                {
                    player.setFalling(true);
                }

                //obstacles
                if (c.checkCollision(sprite, obstacleCollide1) ||
c.checkCollision(sprite, obstacleCollide)
                        || c.checkCollision(sprite, obstacleCollide2) ||
c.checkCollision(sprite, obstacleCollide3)
                        || c.checkCollision(sprite, obstacleCollide4))
                {
                    userMessage.setText("You Died");

                    //thread sleeps for 1 second
                    try
                    {
                        //amount of time in milliseconds
                        Thread.sleep(1000);
                    } catch (InterruptedException ex)
                    {
                    }

                    //players positions is reset to beginning
                    player.setX(800);
                    player.setY(500);
                    lastPosX = foreground.getLocation().getX();

                    //all objects in level are reset to their initial position
                    for (int i = 0; i < 24; i++)
                    {
                        resetLocations(labelsArr(i));
                    }
                }

                //chest
                if (c.checkCollision(sprite, chest) && chestCollided == false)
                {
                    chest.setVisible(false);
                    currency += 50;
                    lblCurrency.setText("Currency: " + currency);
                    chestCollided = true;
                }

                //finish
                if (c.checkCollision(sprite, finish))
                {
                    finishTime = System.currentTimeMillis();
                    UserArray u = new UserArray();
                    u.editTimes(0, 2, "" + (finishTime - startTime) / 1000);
                    userMessage.setText("Level 3 Complete");
                    currency += 100;
                    lblCurrency.setText("Currency: " + currency);
                    gameClosed = true;
                    new Level3Screen().dispose();
                }

                //animatates saws every game tick
                for (int i = 0; i < 5; i++)
                {
                    animateSaws(sawArr(i), animation.sawImageArr(f));
                }

                //animates character every 6 game ticks
                if (y == 0)
                {
                    //uses left image set
                    if (player.getDirection() == -5 && !player.isJumping() && !
player.isFalling())
                    {
                        sprite.setIcon(animation.playerLeftImageArr(e));
                    }

                    //uses right image set
                    if (player.getDirection() == 5 && !player.isJumping() && !
player.isFalling())
                    {
                        sprite.setIcon(animation.playerRightImageArr(e));
                    }
                }

                //controls
                f++;
                if (f == 3)
                {
                    f = 0;
                }

                e++;
                if (e == 6)
                {
                    e = 0;
                }

                y++;
                if (y == 7)
                {
                    y = 0;
                }
                //gets the current milli time of system
                time = System.currentTimeMillis();
            }

        }

    }

    public class MyKeyListener implements KeyListener
    {

        @Override
        public void keyTyped(KeyEvent e)
        {

        }

        @Override
        public void keyPressed(KeyEvent e)
        {
            lblCurrency1.setText("");
            //Left
            if (e.getKeyCode() == KeyEvent.VK_A && player.getDirection() != -5 && !
gameClosed)
            {
                userMessage.setText("");
                sprite.setIcon(new javax.swing.ImageIcon(getClass().getResource("/
anderson/Images/sprite1R.png")));
                player.setRight();
            }

            //Right
            if (e.getKeyCode() == KeyEvent.VK_D && player.getDirection() != 5 && !
gameClosed)
            {
                userMessage.setText("");
                sprite.setIcon(new javax.swing.ImageIcon(getClass().getResource("/
anderson/Images/sprite1.png")));
                player.setLeft();
            }

            //Jump
            if (e.getKeyCode() == KeyEvent.VK_SPACE && !player.isJumping() && !
player.isFalling())
            {
                player.setJumping(true);
                Timer jumpTimer = new Timer("Jumping");

                TimerTask stopJump = new TimerTask()
                {
                    public void run()
                    {
                        player.setJumping(false);
                        player.setFalling(true);

                    }
                };
                jumpTimer.schedule(stopJump, 350);
            }

            //game is closed
            if (gameClosed)
            {
                new LevelSelectScreen().setVisible(true);
                dispose();
            }

