Informe de Laboratorio 22
Tema: GUI : Interfaz Gráfica de Usuario

Nota

Estudiante	Escuela	Asignatura
Ryan Fabian Valdivia Segovia rvaldiviase@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Fundamentos de la programación 2 Semestre: II Código: 1701213

Laboratorio	Tema	Duración
22	GUI : Interfaz Gráfica de Usuario	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 15 de Enero 2023	Al 22 de Enero 2023

1. Tarea

- Cree una versión del videojuego de estrategia usando componentes básicos GUI: Etiquetas, botones, cuadros de texto, JOptionPane, Color.
- Además, utilizar componentes avanzados GUI: Layouts, JPanel, áreas de texto, checkbox, botones de radio y combobox.
- Considerar nivel estratégico y táctico.
- Considerar hasta las unidades especiales de los reinos.
- Hacerlo iterativo.

2. Equipos, materiales y temas utilizados

- Sistema Operativo Windows 11 Home Single Language 64 bits 22621.2283
- VIM 9.0.
- Visual Studio Code 64 bits 1.82.2
- OpenJDK 64-Bits 21.0.2
- Git 2.41.0.windows.1
- IntelliJ IDEA 2023.3 Runtime version: 17.0.9+7-b1087.7 amd64
- Cuenta en GitHub con el correo institucional.

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- `https://github.com/RyanValdivia/fp2-23b`
- URL para el laboratorio 22 en el Repositorio GitHub.
- `https://github.com/RyanValdivia/fp2-23b/tree/main/fase03/lab22`

4. Actividades

- Este laboratorio consistía en elaborar una interfaz gráfica de usuario, portando todo nuestro videojuego de laboratorios anteriores.
- Para esto, reutilicé el código de laboratorios anteriores, usando la misma lógica. Entonces reciclé las clases de laboratorios anteriores

Listing 1: Superclase Soldado

```
   package com.example.lab22;
import java.util.*;

public class Soldier {
    protected String name;
    protected String alias;
    protected int HP;
    protected int attack;
    protected int defense;
    protected int x;
    protected int y;
    protected boolean status;
    private String color;
    private int armyId;

    public Soldier(String name){
        this.name = name;
        this.status = true;
    }
    public Soldier(){
        this.name = "";
        this.alias = "";
        this.color = "#cdcdcd";
        this.status = false;
    }
   public void action(){};
    public static Soldier winner(Soldier s1, Soldier s2){
        Random random = new Random();
        double h1 = s1.getHP();
        double h2 = s2.getHP();
        double total = h1 + h2;
        h1 /= total;
        h2 /= total;
        double ans = random.nextDouble();
        if(h2 < h1){
            if(ans <= h2){
```

```
            return s2;
        }else{
            return s1;
        }
    }else{
        if(ans <= h1){
            return s1;
        }else{
            return s2;
        }
    }

}
public void copy(Soldier source){
    this.name = source.name;
    this.alias = source.alias;
    this.HP = source.HP;
    this.attack = source.attack;
    this.defense = source.defense;
    this.x = source.x;
    this.y = source.y;
    this.color = source.color;
    this.status = source.status;
    this.armyId = source.armyId;
}
public void destroy(){
    this.name = "";
    this.alias = "";
    this.HP = 0;
    this.attack = 0;
    this.defense = 0;
    this.status = false;
    this.color = "#cdcdcd";
    this.armyId = 0;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getAlias() {
    return alias;
}

public void setAlias(String alias) {
    this.alias = alias;
}

public int getHP() {
    return HP;
}
```

```
public void setHP(int HP) {
    this.HP = HP;
}

public int getAttack() {
    return attack;
}

public void setAttack(int attack) {
    this.attack = attack;
}

public int getDefense() {
    return defense;
}

public void setDefense(int defense) {
    this.defense = defense;
}

public int getX() {
    return x;
}

public String getColor() {
    return color;
}

public void setColor(String color) {
    this.color = color;
}

public int getArmyId() {
    return armyId;
}

public void setArmyId(int armyId) {
    this.armyId = armyId;
}

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

public int getY() {
    return y;
}

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

public boolean getStatus() {
    return status;
}
```

```
    public void setStatus(boolean status) {
        this.status = status;
    }
}
class Archer extends Soldier{
    private int arrows;
    Random random = new Random();

    public Archer(String name) {
        super(name);
        this.arrows = random.nextInt(10);
        this.setAttack(7);
        this.setDefense(3);
        this.setHP(random.nextInt(3) + 3);
    }

    public int getArrows() {
        return arrows;
    }

    public void setArrows(int arrows) {
        this.arrows = arrows;
    }

    //Shoots one arrow to an enemy
    @Override
    public void action(){
        this.arrows--;
    }
}
class Knight extends Soldier{
    private String weapon;
    private boolean isMounted;
    Random random = new Random();

    public Knight(String name) {
        super(name);
        this.setAttack(13);
        this.setDefense(7);
        this.setHP(random.nextInt(3) + 10);
    }

    public String getWeapon() {
        return weapon;
    }

    public void setWeapon(String weapon) {
        this.weapon = weapon;
    }

    public boolean isMounted() {
        return isMounted;
    }

    public void setMounted(boolean mounted) {
        isMounted = mounted;
```

```
    }

    //If it's unmounted, it will mount and its weapon will change to a Sword, if not,
    // it will unmount and its weapon will change to a Spear
    @Override
    public void action(){
        this.isMounted = !this.isMounted;

        if(this.isMounted){
            this.weapon = "Sword";
        }else{
            this.weapon = "Spear";
        }
    }
}
class Swordsman extends Soldier{
    private final int swordLong;
    Random random = new Random();

    public Swordsman(String name) {
        super(name);
        this.swordLong = random.nextInt(10);
        this.setAttack(10);
        this.setDefense(8);
        this.setHP(random.nextInt(3) + 8);
    }

    public int getSwordLong() {
        return swordLong;
    }

    //This will set a wall of shields, and increase its defense by 1
    @Override
    public void action(){
        this.setDefense(this.getDefense() + 1);
    }
}
class Spearman extends Soldier{
    private final int spearLong;
    Random random = new Random();

    public Spearman(String name) {
        super(name);
        this.spearLong = random.nextInt(10);
        this.setAttack(5);
        this.setDefense(8);
        this.setHP(random.nextInt(3) + 8);
    }

    public int getSpearLong() {
        return spearLong;
    }

    //This will make a schiltrom and will increase its defense by 1
    @Override
    public void action(){
```

```
        this.setDefense(this.getDefense() + 1);
    }
}
```

