CS643 – Object Oriented Programming (OOP) Project

OOP Game

Under supervision of:

Under Supervision of:

Professor Doctor \ Mohamed Kholief

Eng \ Mohamed Galal

By Students:

Khaled Mohamed Ashraf Dokmak (Reg Num 21135001)

Lamiaa Ibrahim Mohamed Khamis (Reg Num 20235023)

Project Overview:

This project is (Application Game) It was implemented using the Java programming language, These classes interacting all together through the Object-Oriented Programming concepts that we have studied through the course such as;

- Encapsulation.
- Abstraction.
- Association, (Aggregation, Composition, Inheritance).
- Polymorphism.

Application Structure & Design Process:

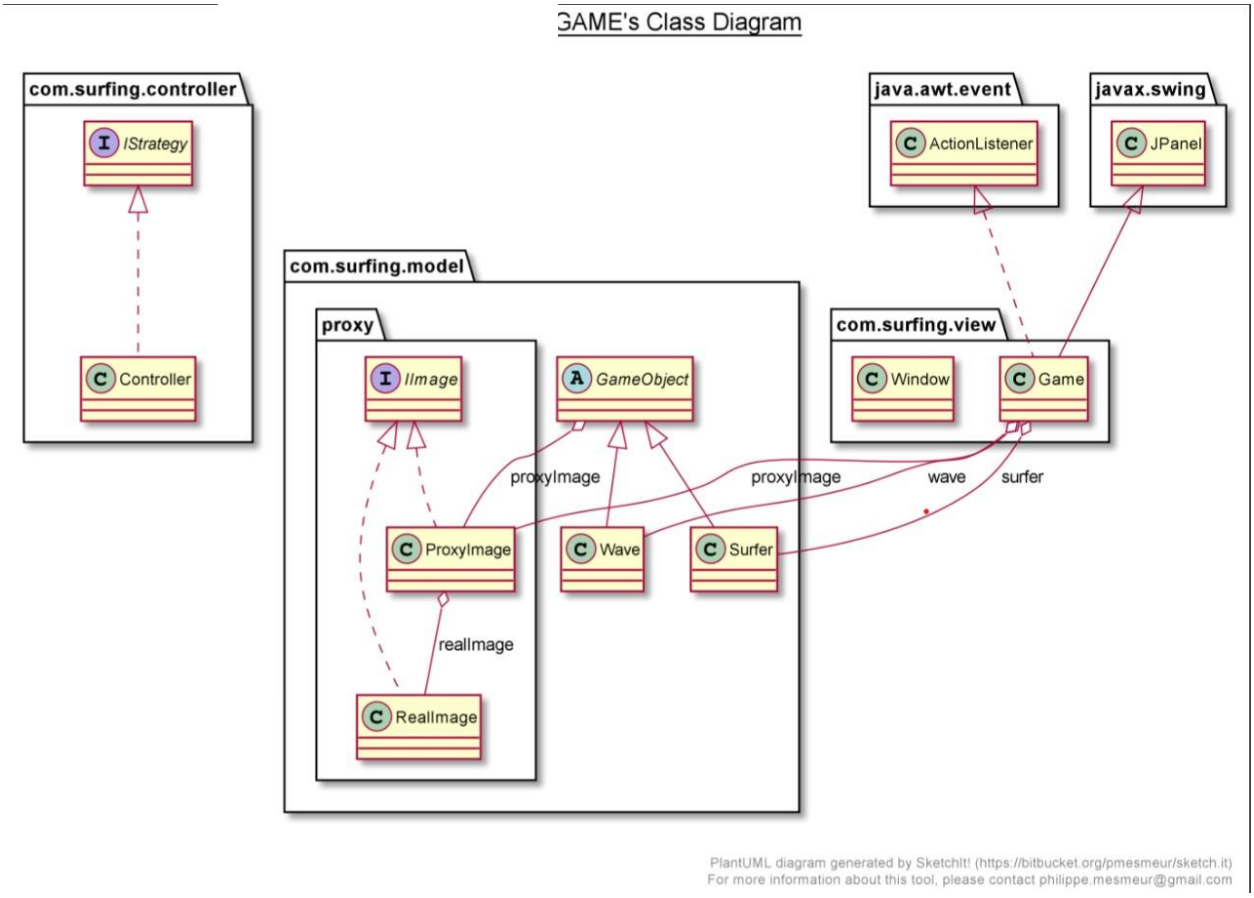
- **Package :** com.sufing.view
- **Class:** Windows
- **Class:** Game

- **Package :** com.sufing.model
- **Abstract Class:** GameObject
- **Class:** Wave
- **Class:** Surfer