        }

        @Override
        public void keyReleased(KeyEvent e)
        {
            if (e.getKeyCode() == KeyEvent.VK_A && player.getDirection() == -5)
            {
                player.stop();
            }
            if (e.getKeyCode() == KeyEvent.VK_D && player.getDirection() == 5)
            {
                player.stop();
            }

        }
    }

    //refreshes locations of all labels on the screen
    public void refreshLocations(JLabel x)
    {
        int d = -1 * player.getDirection();

        x.setLocation((int) x.getLocation().getX() + d, (int) x.getLocation().getY());
    }

    //restets locations of all labels on the screen
    public void resetLocations(JLabel x)
    {
        int i = (int) (initialPosX - lastPosX);
        x.setLocation((int) x.getLocation().getX() + i, (int) x.getLocation().getY());
    }

    //array of all labels on the screen
    public JLabel labelsArr(int i)
    {
        JLabel[] labels = new JLabel[]
        {
            obstacle, obstacle1, obstacle2, obstacle3, obstacle4, obstacleCollide,
obstacleCollide1, obstacleCollide2, obstacleCollide3, obstacleCollide4,
            platformCollide1, platformCollide2, platformCollide3, platform, platform1,
platform2, platform3,
            foreground1, ground, platformCollide, foreground, chest, finish, foreground2
        };
        return labels[i];
    }

    //array of all saws on screen
    public JLabel sawArr(int i)
    {
        JLabel[] labels = new JLabel[]
        {
            obstacle, obstacle1, obstacle2, obstacle3, obstacle4
        };
        return labels[i];
    }

    //sets icon of every saw on the screen according to the parameters
    public void animateSaws(JLabel x, ImageIcon i)
    {
        x.setIcon(i);
    }

    public static void main(String args[])
    {

        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default
look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/
lookandfeel/plaf.html
         */
        try
        {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels())
            {
                if ("Nimbus".equals(info.getName()))
                {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;

                }
            }
        } catch (ClassNotFoundException ex)
        {
            java.util.logging.Logger.getLogger(Level3Screen.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex)
        {
            java.util.logging.Logger.getLogger(Level3Screen.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex)
        {
            java.util.logging.Logger.getLogger(Level3Screen.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex)
        {
            java.util.logging.Logger.getLogger(Level3Screen.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>
        //</editor-fold>
    }