- Es la misma clase de los laboratorios anteriores, solo que con pequeñas adaptaciones.
- Igual con las clases Army y Board (Mapa).

Listing 2: Clase Army

```
package com.example.lab22;
import java.util.*;
public class Army {
    private List<Soldier> soldiers = new ArrayList<>();
    private String[] realms = new String[]{
            "Inglaterra", "Francia", "Castilla-Aragon", "Moros", "Sacro Imperio Romano
                Germanico"
    };
    private String kingdom;
    private int id;
    private int nArchers = 0;
    private int nKnights = 0;
    private int nSwords = 0;
    private int nSpears = 0;


    Random random = new Random();

    public Army(int id) {
        this.kingdom = realms[random.nextInt(realms.length)];
        this.id = id;
        this.initialize();

    }
    public void initialize(){
        for (int i = 0; i < random.nextInt(10) + 1; i++) {
            switch (random.nextInt(4) + 1) {
                case 1:
                    Archer a = new Archer("Arquero" + i + "X" + id);
                    a.setAlias("A" + i + "X" + id);
                    a.setArmyId(id);
                    if(id == 1){
                        a.setColor("#b81414");
                    }else{
                        a.setColor("#00ffff");
                    }
                    nArchers++;
                    soldiers.add(a);
                    break;
                case 2:
                    Knight k = new Knight("Caballero" + i + "X" + id);
                    k.setAlias("C" + i + "X" + id);
                    k.setArmyId(id);
                    if(id == 1){
                        k.setColor("#b81414");
                    }else{
```

```
                    k.setColor("#00ffff");
                }
                nKnights++;
                soldiers.add(k);
                break;
            case 3:
                Spearman s = new Spearman("Lancero" + i + "X" + id);
                s.setAlias("L" + i + "X" + id);
                s.setArmyId(id);
                if(id == 1){
                    s.setColor("#b81414");
                }else{
                    s.setColor("#00ffff");
                }
                nSpears++;
                soldiers.add(s);
                break;
            case 4:
                Swordsman w = new Swordsman("Espadachin" + i + "X" + id);
                w.setAlias("E" + i + "X" + id);
                w.setArmyId(id);
                if(id == 1){
                    w.setColor("#b81414");
                }else{
                    w.setColor("#00ffff");
                }
                nSwords++;
                soldiers.add(w);
                break;
        }
    }
}
public void perk() {
    for (Soldier s : soldiers) {
        s.setHP(s.getHP() + 1);
    }
}

public List<Soldier> getSoldiers() {
    return soldiers;
}

public void setSoldiers(List<Soldier> soldiers) {
    this.soldiers = soldiers;
}

public String[] getRealms() {
    return realms;
}

public void setRealms(String[] realms) {
    this.realms = realms;
}

public String getKingdom() {
    return kingdom;
```

```
367     }
368
369     public void setKingdom(String kingdom) {
370         this.kingdom = kingdom;
371     }
372
373     public int getId() {
374         return id;
375     }
376
377     public void setId(int id) {
378         this.id = id;
379     }
380
381     public int getnArchers() {
382         return nArchers;
383     }
384
385     public void setnArchers(int nArchers) {
386         this.nArchers = nArchers;
387     }
388
389     public int getnKnights() {
390         return nKnights;
391     }
392
393     public void setnKnights(int nKnights) {
394         this.nKnights = nKnights;
395     }
396
397     public int getnSwords() {
398         return nSwords;
399     }
400
401     public void setnSwords(int nSwords) {
402         this.nSwords = nSwords;
403     }
404
405     public int getnSpears() {
406         return nSpears;
407     }
408
409     public void setnSpears(int nSpears) {
410         this.nSpears = nSpears;
411     }
412 }
```