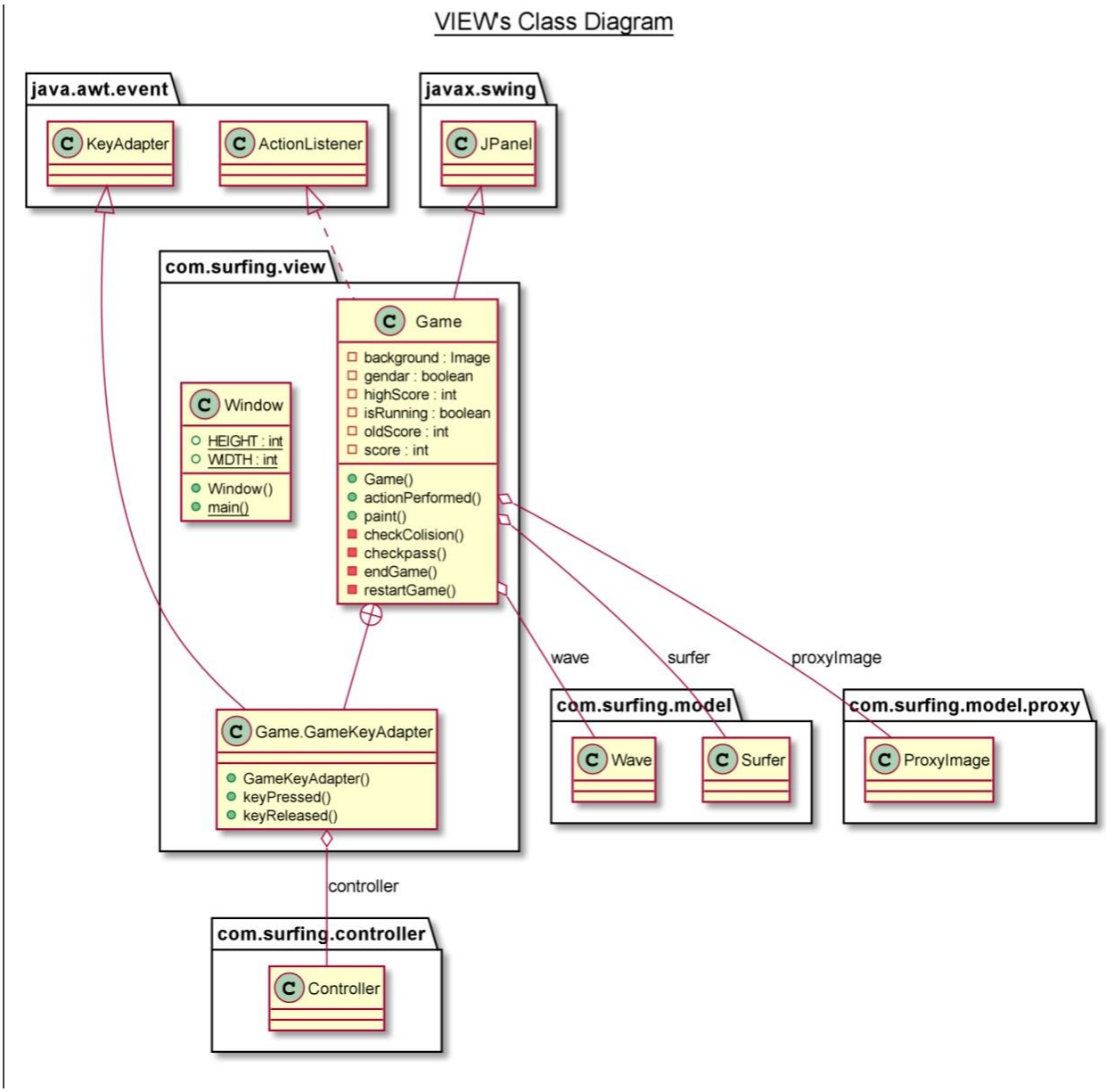
- **Package :** com.sufing.view
- **Class:** Windows
- **Class:** Game

- **Package :** proxy
- **Interface :** IImage
- **Class:** ProxyImage
- **Class:** RealImage

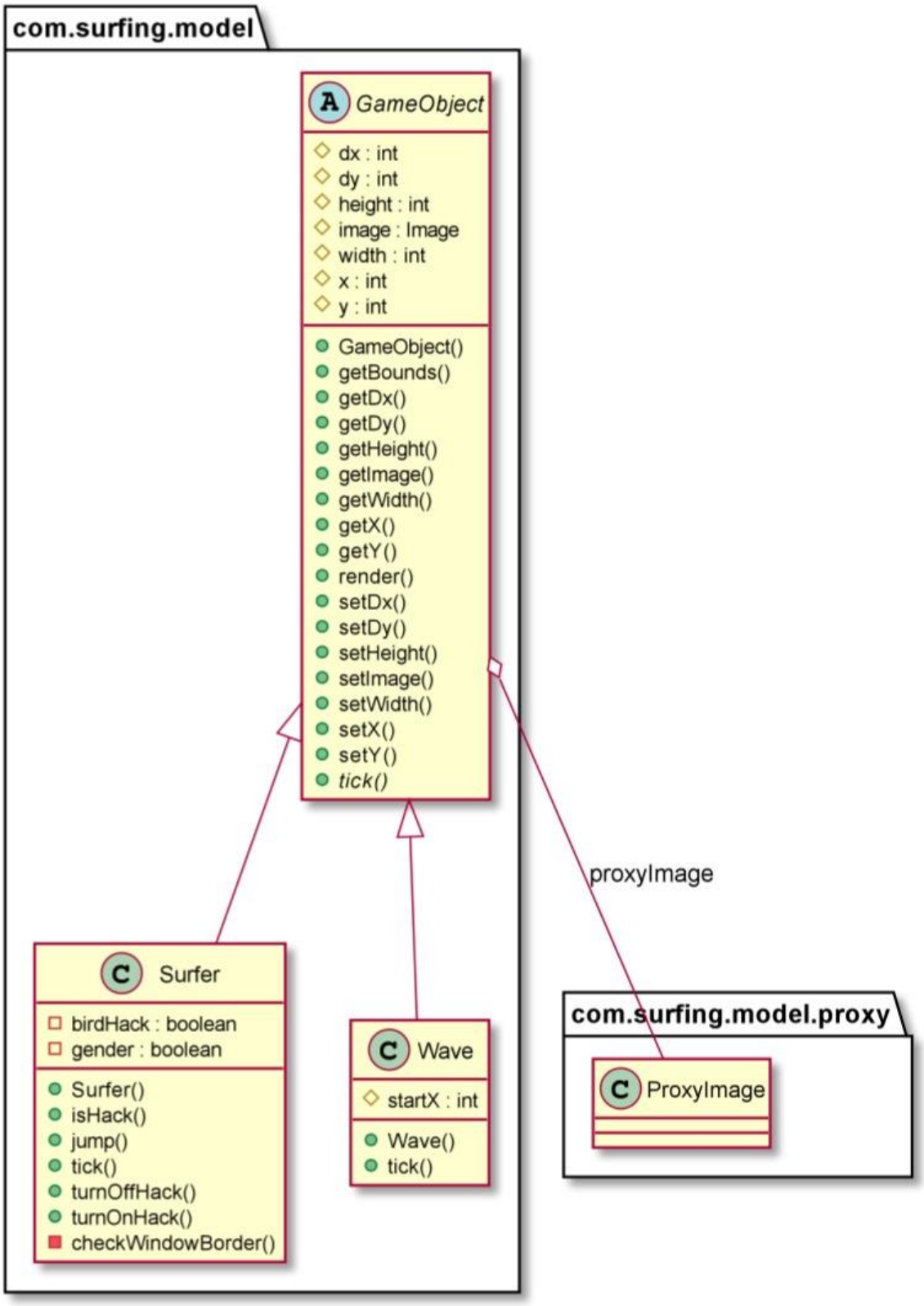
- **Package :** Controller
- **Interface:** IStrategy
- **Class:** Controlle