    // Variables declaration - do not modify
    private javax.swing.JLabel background;
    private javax.swing.JLabel chest;
    private javax.swing.JLabel finish;
    private javax.swing.JLabel foreground;
    private javax.swing.JLabel foreground1;
    private javax.swing.JLabel foreground2;
    private javax.swing.JLabel ground;
    private javax.swing.JLabel lblCurrency;
    private javax.swing.JLabel lblCurrency1;
    private javax.swing.JLabel lblCurrency2;
    private javax.swing.JLabel obstacle;
    private javax.swing.JLabel obstacle1;
    private javax.swing.JLabel obstacle2;
    private javax.swing.JLabel obstacle3;
    private javax.swing.JLabel obstacle4;
    private javax.swing.JLabel obstacleCollide;
    private javax.swing.JLabel obstacleCollide1;
    private javax.swing.JLabel obstacleCollide2;
    private javax.swing.JLabel obstacleCollide3;
    private javax.swing.JLabel obstacleCollide4;
    private javax.swing.JLabel platform;
    private javax.swing.JLabel platform1;
    private javax.swing.JLabel platform2;
    private javax.swing.JLabel platform3;
    private javax.swing.JLabel platformCollide;
    private javax.swing.JLabel platformCollide1;
    private javax.swing.JLabel platformCollide2;
    private javax.swing.JLabel platformCollide3;
    private javax.swing.JLabel sprite;
    private javax.swing.JLabel userMessage;
    // End of variables declaration
}
```

Level Select Screen

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package anderson;

/**
 *
 * @author Kian
 */
public class LevelSelectScreen extends javax.swing.JFrame
{

    UserArray user = new UserArray();

    /**
     * Creates new form LevelSelectScreen
     */
    public LevelSelectScreen()
    {
        initComponents();
        if (!user.getBestTimes(0, 1).equals("Not completed"))
        {

        }
        if (!user.getBestTimes(0, 2).equals("Not completed"))
        {

        }

    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        lblLevel1 = new javax.swing.JLabel();
        lblLevel2 = new javax.swing.JLabel();
        lblTitle = new javax.swing.JLabel();
        lblLevel3 = new javax.swing.JLabel();
        btnLvl3 = new javax.swing.JButton();
        btnLvl2 = new javax.swing.JButton();
        btnLvl1 = new javax.swing.JButton();
        btnClose = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Level Select");
        getContentPane().setLayout(null);

        jPanel1.setBackground(new java.awt.Color(0, 0, 0));

        lblLevel1.setFont(new java.awt.Font("Dialog", 1, 18)); // NOI18N
        lblLevel1.setForeground(new java.awt.Color(255, 255, 255));
        lblLevel1.setText("Level 1");

        lblLevel2.setFont(new java.awt.Font("Dialog", 1, 18)); // NOI18N
        lblLevel2.setForeground(new java.awt.Color(255, 255, 255));
        lblLevel2.setText("Level 2");

        lblTitle.setFont(new java.awt.Font("Consolas", 1, 48)); // NOI18N
        lblTitle.setForeground(new java.awt.Color(255, 255, 255));
        lblTitle.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblTitle.setText("Level Select");

        lblLevel3.setFont(new java.awt.Font("Dialog", 1, 18)); // NOI18N
        lblLevel3.setForeground(new java.awt.Color(255, 255, 255));
        lblLevel3.setText("Level 3");

        btnLvl3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/level3.png"))); // NOI18N
        btnLvl3.setActionCommand("3");
        btnLvl3.setBorder(null);
        btnLvl3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnLvlActionPerformed(evt);
            }
        });

        btnLvl2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/level2.png"))); // NOI18N
        btnLvl2.setToolTipText("");
        btnLvl2.setActionCommand("2");
        btnLvl2.setBorder(null);
        btnLvl2.setFocusCycleRoot(true);
        btnLvl2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnLvlActionPerformed(evt);
            }
        });

        btnLvl1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/level1.png"))); // NOI18N
        btnLvl1.setActionCommand("1");
        btnLvl1.setBorder(null);
        btnLvl1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnLvlActionPerformed(evt);
            }
        });

        btnClose.setBackground(new java.awt.Color(51, 51, 51));
        btnClose.setFont(new java.awt.Font("Consolas", 1, 14)); // NOI18N
        btnClose.setForeground(new java.awt.Color(255, 255, 255));
        btnClose.setText("Close");
        btnClose.setBorder(null);
        btnClose.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnCloseActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(209, 209, 209)
                .addComponent(btnLvl1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(btnLvl2)
                .addGap(227, 227, 227)
                .addComponent(btnLvl3)
                .addGap(351, 351, 351))
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(343, 343, 343)
                .addComponent(lblLevel1)
                .addGap(527, 527, 527)
                .addComponent(lblLevel2)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(lblLevel3)
                .addGap(484, 484, 484))
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(lblTitle, javax.swing.GroupLayout.PREFERRED_SIZE, 1898,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(156, Short.MAX_VALUE))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(btnClose, javax.swing.GroupLayout.PREFERRED_SIZE, 66,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(173, 173, 173))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(21, 21, 21)
                .addComponent(btnClose, javax.swing.GroupLayout.PREFERRED_SIZE, 29, 
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(lblTitle)
                .addGap(99, 99, 99)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                    .addComponent(btnLvl3)
                    .addComponent(btnLvl2)
                    .addComponent(btnLvl1))
                .addGap(18, 18, 18)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                    .addComponent(lblLevel1)
                    .addComponent(lblLevel2)
                    .addComponent(lblLevel3))
                .addContainerGap(584, Short.MAX_VALUE))
        );

        getContentPane().add(jPanel1);
        jPanel1.setBounds(0, 0, 2060, 1200);

        setSize(new java.awt.Dimension(1931, 1121));
        setLocationRelativeTo(null);
    }// </editor-fold>

    private void btnLvlActionPerformed(java.awt.event.ActionEvent evt)
    {
        String lvl = evt.getActionCommand();

        if (!user.getBestTimes(0, Integer.parseInt(lvl) - 1).equals("Not completed"))
        {
            switch (lvl)
            {
                case "1":
                    Level1Screen m1 = new Level1Screen();
                    Thread t1 = new Thread(m1);
                    t1.start();
                    m1.setVisible(true);
                    this.dispose();
                    break;

                case "2":
                        Level2Screen m2 = new Level2Screen();
                        Thread t2 = new Thread(m2);
                        t2.start();
                        m2.setVisible(true);
                        this.dispose();
                    break;

                case "3":
                        Level3Screen m3 = new Level3Screen();
                        Thread t3 = new Thread(m3);
                        t3.start();
                        m3.setVisible(true);
                        this.dispose();
                    break;
            }
        }


    }

    private void btnCloseActionPerformed(java.awt.event.ActionEvent evt)
    {
        new MainScreen().setVisible(true);
        dispose();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[])
    {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code 
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default 
look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/
lookandfeel/plaf.html
         */
        try
        {
            for (javax.swing.UIManager.LookAndFeelInfo info : 
javax.swing.UIManager.getInstalledLookAndFeels())
            {
                if ("Nimbus".equals(info.getName()))
                {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex)
        {

java.util.logging.Logger.getLogger(LevelSelectScreen.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        } catch (InstantiationException ex)
        {
```

```
java.util.logging.Logger.getLogger(LevelSelectScreen.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex)
        {

java.util.logging.Logger.getLogger(LevelSelectScreen.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex)
        {

java.util.logging.Logger.getLogger(LevelSelectScreen.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                new LevelSelectScreen().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton btnClose;
    private javax.swing.JButton btnLvl1;
    private javax.swing.JButton btnLvl2;
    private javax.swing.JButton btnLvl3;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JLabel lblLevel1;
    private javax.swing.JLabel lblLevel2;
    private javax.swing.JLabel lblLevel3;
    private javax.swing.JLabel lblTitle;
    // End of variables declaration
}
```