Listing 3: Clase Board

```
414 package com.example.lab22;
415 import java.util.*;
416
417 public class Board {
418     private Soldier[][] table = new Soldier[10][10];
419
420     private String[] territories = new String[] {
```

```
421             "Bosque", "Campo Abierto", "Montana", "Desierto", "Playa" };
422     private String territory;
423     private Map<String, String> perks = new HashMap<>();
424     Random random = new Random();
425     public Board() {
426         this.territory = territories[random.nextInt(territories.length)];
427     }
428     public void initializeTable(){
429         for(int i = 0; i < table.length; i++){
430             for(int j = 0; j < table[i].length; j++){
431                 table[i][j] = new Soldier();
432             }
433         }
434     }
435     public void initializeArmy(Army a){
436         for(Soldier s: a.getSoldiers()){
437             int x = 0;
438             int y = 0;
439             do{
440                 x = random.nextInt(9);
441                 y = random.nextInt(9);
442             }while(table[y][x].getStatus());
443             s.setY(y);
444             s.setX(x);
445             table[y][x] = s;
446         }
447     }
448 
449     public Soldier[][] getTable() {
450         return table;
451     }
452 
453     public void setTable(Soldier[][] table) {
454         this.table = table;
455     }
456 
457     public String[] getTerritories() {
458         return territories;
459     }
460 
461     public void setTerritories(String[] territories) {
462         this.territories = territories;
463     }
464 
465     public String getTerritory() {
466         return territory;
467     }
468 
469     public void setTerritory(String territory) {
470         this.territory = territory;
471     }
472 
473     public Map<String, String> getPerks() {
474         return perks;
475     }
476 
```

```
    public void setPerks(Map<String, String> perks) {
        this.perks = perks;
    }
}
```

- Ahora si viene lo interesante, crear y aplicar la interfaz gráfica de usuario, para esto utilizaré JavaFX, usando IntelliJ IDEA como IDE, y SceneBuilder para realizar esto.
- Para comenzar el proyecto, creo una clase que implemente la interfaz gráfica en base a un archivo fxml, una clase que herede a la clase Application de JavaFx.