GAME's Class Diagram
com.surfing.controller
IStrategy
Controller
java.awt.event
ActionListener
javax.swing
JPanel
com.surfing.model
proxy
IImage
GameObject
ProxyImage
Wave
Surfer
RealImage
realImage
com.surfing.view
Window
Game
proxyImage
proxyImage
wave
surfer
PlantUML diagram generated by SketchIt! (https://bitbucket.org/pmesmeur/sketch.it)
For more information about this tool, please contact philippe.mesmeur@gmail.com

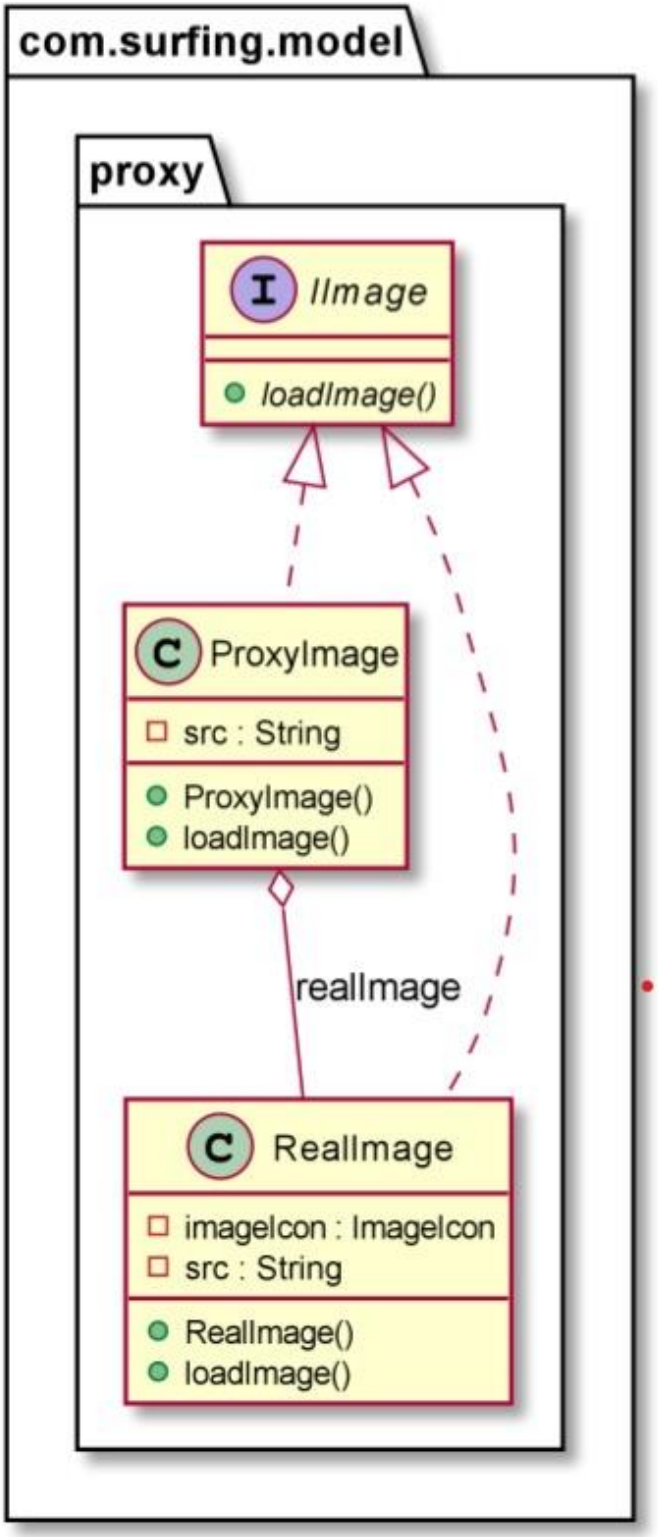
VIEW's Class Diagram



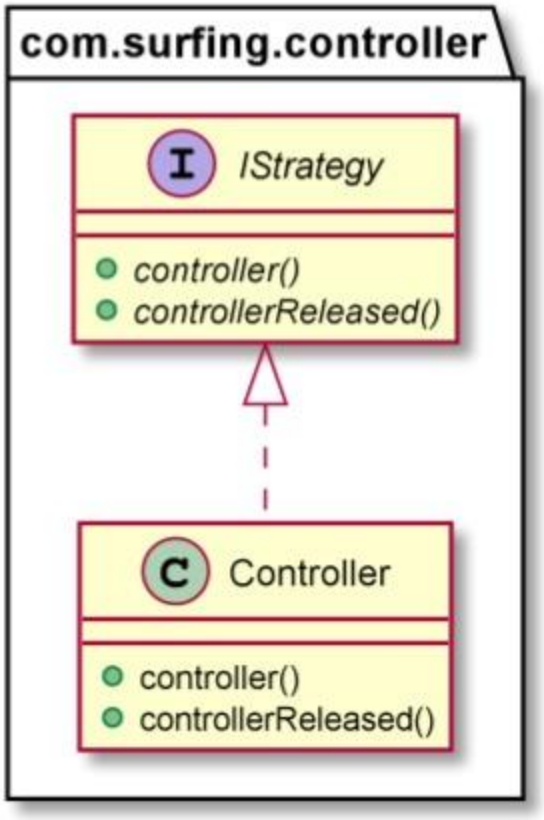
MODEL's Class Diagram



PROXY's Class Diagram



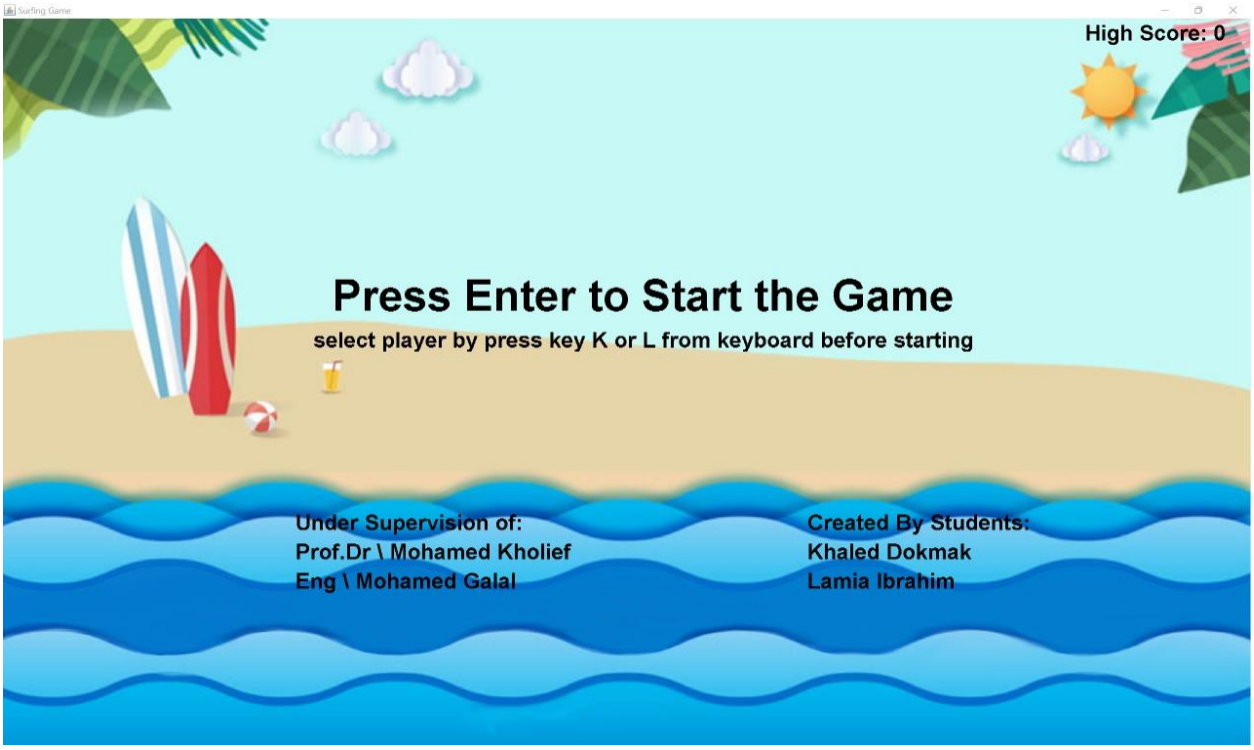
CONTROLLER's Class Diagram

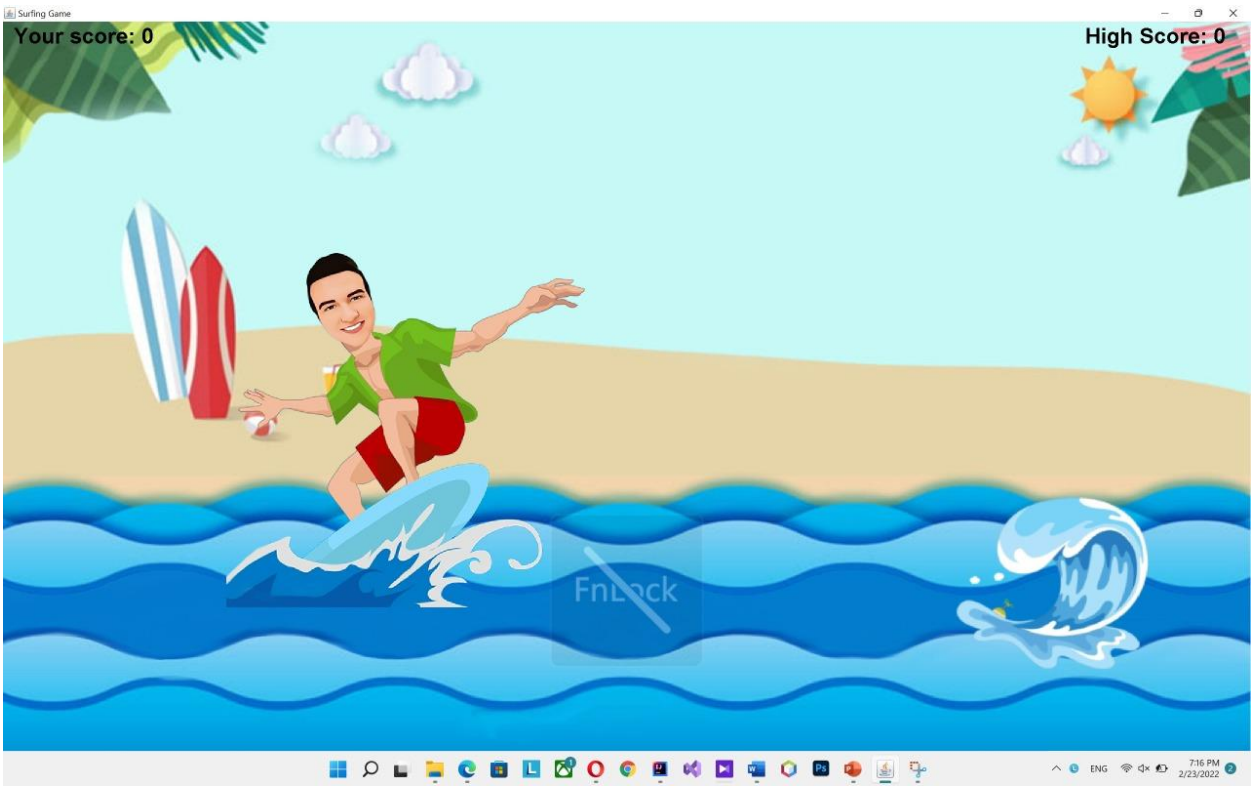