Login Screen

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package anderson;

import java.awt.Color;

public class LoginScreen extends javax.swing.JFrame
{

    public LoginScreen()
    {
        initComponents();

    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        lblLogin = new javax.swing.JLabel();
        jPanel1 = new javax.swing.JPanel();
        lblUsername1 = new javax.swing.JLabel();
        txtUserError = new javax.swing.JLabel();
        txtPassNew2 = new javax.swing.JPasswordField();
        lblConfirmPassword = new javax.swing.JLabel();
        txtPassExist = new javax.swing.JPasswordField();
        lblPassword = new javax.swing.JLabel();
        txtUsernameExist = new javax.swing.JTextField();
        lblExistUser = new javax.swing.JLabel();
        lblNewUser1 = new javax.swing.JLabel();
        btnSignUp = new javax.swing.JButton();
        btnLogin = new javax.swing.JButton();
        txtUsernameNew = new javax.swing.JPasswordField();
        txtPassNew1 = new javax.swing.JPasswordField();
        lblPassword2 = new javax.swing.JLabel();
        lblUsername2 = new javax.swing.JLabel();
        pnlBackground = new javax.swing.JScrollPane();
        txtArea = new javax.swing.JTextArea();
        lblNewUser = new javax.swing.JLabel();
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Login");
        getContentPane().setLayout(null);

        lblLogin.setFont(new java.awt.Font("Consolas", 1, 24)); // NOI18N
        lblLogin.setForeground(new java.awt.Color(255, 255, 255));
        lblLogin.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblLogin.setText("Login");
        getContentPane().add(lblLogin);
        lblLogin.setBounds(6, 16, 820, 30);

        jPanel1.setBackground(new java.awt.Color(25, 25, 25));

        lblUsername1.setFont(new java.awt.Font("Consolas", 1, 12)); // NOI18N
        lblUsername1.setForeground(new java.awt.Color(255, 255, 255));
        lblUsername1.setText("Username: ");