Listing 4: Clase HelloApplication

```
package com.example.lab22;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.stage.Stage;

import java.io.IOException;

public class HelloApplication extends Application {


    @Override
    public void start(Stage stage) throws IOException {
        FXMLLoader fxmlLoader = new
            FXMLLoader(HelloApplication.class.getResource("hello-view.fxml"));
        Scene scene = new Scene(fxmlLoader.load(), 670, 670);
        stage.setTitle("VideoJuego");
        stage.setScene(scene);
        stage.show();
    }

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

- Esta clase implementa la interfaz gráfica en base a un fichero fxml, en el cual desarrollo lo que se va a mostrar al usuario.
- Para esta interfaz, decidí implementar 100 botones que representen un tablero de 10x10 usando el contenedor GridPane, para poder llevar una cuenta de las coordenadas de cada botón, además de una barra arriba del tablero para cambiar de turno, minimizar, cerrar o maximizar.

Listing 5: Archivo FXML

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.layout.ColumnConstraints?>
```

```
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.RowConstraints?>
<?import javafx.scene.layout.VBox?>

<VBox fx:id="rootVBox" prefHeight="400.0" prefWidth="666.0" spacing="20.0" styleClass="vbox"
    stylesheets="@style.css" xmlns="http://javafx.com/javafx/21"
    xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.lab22.HelloController">
   <padding>
      <Insets bottom="20.0" left="20.0" right="20.0" top="20.0" />
   </padding>
  <children>
     <Pane prefHeight="200.0" prefWidth="200.0" styleClass="toolBar">
        <children>
           <Pane fx:id="closeButton" layoutX="580.0" layoutY="-12.0"
               onMouseClicked="#handleCloseButton" />
            <Pane fx:id="minimizeButton" layoutX="520.0" layoutY="-12.0"
                onMouseClicked="#handleMinimizeButton" />
            <Pane fx:id="expandButton" layoutX="550.0" layoutY="-12.0"
                onMouseClicked="#handleExpandButton" />
            <Pane fx:id="nextButton" layoutX="15.0" layoutY="-12.0"
                onMouseClicked="#changeTurn" />
            <Pane fx:id="changeButton" />
        </children>
     </Pane>


     <GridPane id="board" fx:id="gridPane" prefHeight="330.0" prefWidth="352.0">
         <columnConstraints>
             <ColumnConstraints percentWidth="10" />
             <ColumnConstraints percentWidth="10" />
             <ColumnConstraints percentWidth="10" />
             <ColumnConstraints percentWidth="10" />
             <ColumnConstraints percentWidth="10" />
             <ColumnConstraints percentWidth="10" />
             <ColumnConstraints percentWidth="10" />
             <ColumnConstraints percentWidth="10" />
             <ColumnConstraints percentWidth="10" />
             <ColumnConstraints percentWidth="10" />
         </columnConstraints>
         <rowConstraints>
             <RowConstraints percentHeight="10" vgrow="SOMETIMES" />
             <RowConstraints percentHeight="10" vgrow="SOMETIMES" />
             <RowConstraints percentHeight="10" vgrow="SOMETIMES" />
             <RowConstraints percentHeight="10" vgrow="SOMETIMES" />
             <RowConstraints percentHeight="10" vgrow="SOMETIMES" />
             <RowConstraints percentHeight="10" vgrow="SOMETIMES" />
             <RowConstraints percentHeight="10" vgrow="SOMETIMES" />
             <RowConstraints percentHeight="10" vgrow="SOMETIMES" />
             <RowConstraints percentHeight="10" vgrow="SOMETIMES" />
             <RowConstraints percentHeight="10" vgrow="SOMETIMES" />
         </rowConstraints>
        <children>

            <!-- Row 0 -->
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
```

```
            styleClass="button" text="" GridPane.columnIndex="0" GridPane.rowIndex="0" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="1" GridPane.rowIndex="0" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="2" GridPane.rowIndex="0" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="3" GridPane.rowIndex="0" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="4" GridPane.rowIndex="0" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="5" GridPane.rowIndex="0" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="6" GridPane.rowIndex="0" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="7" GridPane.rowIndex="0" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="8" GridPane.rowIndex="0" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="9" GridPane.rowIndex="0" />

        <!-- Row 1 -->
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="0" GridPane.rowIndex="1" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="1" GridPane.rowIndex="1" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="2" GridPane.rowIndex="1" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="3" GridPane.rowIndex="1" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="4" GridPane.rowIndex="1" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="5" GridPane.rowIndex="1" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="6" GridPane.rowIndex="1" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="7" GridPane.rowIndex="1" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="8" GridPane.rowIndex="1" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="9" GridPane.rowIndex="1" />

        <!-- Row 2 -->
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="0" GridPane.rowIndex="2" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="1" GridPane.rowIndex="2" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="2" GridPane.rowIndex="2" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="3" GridPane.rowIndex="2" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="4" GridPane.rowIndex="2" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="5" GridPane.rowIndex="2" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
```

```
            styleClass="button" text="" GridPane.columnIndex="6" GridPane.rowIndex="2" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="7" GridPane.rowIndex="2" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="8" GridPane.rowIndex="2" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="9" GridPane.rowIndex="2" />

        <!-- Row 3 -->