3.5 Defining visibility

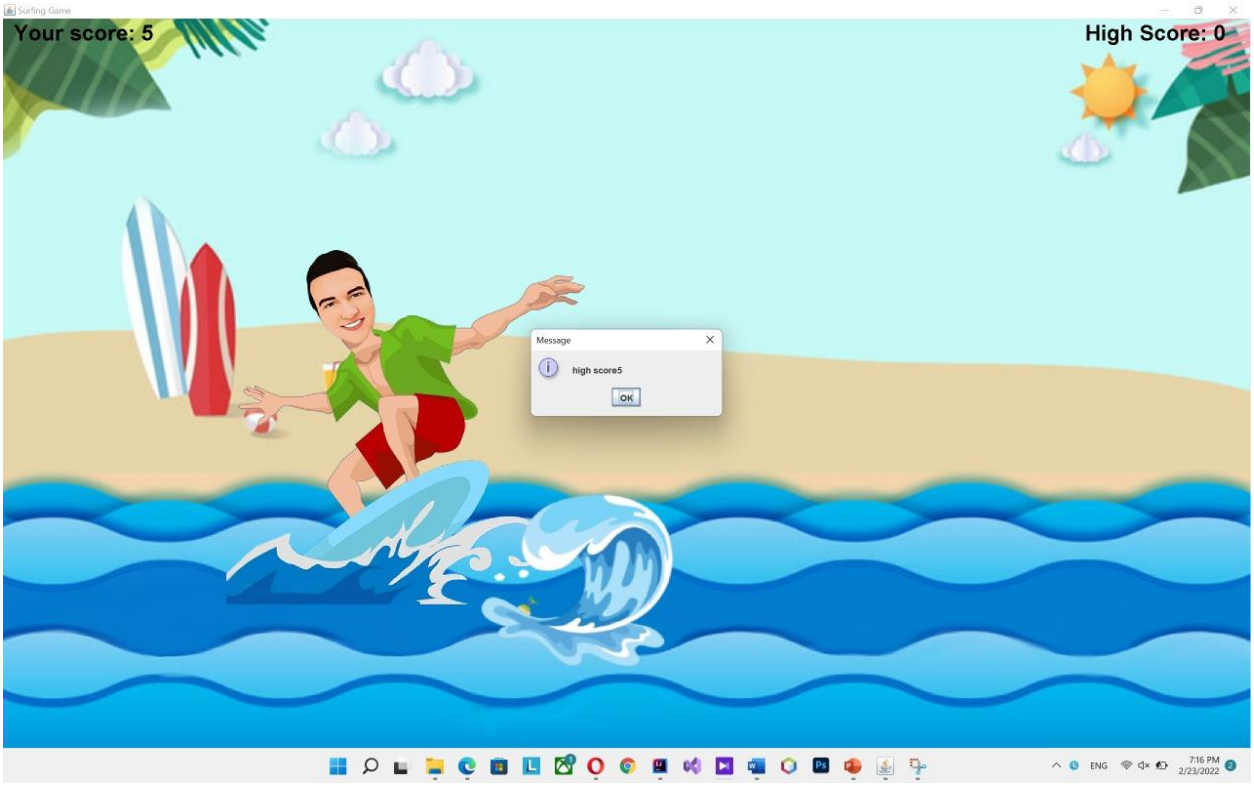
When you define methods or fields, you can use characters to define the visibility of the corresponding item:

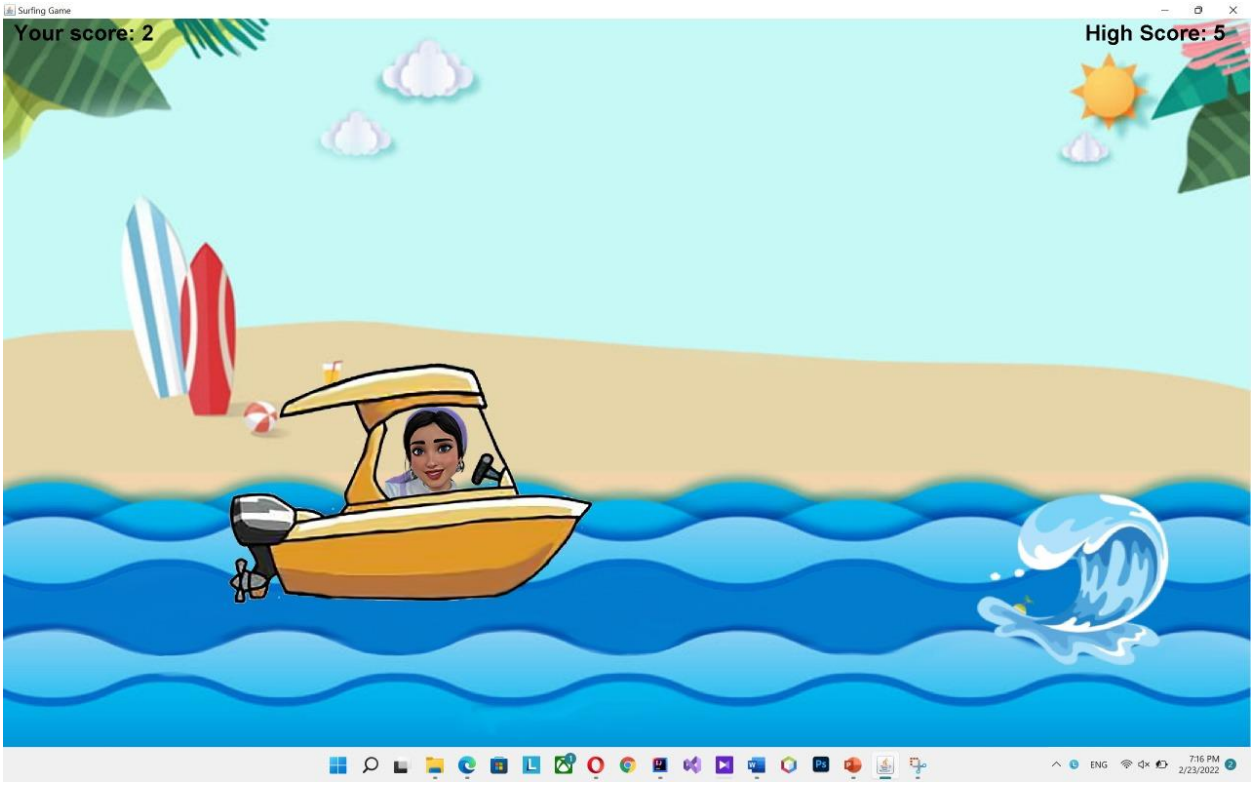
Character	Icon for field	Icon for method	Visibility
-	□	■	private
#	◇	◆	protected
~	△	▲	package private
+	○	●	public