        txtUserError.setFont(new java.awt.Font("Lucida Grande", 0, 18)); // NOI18N
        txtUserError.setForeground(new java.awt.Color(255, 255, 255));
        txtUserError.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

        lblConfirmPassword.setFont(new java.awt.Font("Consolas", 1, 12)); // NOI18N
        lblConfirmPassword.setForeground(new java.awt.Color(255, 255, 255));
        lblConfirmPassword.setText("Confirm Password:");

        lblPassword.setFont(new java.awt.Font("Consolas", 1, 12)); // NOI18N
        lblPassword.setForeground(new java.awt.Color(255, 255, 255));
        lblPassword.setText("Password:");

        lblExistUser.setFont(new java.awt.Font("Consolas", 1, 14)); // NOI18N
        lblExistUser.setForeground(new java.awt.Color(255, 255, 255));
        lblExistUser.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblExistUser.setText("Existing User");

        lblNewUser1.setFont(new java.awt.Font("Consolas", 1, 14)); // NOI18N
        lblNewUser1.setForeground(new java.awt.Color(255, 255, 255));
        lblNewUser1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblNewUser1.setText("New User");

        btnSignUp.setBackground(new java.awt.Color(51, 51, 51));
        btnSignUp.setFont(new java.awt.Font("Consolas", 1, 12)); // NOI18N
        btnSignUp.setForeground(new java.awt.Color(255, 255, 255));
        btnSignUp.setText("Sign Up");
        btnSignUp.setBorder(null);
        btnSignUp.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnSignUpActionPerformed(evt);
            }
        });

        btnLogin.setBackground(new java.awt.Color(51, 51, 51));
        btnLogin.setFont(new java.awt.Font("Consolas", 1, 12)); // NOI18N
        btnLogin.setForeground(new java.awt.Color(255, 255, 255));
        btnLogin.setText("Login");
        btnLogin.setBorder(null);
        btnLogin.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnLoginActionPerformed(evt);
            }
        });

        lblPassword2.setFont(new java.awt.Font("Consolas", 1, 12)); // NOI18N
        lblPassword2.setForeground(new java.awt.Color(255, 255, 255));
        lblPassword2.setText("Password:");

        lblUsername2.setFont(new java.awt.Font("Consolas", 1, 12)); // NOI18N
        lblUsername2.setForeground(new java.awt.Color(255, 255, 255));
        lblUsername2.setText("Username:");

        txtArea.setColumns(20);
        txtArea.setRows(5);
        txtArea.setText("Change to label");
        pnlBackground.setViewportView(txtArea);

        lblNewUser.setFont(new java.awt.Font("Consolas", 1, 14)); // NOI18N
        lblNewUser.setForeground(new java.awt.Color(255, 255, 255));
        lblNewUser.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblNewUser.setText("Existing User");