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="0" GridPane.rowIndex="3" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="1" GridPane.rowIndex="3" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="2" GridPane.rowIndex="3" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="3" GridPane.rowIndex="3" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="4" GridPane.rowIndex="3" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="5" GridPane.rowIndex="3" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="6" GridPane.rowIndex="3" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="7" GridPane.rowIndex="3" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="8" GridPane.rowIndex="3" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="9" GridPane.rowIndex="3" />

        <!-- Row 4 -->
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="0" GridPane.rowIndex="4" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="1" GridPane.rowIndex="4" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="2" GridPane.rowIndex="4" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="3" GridPane.rowIndex="4" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="4" GridPane.rowIndex="4" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="5" GridPane.rowIndex="4" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="6" GridPane.rowIndex="4" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="7" GridPane.rowIndex="4" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="8" GridPane.rowIndex="4" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="9" GridPane.rowIndex="4" />

        <!-- Row 5 -->
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
            styleClass="button" text="" GridPane.columnIndex="0" GridPane.rowIndex="5" />
        <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
```

```
                styleClass="button" text="" GridPane.columnIndex="1" GridPane.rowIndex="5" />
625             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="2" GridPane.rowIndex="5" />
626             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="3" GridPane.rowIndex="5" />
627             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="4" GridPane.rowIndex="5" />
628             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="5" GridPane.rowIndex="5" />
629             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="6" GridPane.rowIndex="5" />
630             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="7" GridPane.rowIndex="5" />
631             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="8" GridPane.rowIndex="5" />
632             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="9" GridPane.rowIndex="5" />
633
634             <!-- Row 6 -->
635             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="0" GridPane.rowIndex="6" />
636             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="1" GridPane.rowIndex="6" />
637             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="2" GridPane.rowIndex="6" />
638             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="3" GridPane.rowIndex="6" />
639             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="4" GridPane.rowIndex="6" />
640             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="5" GridPane.rowIndex="6" />
641             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="6" GridPane.rowIndex="6" />
642             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="7" GridPane.rowIndex="6" />
643             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="8" GridPane.rowIndex="6" />
644             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="9" GridPane.rowIndex="6" />
645
646             <!-- Row 7 -->
647             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="0" GridPane.rowIndex="7" />
648             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="1" GridPane.rowIndex="7" />
649             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="2" GridPane.rowIndex="7" />
650             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="3" GridPane.rowIndex="7" />
651             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="4" GridPane.rowIndex="7" />
652             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="5" GridPane.rowIndex="7" />
653             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="6" GridPane.rowIndex="7" />
654             <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
```

```
                styleClass="button" text="" GridPane.columnIndex="7" GridPane.rowIndex="7" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="8" GridPane.rowIndex="7" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="9" GridPane.rowIndex="7" />

            <!-- Row 8 -->
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="0" GridPane.rowIndex="8" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="1" GridPane.rowIndex="8" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="2" GridPane.rowIndex="8" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="3" GridPane.rowIndex="8" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="4" GridPane.rowIndex="8" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="5" GridPane.rowIndex="8" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="6" GridPane.rowIndex="8" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="7" GridPane.rowIndex="8" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="8" GridPane.rowIndex="8" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="9" GridPane.rowIndex="8" />

            <!-- Row 9 -->
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="0" GridPane.rowIndex="9" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="1" GridPane.rowIndex="9" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="2" GridPane.rowIndex="9" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="3" GridPane.rowIndex="9" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="4" GridPane.rowIndex="9" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="5" GridPane.rowIndex="9" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="6" GridPane.rowIndex="9" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="7" GridPane.rowIndex="9" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="8" GridPane.rowIndex="9" />
            <Button mnemonicParsing="false" onMouseClicked="#onButtonClick"
                styleClass="button" text="" GridPane.columnIndex="9" GridPane.rowIndex="9" />


        </children>
      </GridPane>
    </children>
</VBox>
```