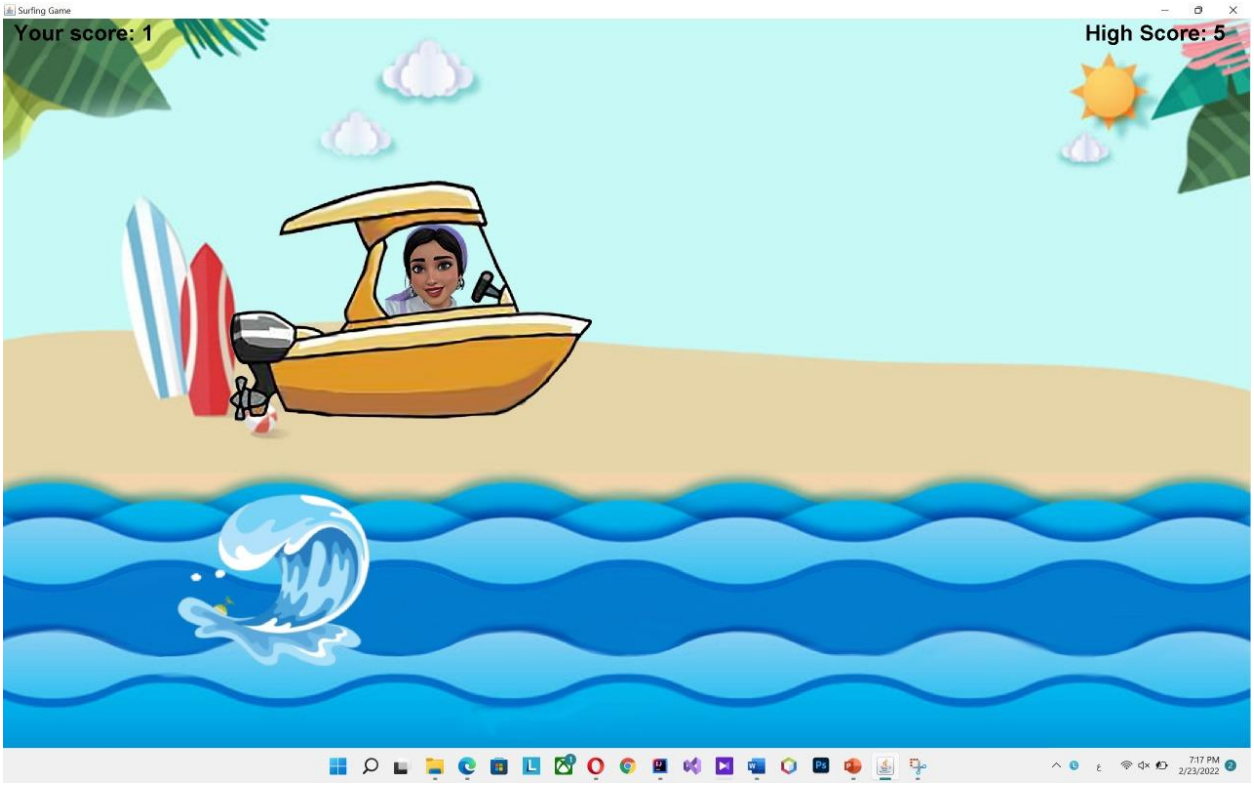

Surfing Game
High Score: 0
Press Enter to Start the Game
select player by press key K or L from keyboard before starting
Under Supervision of:
Prof.Dr \ Mohamed Kholief
Eng \ Mohamed Galal
Created By Students:
Khaled Dokmak
Lamia Ibrahim

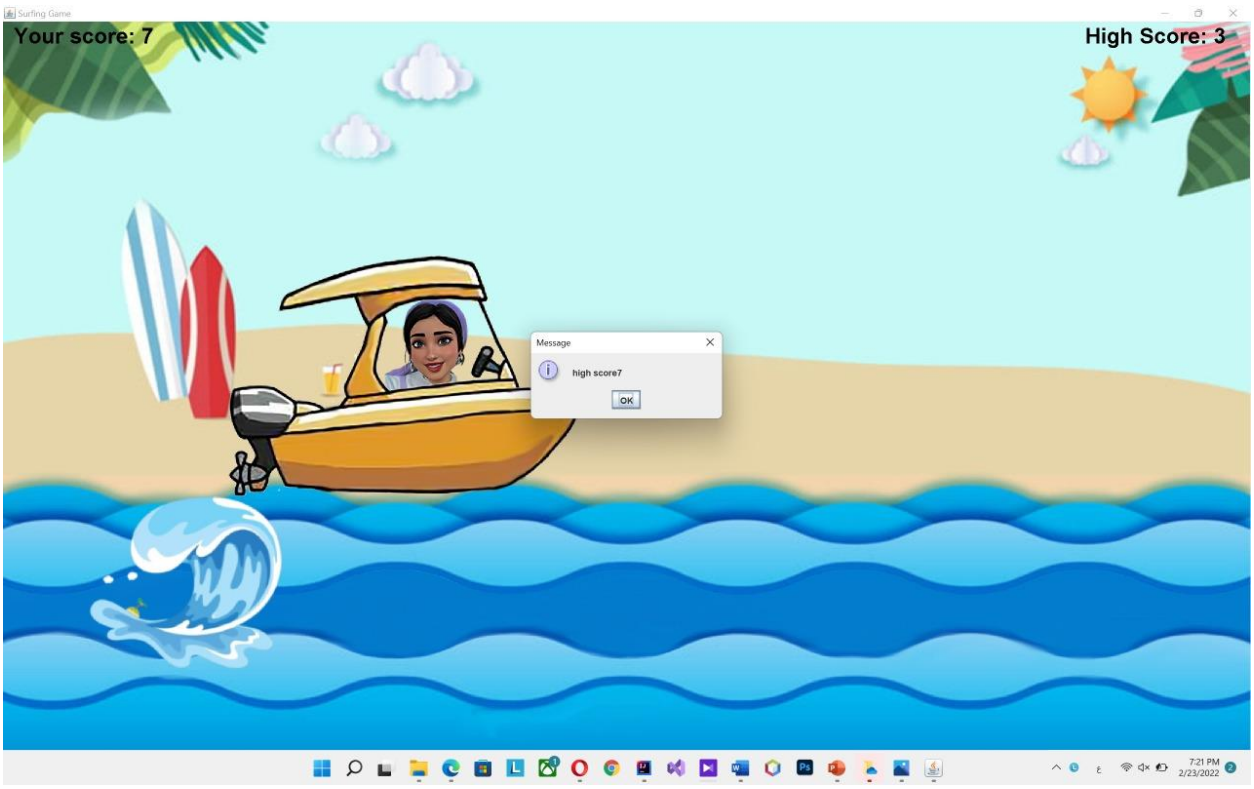

Surfing Game
Your score: 0
High Score: 0
FnLock
ENG
7:16 PM
2/23/2022


Surfing Game
Your score: 3
High Score: 0
ENG
7:16 PM
2/23/2022


Surfing Game
Your score: 5
High Score: 0
Message
high score5
OK
ENG
7:16 PM
2/23/2022