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(310, 310, 310)
                        .addComponent(pnlBackground,
javax.swing.GroupLayout.PREFERRED_SIZE, 269, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
                        .addComponent(txtUserError,
javax.swing.GroupLayout.PREFERRED_SIZE, 804, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(107, 107, 107)
                        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.TRAILING)
                            .addGroup(jPanel1Layout.createSequentialGroup()
                                .addComponent(lblUsername1)
                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlaceme
nt.UNRELATED)
                                .addComponent(txtUsernameExist,
javax.swing.GroupLayout.PREFERRED_SIZE, 116, javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.G
roupLayout.Alignment.LEADING)
                                    .addGroup(jPanel1Layout.createSequentialGroup()
                                        .addGap(210, 210, 210)
                                        .addComponent(lblConfirmPassword,
javax.swing.GroupLayout.PREFERRED_SIZE, 120, javax.swing.GroupLayout.PREFERRED_SIZE)
                                        .addGap(18, 18, 18)
                                        .addComponent(txtPassNew2,
javax.swing.GroupLayout.PREFERRED_SIZE, 116, javax.swing.GroupLayout.PREFERRED_SIZE))
                                    .addGroup(jPanel1Layout.createSequentialGroup()
                                        .addGap(259, 259, 259)
                                        .addGroup(jPanel1Layout.createParallelGroup(javax
.swing.GroupLayout.Alignment.LEADING)
                                            .addComponent(lblPassword2,
javax.swing.GroupLayout.PREFERRED_SIZE, 86, javax.swing.GroupLayout.PREFERRED_SIZE)
                                            .addGroup(jPanel1Layout.createSequentialGroup
()
                                                .addComponent(lblUsername2,
javax.swing.GroupLayout.PREFERRED_SIZE, 71, javax.swing.GroupLayout.PREFERRED_SIZE)
                                                .addGap(18, 18, 18)
                                                .addComponent(txtUsernameNew,
javax.swing.GroupLayout.PREFERRED_SIZE, 116, javax.swing.GroupLayout.PREFERRED_SIZE))
                                            .addComponent(lblNewUser,
javax.swing.GroupLayout.PREFERRED_SIZE, 106, javax.swing.GroupLayout.PREFERRED_SIZE)))))
                            .addGroup(jPanel1Layout.createSequentialGroup()
                                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.G
roupLayout.Alignment.TRAILING)
                                    .addComponent(btnLogin,
javax.swing.GroupLayout.PREFERRED_SIZE, 50, javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addGroup(jPanel1Layout.createSequentialGroup()
                                        .addComponent(lblPassword)
                                        .addPreferredGap(javax.swing.LayoutStyle.Componen
tPlacement.UNRELATED)
                                        .addComponent(txtPassExist,
javax.swing.GroupLayout.PREFERRED_SIZE, 116, javax.swing.GroupLayout.PREFERRED_SIZE)))
                                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.G
roupLayout.Alignment.LEADING)
                                    .addGroup(jPanel1Layout.createSequentialGroup()
                                        .addGap(349, 349, 349)
                                        .addComponent(txtPassNew1,
javax.swing.GroupLayout.PREFERRED_SIZE, 116, javax.swing.GroupLayout.PREFERRED_SIZE))
                                    .addGroup(jPanel1Layout.createSequentialGroup()
                                        .addGap(405, 405, 405)
                                        .addComponent(btnSignUp,
javax.swing.GroupLayout.PREFERRED_SIZE, 60, javax.swing.GroupLayout.PREFERRED_SIZE))))))
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(177, 177, 177)
                .addComponent(lblExistUser, javax.swing.GroupLayout.PREFERRED_SIZE, 106,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
                .addComponent(lblNewUser1, javax.swing.GroupLayout.PREFERRED_SIZE, 93,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(167, 167, 167))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(61, 61, 61)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
                    .addComponent(lblExistUser, javax.swing.GroupLayout.PREFERRED_SIZE,
26, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(lblNewUser1, javax.swing.GroupLayout.PREFERRED_SIZE,
26, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(lblNewUser, javax.swing.GroupLayout.PREFERRED_SIZE, 26,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(28, 28, 28)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
                    .addComponent(txtUsernameExist,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(lblUsername1)
                    .addComponent(txtUsernameNew, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(lblUsername2))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
                    .addComponent(txtPassExist, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(lblPassword)
                    .addComponent(txtPassNew1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(lblPassword2))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
                    .addComponent(txtPassNew2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(lblConfirmPassword))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 24,
Short.MAX_VALUE)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
                    .addComponent(btnLogin, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(btnSignUp, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(51, 51, 51)
                        .addComponent(txtUserError,
javax.swing.GroupLayout.PREFERRED_SIZE, 36, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(18, 18, 18)
                        .addComponent(pnlBackground,
javax.swing.GroupLayout.PREFERRED_SIZE, 43, javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGap(152, 152, 152))
        );

        getContentPane().add(jPanel1);
        jPanel1.setBounds(-20, -10, 900, 510);

        setSize(new java.awt.Dimension(839, 416));
        setLocationRelativeTo(null);
    }// </editor-fold>