- Esto con el fin de asignar una función llamada onButtonClick, que implementaré mas adelante, a cada botón, para
- Por último, programé el resumen final, donde se dice cuál de los dos ejércitos es el ganador, basado en probabilidad según la cantidad de vida total por ejército. Eso es todo. Al final, solo me quedaba armar todo en la clase principal, sin embargo, se iba a ver algo feo, así que apliqué una hoja de estilos css para que se vea mucho más agradable a la vista.

Listing 6: Style.css

```
/* Estilos para el VBox (contenedor principal) */
.vbox {
    -fx-spacing: 20px;
    -fx-background-color: #000;
}

/* Estilos para el GridPane */
#board {
    -fx-padding: 10px;
    -fx-border-color: #000;
    -fx-border-width: 2px;
}

/* Estilos para los botones */
.button {
    -fx-min-width: 60px;
    -fx-min-height: 60px;
    -fx-max-width: 60px;
    -fx-max-height: 60px;
    -fx-background-color: #fff; /* Color de fondo blanco por defecto */
    -fx-font-size: 14px;
    -fx-font-weight: bold;
    -fx-border-color: #000;
    -fx-border-width: 2px;
}
#closeButton{
    -fx-min-width: 30px;
    -fx-min-height: 30px;
    -fx-background-color: transparent;
    -fx-background-image: url('./img/x-circle-regular-24.png');
    -fx-background-repeat: no-repeat;
    -fx-background-size: cover;
}
#minimizeButton{
    -fx-min-width: 30px;
    -fx-min-height: 30px;
    -fx-background-color: transparent;
    -fx-background-image: url('./img/minus-regular-24.png');
    -fx-background-repeat: no-repeat;
    -fx-background-size: cover;
}
#expandButton{
    -fx-min-width: 30px;
    -fx-min-height: 30px;
    -fx-background-color: transparent;
    -fx-background-image: url('./img/expand-alt-regular-24.png');
```

```
    -fx-background-repeat: no-repeat;
    -fx-background-size: cover;
}
#nextButton{
    -fx-min-width: 30px;
        -fx-min-height: 30px;
        -fx-background-color: transparent;
        -fx-background-image: url('./img/skip-next-circle-regular-24.png');
        -fx-background-repeat: no-repeat;
        -fx-background-size: cover;
}
```

- Con esto, solo me falta implementar la clase principal controladora para manejar el videojuego.

Listing 7: Clase controladora

```
   package com.example.lab22;

import javafx.animation.Animation;
import javafx.animation.KeyFrame;
import javafx.animation.Timeline;
import javafx.application.Platform;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.Node;
import javafx.scene.control.*;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
import javafx.util.Duration;

import java.util.Optional;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.Semaphore;

public class HelloController {
    @FXML
    private GridPane gridPane = new GridPane();

    private Army a1 = new Army(1);
    private Army a2 = new Army(2);
    private Board map = new Board();
    private static int actualTurn = 1;
    private int nEj1 = a1.getSoldiers().size();
    private int nEj2 = a2.getSoldiers().size();
    private boolean gameOver = false;
    private CountDownLatch buttonClickLatch = new CountDownLatch(1);

    @FXML
    private VBox rootVBox;


    @FXML
    public void initialize() {
```