Surfing Game
Your score: 2
High Score: 5
ENG
7:16 PM
2/23/2022


Surfing Game
Your score: 1
High Score: 5
7:17 PM
2/23/2022


Surfing Game
Your score: 7
High Score: 3
Message
high score7
OK
7:21 PM
2/23/2022

Appendix I: package com.surfing.view :

I.I Main Class Window:

```
package com.surfing.view;

import java.awt.Dimension;

import javax.swing.*;


public class Window {

    //WIDTH and HEIGHT of Window

    public static int WIDTH = 1700;

    public static int HEIGHT = 1000;

    //construction

    public Window(int width, int height, String title, Game game) {

        JFrame frame = new JFrame();

        frame.add(game);

        //title of Window

        frame.setTitle(title);

        //Exit the application

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

       // frame.setMaximumSize(new Dimension(width, height));

        frame.setPreferredSize(new Dimension(width, height));

        frame.setMinimumSize(new Dimension(width, height));

        frame.setLocationRelativeTo(null);

        //frame.setResizable(false);

        frame.setVisible(true);


    }

    public static void main(String[] args) {
```

```
        Game game = new Game();


       try {

javax.swing.UIManager.setLookAndFeel(javax.swing.UIManager.getSystemLookAndFe
elClassName());

        }

        catch (ClassNotFoundException | InstantiationException |
IllegalAccessException | javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Window.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);

        }

        java.awt.EventQueue.invokeLater(() -> {


            Window window = new Window(WIDTH, HEIGHT, "Surfing Game", game);


       });




    }

}
```

I.II Class Game:

```
package com.surfing.view;
```

```java
import com.surfing.controller.Controller;
import com.surfing.model.*;
import com.surfing.model.proxy.ProxyImage;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.Rectangle;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import javax.swing.*;

public class Game extends JPanel implements ActionListener {
    private boolean isRunning = false;
    private ProxyImage proxyImage;
    private Image background;
    private Surfer surfer;
    private Wave wave;
    private int score;
    private int oldScore;
    private int highScore;
    private boolean gendar;

    public Game() {
```

```
        proxyImage = new ProxyImage("/assets/background.png");
        background = proxyImage.loadImage().getImage();
        setFocusable(true);
        setDoubleBuffered(false);
        GameKeyAdapter gameKeyAdapter = new GameKeyAdapter();
        addKeyListener(gameKeyAdapter);
        Timer timer = new Timer(15, this);
        timer.start();
    }

        @Override
    public void paint(Graphics g) {
        Graphics2D g2 = (Graphics2D) g;
        g2.drawImage(background, 0, 0, null);
        if (isRunning) {
            this.surfer.render(g2, this);
            this.wave.render(g2, this);

            g2.setColor(Color.black);
            g.setFont(new Font("Arial", 1, 30));
            g2.drawString("Your score: " + this.score, 15, 30);

        } else {
            g2.setColor(Color.black);
            g.setFont(new Font("Arial", 1, 60));
            g2.drawString("Press Enter to Start the Game", Window.WIDTH / 2 -
400, Window.HEIGHT / 2 - 100);
            g2.setColor(Color.black);
```

```
            g.setFont(new Font("Arial", 1, 30));

            g2.drawString("select player by press key K or L from keyboard
before starting" , Window.WIDTH / 2 - 425, Window.HEIGHT / 2 - 50);


           g.setFont(new Font("Arial", 1, 30));

            g2.drawString("Under Supervision of:", 400, Window.HEIGHT - 300);

            g2.drawString("Prof.Dr \\ Mohamed Kholief", 400, Window.HEIGHT -
260);

            g2.drawString("Eng \\ Mohamed Galal", 400, Window.HEIGHT - 220);


            g2.drawString("Created By Students: ", Window.WIDTH - 600,
Window.HEIGHT - 300);

            g2.drawString("Khaled Dokmak", Window.WIDTH - 600, Window.HEIGHT
- 260);

            g2.drawString("Lamia Ibrahim", Window.WIDTH - 600, Window.HEIGHT
- 220);


        }

        g2.setColor(Color.black);

        g.setFont(new Font("Arial", 1, 30));

        g2.drawString("High Score: " + highScore, Window.WIDTH - 220, 30);

        g.dispose();

    }


        @Override

    public void actionPerformed(ActionEvent e) {

        Toolkit.getDefaultToolkit().sync();

        if (isRunning) {

            surfer.tick();

            wave.tick();

            checkColision();
```

```
            checkpass();
        }

        repaint();
    }

    private void restartGame() {
        if (!isRunning) {
            this.isRunning = true;
            this.surfer = new Surfer(Window.WIDTH / 3 + 100, Window.HEIGHT /
2 ,gendar);
            this.wave = new Wave(Window.WIDTH+400, Window.HEIGHT - 400);

            oldScore = score = 0;
        }
    }

    private void endGame() {
        this.isRunning = false;
        if (score > highScore) {
            this.highScore = score;
            JOptionPane.showMessageDialog(null, "high score" + highScore);
        } else
            JOptionPane.showMessageDialog(null, "your score" + this.score);

    }
```

```
private void checkColision() {
    Rectangle rectSurfer = this.surfer.getBounds();
    Rectangle rectWave = wave.getBounds();

    if (rectSurfer.intersects(rectWave) && surfer.isHack()) {
        endGame();
    }
}

private void checkpass() {
    if (wave.getX() < surfer.getX() && oldScore == score) score++;
    if (wave.getX() > surfer.getX() && oldScore != score) oldScore++;
}

    private class GameKeyAdapter extends KeyAdapter {

    private final Controller controller;

    public GameKeyAdapter() {

        controller = new Controller();

    }

    //Invoked when a key has been pressed.
    @Override
    public void keyPressed(KeyEvent e) {
        if (e.getKeyCode() == KeyEvent.VK_K) {
```

```
                gendar=false;
            }
            if (e.getKeyCode() == KeyEvent.VK_L) {
                gendar=true;
            }
            if (e.getKeyCode() == KeyEvent.VK_ENTER) {
                restartGame();
            }
        }

        //Invoked when a key has been released.
        @Override
        public void keyReleased(KeyEvent e) {
            if (isRunning) {
                controller.controllerReleased(surfer, e);
            }
        }
    }


}
```