    private void btnSignUpActionPerformed(java.awt.event.ActionEvent evt)
    {
        UserArray m = new UserArray();
        if (m.getUser(txtUsernameNew.getText()) != -1)
        {
            if ((txtPassNew1.getText()).equals(txtPassNew2.getText()))
            {
                m.addUser(txtUsernameNew.getText(), txtPassNew1.getText(), "Not
completed#Not completed#Not completed", 1);

                new MainScreen().setVisible(true);
                this.dispose();
            } else
            {
                txtArea.setText("Passwords do not match");
                txtArea.setBackground(Color.red);
            }
        } else
        {
            txtArea.setText("User already exists");
            txtArea.setBackground(Color.red);
        }
    }

    private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
        UserArray m = new UserArray();
        if (m.login(txtUsernameExist.getText(), txtPassExist.getText()))
        {
            new MainScreen().setVisible(true);
            this.dispose();
        } else
        {
            txtArea.setText("Password or Username is incorrect");
            txtArea.setBackground(Color.red);
        }
    }

    public static void main(String args[])
    {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code 
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default 
look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/
lookandfeel/plaf.html
         */
        try
        {
            for (javax.swing.UIManager.LookAndFeelInfo info : 
javax.swing.UIManager.getInstalledLookAndFeels())
            {
                if ("Nimbus".equals(info.getName()))
                {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex)
        {

java.util.logging.Logger.getLogger(LoginScreen.class.getName()).log(java.util.logging.Lev
el.SEVERE, null, ex);
        } catch (InstantiationException ex)
        {

java.util.logging.Logger.getLogger(LoginScreen.class.getName()).log(java.util.logging.Lev
el.SEVERE, null, ex);
        } catch (IllegalAccessException ex)
        {

java.util.logging.Logger.getLogger(LoginScreen.class.getName()).log(java.util.logging.Lev
el.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex)