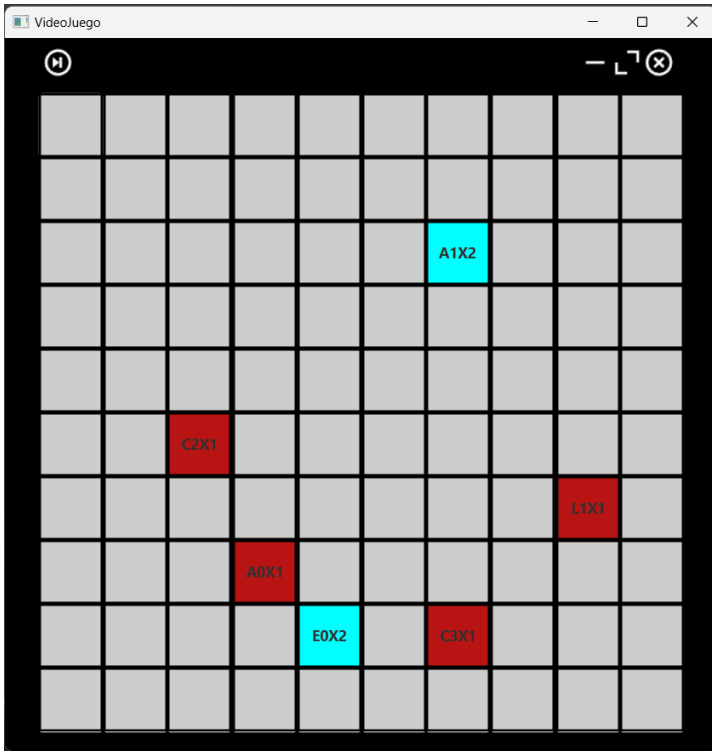
```
787         initializeBoard();
788         runGame();
789     }
790 
791     public void runGame() {
792         Timeline timeline = new Timeline(new KeyFrame(Duration.seconds(1), event -> {
793             if (!gameOver) {
794                 gameOver();
795                 placeSoldiers();
796 
797             }
798         }));
799         timeline.setCycleCount(Timeline.INDEFINITE);
800         timeline.play();
801 
802     }
803     private void initializeBoard(){
804         map.initializeTable();
805         map.initializeArmy(a1);
806         map.initializeArmy(a2);
807     }
808 
809     private void placeSoldiers() {
810 
811         int rowCount = 1;
812         int colCount = 1;
813 
814         for (javafx.scene.Node node : gridPane.getChildren()) {
815             Soldier s = map.getTable()[rowCount - 1][colCount - 1];
816             String color = s.getColor();
817             String n = s.getAlias();
818             if (node instanceof Button) {
819                 Button button = (Button) node;
820                 String buttonName = n;
821                 button.setText(buttonName);
822                 button.setStyle("-fx-background-color: " + color);
823                 colCount++;
824 
825                 if (colCount > 10) {
826                     colCount = 1;
827                     rowCount++;
828                 }
829             }
830         }
831     }
832     public void onButtonClick(MouseEvent event){
833         if (event.getSource() instanceof Button) {
834             Button clickedButton = (Button) event.getSource();
835             int columnIndex = GridPane.getColumnIndex(clickedButton);
836             int rowIndex = GridPane.getRowIndex(clickedButton);
837             Soldier s = map.getTable()[rowIndex][columnIndex];
838             if(s.getStatus()){
839                 if(s.getArmyId() == actualTurn){
840                     String res = showInputAlert(s);
841                     if(res == null){
842                         return;
```

```
843                 }
844
845                 int x = Integer.parseInt(res.split(", ")[0]);
846                 int y = Integer.parseInt(res.split(", ")[1]);
847                 Soldier s2 = map.getTable()[y][x];
848
849                 if((Math.abs(columnIndex - x) > 1) || (Math.abs(rowIndex - y) > 1)){
850                     showAlert("Casilla invalida (Solo puedes moverte una casilla en cada
                         direccion)", "Error");
851                 }else{
852                     if(s2.getStatus()){
853                         if(s2.getArmyId() == s.getArmyId()){
854                             showAlert("No se permite el fuego aliado", "Error");
855                         }else {
856                             battle(s, s2);
857                         }
858                     }else{
859                         move(s, s2);
860                     }
861                 }
862             }else{
863                 showAlert("Elige un soldado de tu propio Ejercito", "Error");
864             }
865         }
866
867     }
868 }
869
870 private String showInputAlert(Soldier s){
871     String text = s.getName() + " " + s.getHP() + " HP";
872     TextInputDialog dialog = new TextInputDialog(null);
873     dialog.setTitle("Input Dialog");
874     dialog.setHeaderText(text + " ha sido seleccionado");
875     dialog.setContentText("Coordenadas de destino (x, y): ");
876     Optional<String> result = dialog.showAndWait();
877     return result.orElse(null);
878 }
879 private static void showAlert(String contentText, String title) {
880     Alert alert = new Alert(Alert.AlertType.WARNING);
881     alert.setTitle(title);
882     alert.setHeaderText(null);
883     alert.setContentText(contentText);
884     alert.showAndWait();
885 }
886 private void battle(Soldier s1, Soldier s2){
887     Soldier winner = Soldier.winner(s1, s2);
888
889     if(winner.getArmyId() == 1){
890         nEj2--;
891     }else{
892         nEj1--;
893     }
894
895
896     Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
897     alert.setTitle("Batalla");
```