Appendix II: package com.surfing.model:

II.I Abstract Class GameObject:

```
package com.surfing.model;

import com.surfing.model.proxy.ProxyImage;

import java.awt.*;

import java.awt.image.ImageObserver;


public abstract class GameObject {

    protected int x;

    protected int y;

    protected int dx;

    protected int dy;

    protected int width;

    protected int height;

    protected Image image;

    protected ProxyImage proxyImage;



    public GameObject(int x, int y) {

        this.x = x;

        this.y = y;

    }

    public int getX() {

        return x;

    }
```

```
public int getY() {
    return y;
}

public int getDx() {
    return dx;
}

public int getDy() {
    return dy;
}

public int getWidth() {
    return width;
}

public int getHeight() {
    return height;
}

public Image getImage() {
    return image;
}

public void setX(int x) {
    this.x = x;
}
```

```
public void setY(int y) {
    this.y = y;
}

public void setDx(int dx) {
    this.dx = dx;
}

public void setDy(int dy) {
    this.dy = dy;
}

public void setWidth(int width) {
    this.width = width;
}

public void setHeight(int height) {
    this.height = height;
}

public void setImage(Image image) {
    this.image = image;
}

public Rectangle getBounds() {
    return new Rectangle(x, y, width, height);
}
```

```
    public void render(Graphics2D g, ImageObserver obs) {
        g.drawImage(image, x, y, obs);
    }

        public abstract void tick();

}
```

II.II Class Wave:

```
package com.surfing.model;


import com.surfing.model.proxy.ProxyImage;


import java.util.Random;


public class Wave extends GameObject {
    protected int startX;


    public Wave(int x, int y) {
        super(x, y);
        if (proxyImage == null) {
```

```
            proxyImage = new ProxyImage("/assets/wave.png");



        }

        this.image = proxyImage.loadImage().getImage();

        this.width = image.getWidth(null);

        this.height = image.getHeight(null);

        this.x = x;

        this.y = y;

        this.startX=x;

        this.dx = 15;



    }



    @Override

    public void tick( ) {

        this.x -=  this.dx;

        if(x < 0) {

            this.x=startX;

            dx++;
```

```
            if (dx>30) dx=30;

        }

    }


}
```

II.III Class Surfer:

```
package com.surfing.model;

import com.surfing.model.proxy.ProxyImage;

import com.surfing.view.Window;



public class Surfer extends GameObject {

    private boolean gender;

    private boolean Hack;

    public Surfer(int x, int y,boolean gendar){

        super(x, y);

        if(this.proxyImage == null && gendar==false) {

                this.proxyImage = new ProxyImage("/assets/khaled.png");

        }

        if(this.proxyImage == null && gendar==true) {

            this.proxyImage = new ProxyImage("/assets/lamia.png");

        }


        this.image = this.proxyImage.loadImage().getImage();

        this.width = this.image.getWidth(null)-170;
```

```
        this.height = this.image.getHeight(null)-50;
        this.x -= this.width+30;
        this.y -= this.height;
        this.dy = 4;
    }
    @Override
    public void tick() {
        if(dy < 10) {
            dy += 2;
        }
        this.y += dy;
        checkWindowBorder();
    }

    public void jump() {
        if(dy > 0) {
            dy = 0;
        }
        if(this.y == Window.HEIGHT - 700)
        dy -= 45;
    }
    public boolean isHack() {
        return !Hack;
    }
    public void turnOnHack() {
        this.Hack = true;
    }
    public void turnOffHack() {
```

```
            this.Hack = false;
        }
        private void checkWindowBorder() {
            if(this.x > Window.WIDTH) {
                this.x = Window.WIDTH;
            }
            if(this.x < 0) {
                this.x = 0;
            }
            if(this.y > Window.HEIGHT - 700) {
                this.y = Window.HEIGHT - 700;
            }

        }
}
```

II.V package com.surfing.model.proxy:

II.V.I Interface IImage:

```
package com.surfing.model.proxy;


import javax.swing.ImageIcon;


public interface IImage {

    public ImageIcon loadImage();
```

```
}
```

II.V.II Class ProxyImage:

```
package com.surfing.model.proxy;

import javax.swing.ImageIcon;


public class ProxyImage implements IImage {


    private final String src;
    private RealImage realImage;


    public ProxyImage(String src) {
        this.src = src;
    }
    @Override
    public ImageIcon loadImage() {
        if(realImage == null) {
            this.realImage = new RealImage(src);
        }


        return this.realImage.loadImage();
    }


}
```

II.V.III Class RealImage:

```java
package com.surfing.model.proxy;

import javax.swing.ImageIcon;

public class RealImage implements IImage {

    private final String src;
    private ImageIcon imageIcon;

    public RealImage(String src) {
        this.src = src;
    }
    @Override
    public ImageIcon loadImage() {
        if(imageIcon == null) {
            this.imageIcon = new ImageIcon(getClass().getResource(src));
        }
        return imageIcon;
    }

}
```

Appendix III: package com.surfing.controller:

III.I Interface IStrategy:

```
package com.surfing.controller;


import com.surfing.model.Surfer;
import java.awt.event.KeyEvent;



public interface IStrategy {


    public void controller(Surfer surfer, KeyEvent kevent);
    public void controllerReleased(Surfer surfer, KeyEvent kevent);
}
```

III.II Class Controller:

```java
package com.surfing.controller;

import com.surfing.model.Surfer;

import java.awt.event.KeyEvent;


public class Controller implements IStrategy {


    @Override

    public void controller(Surfer surfer, KeyEvent kevent) {

    }


    @Override

    public void controllerReleased(Surfer surfer, KeyEvent kevent) {

        if(kevent.getKeyCode() == KeyEvent.VK_SPACE) {

            surfer.jump();

        }

        if(kevent.getKeyCode() == KeyEvent.VK_H ) {

            surfer.turnOnHack();

        }

        if(kevent.getKeyCode() == KeyEvent.VK_J ) {

            surfer.turnOffHack();

        }

    }


}
```