{  java.util.logging.Logger.getLogger(LoginScreen.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                new LoginScreen().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton btnLogin;
    private javax.swing.JButton btnSignUp;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JLabel lblConfirmPassword;
    private javax.swing.JLabel lblExistUser;
    private javax.swing.JLabel lblLogin;
    private javax.swing.JLabel lblNewUser;
    private javax.swing.JLabel lblNewUser1;
    private javax.swing.JLabel lblPassword;
    private javax.swing.JLabel lblPassword2;
    private javax.swing.JLabel lblUsername1;
    private javax.swing.JLabel lblUsername2;
    private javax.swing.JScrollPane pnlBackground;
    private javax.swing.JTextArea txtArea;
    private javax.swing.JPasswordField txtPassExist;
    private javax.swing.JPasswordField txtPassNew1;
    private javax.swing.JPasswordField txtPassNew2;
    private javax.swing.JLabel txtUserError;
    private javax.swing.JTextField txtUsernameExist;
    private javax.swing.JPasswordField txtUsernameNew;
    // End of variables declaration
}
```

Main Screen

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package anderson;


public class MainScreen extends javax.swing.JFrame
{


    public MainScreen()
    {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        pnlBackground = new javax.swing.JPanel();
        txtMainScreen = new javax.swing.JLabel();
        txtHelp = new javax.swing.JLabel();
        lblBestTime = new javax.swing.JLabel();
        txtLevelSelect = new javax.swing.JLabel();
        btnBestTime = new javax.swing.JButton();
        btnLevelSelect = new javax.swing.JButton();
        btnHelp = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Main");
        setPreferredSize(new java.awt.Dimension(1934, 1121));
        getContentPane().setLayout(null);

        pnlBackground.setBackground(new java.awt.Color(0, 0, 0));
        pnlBackground.setLayout(null);

        txtMainScreen.setFont(new java.awt.Font("Consolas", 1, 48)); // NOI18N
        txtMainScreen.setForeground(new java.awt.Color(255, 255, 255));
        txtMainScreen.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        txtMainScreen.setText("Main Screen");
        pnlBackground.add(txtMainScreen);
        txtMainScreen.setBounds(35, 104, 1900, 58);

        txtHelp.setFont(new java.awt.Font("Consolas", 1, 36)); // NOI18N
        txtHelp.setForeground(new java.awt.Color(255, 255, 255));
        txtHelp.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        txtHelp.setText("Help");
        pnlBackground.add(txtHelp);
        txtHelp.setBounds(1410, 420, 180, 40);

        lblBestTime.setFont(new java.awt.Font("Consolas", 1, 36)); // NOI18N
        lblBestTime.setForeground(new java.awt.Color(102, 102, 102));
        lblBestTime.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lblBestTime.setText("Best Time");
        pnlBackground.add(lblBestTime);
        lblBestTime.setBounds(280, 420, 290, 43);

        txtLevelSelect.setFont(new java.awt.Font("Consolas", 1, 36)); // NOI18N
        txtLevelSelect.setForeground(new java.awt.Color(255, 255, 255));
        txtLevelSelect.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        txtLevelSelect.setText("Level Select");
        pnlBackground.add(txtLevelSelect);
        txtLevelSelect.setBounds(840, 430, 260, 43);

        btnBestTime.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/level1.png"))); // NOI18N
        btnBestTime.setActionCommand("3");
        btnBestTime.setBorder(null);
        btnBestTime.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnHelpActionPerformed(evt);
            }
        });
        pnlBackground.add(btnBestTime);
        btnBestTime.setBounds(250, 330, 350, 350);

        btnLevelSelect.setIcon(new javax.swing.ImageIcon(getClass().getResource("/
anderson/Images/level2.png"))); // NOI18N
        btnLevelSelect.setActionCommand("1");
        btnLevelSelect.setBorder(null);
        btnLevelSelect.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnHelpActionPerformed(evt);
            }
        });
        pnlBackground.add(btnLevelSelect);
        btnLevelSelect.setBounds(800, 330, 350, 350);

        btnHelp.setIcon(new javax.swing.ImageIcon(getClass().getResource("/anderson/
Images/level3.png"))); // NOI18N
        btnHelp.setActionCommand("2");
        btnHelp.setBorder(null);
        btnHelp.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnHelpActionPerformed(evt);
            }
        });
        pnlBackground.add(btnHelp);
        btnHelp.setBounds(1330, 320, 350, 350);

        getContentPane().add(pnlBackground);
        pnlBackground.setBounds(-10, -10, 1970, 1160);

        setSize(new java.awt.Dimension(1952, 1168));
        setLocationRelativeTo(null);
    }// </editor-fold>

    private void btnHelpActionPerformed(java.awt.event.ActionEvent evt)
    {
        if (evt.getActionCommand().equals("2"))
            new HelpScreen().setVisible(true);

        if (evt.getActionCommand().equals("3"))
            new BestTimeScreen().setVisible(true);

        if (evt.getActionCommand().equals("1"))
            new LevelSelectScreen().setVisible(true);

        this.dispose();
    }


    public static void main(String args[])
    {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code 
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default 
look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/
lookandfeel/plaf.html
         */
        try
        {
            for (javax.swing.UIManager.LookAndFeelInfo info : 
javax.swing.UIManager.getInstalledLookAndFeels())
            {
                if ("Nimbus".equals(info.getName()))
                {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex)
        {
```

```
java.util.logging.Logger.getLogger(MainScreen.class.getName()).log(java.util.logging.Leve
l.SEVERE, null, ex);
        } catch (InstantiationException ex)
        {

java.util.logging.Logger.getLogger(MainScreen.class.getName()).log(java.util.logging.Leve
l.SEVERE, null, ex);
        } catch (IllegalAccessException ex)
        {

java.util.logging.Logger.getLogger(MainScreen.class.getName()).log(java.util.logging.Leve
l.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex)
        {

java.util.logging.Logger.getLogger(MainScreen.class.getName()).log(java.util.logging.Leve
l.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                new MainScreen().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton btnBestTime;
    private javax.swing.JButton btnHelp;
    private javax.swing.JButton btnLevelSelect;
    private javax.swing.JLabel lblBestTime;
    private javax.swing.JPanel pnlBackground;
    private javax.swing.JLabel txtHelp;
    private javax.swing.JLabel txtLevelSelect;
    private javax.swing.JLabel txtMainScreen;
    // End of variables declaration
}
```