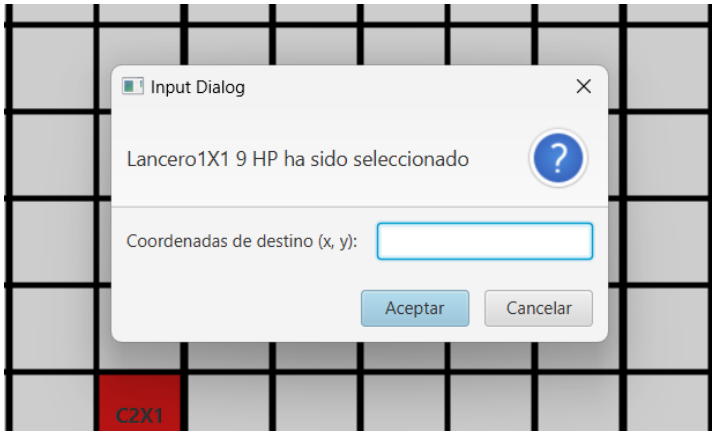
```
898         alert.setHeaderText(null);
899         alert.setContentText("Los dos soldados batallaron, el ganador fue: " +
                winner.getName());
900
901         alert.showAndWait();
902
903         s2.copy(winner);
904         s1.destroy();
905
906     }
907     public static void move(Soldier s1, Soldier s2){
908         s2.copy(s1);
909         s1.destroy();
910     }
911     public void gameOver(){
912             if (nEj1 == 0 || nEj2 == 0) {
913                 gameOver = true;
914                 Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
915                 alert.setTitle("Juego Terminado");
916                 alert.setHeaderText(null);
917                 alert.setContentText("El ganador es el Ejercito " + (nEj1 == 0 ? 2 : 1));
918                 alert.showAndWait();
919                 // Cerrar la aplicacion despues de mostrar el mensaje de juego terminado
920                 Platform.exit();
921             }
922     }
923     public void changeTurn(MouseEvent event) {
924         Alert alerta = new Alert(Alert.AlertType.CONFIRMATION);
925         alerta.setTitle("Confirmacion");
926         alerta.setHeaderText("Quieres terminar tu turno?");
927         alerta.setContentText("Por favor, elige una opcion.");
928
929         // Personalizar los botones de la alerta
930         ButtonType botonSi = new ButtonType("Si");
931         ButtonType botonNo = new ButtonType("No");
932         alerta.getButtonTypes().setAll(botonSi, botonNo);
933
934         // Mostrar la alerta y esperar a que el usuario elija una opcion
935         alerta.showAndWait().ifPresent(resultado -> {
936             if (resultado == botonSi) {
937                 actualTurn = (actualTurn == 1) ? 2 : 1;
938             }
939         });
940
941     }
942
943     // Metodo para cerrar la ventana
944     @FXML
945     private void handleCloseButton(MouseEvent event) {
946         Stage stage = (Stage) rootVBox.getScene().getWindow();
947         stage.close();
948     }
949
950     // Metodo para minimizar la ventana
951     @FXML
952     private void handleMinimizeButton(MouseEvent event) {
```

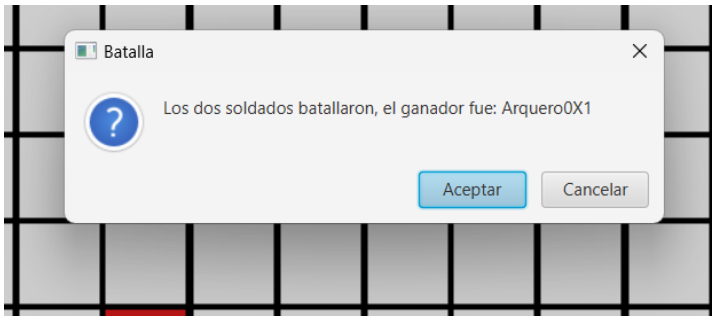
```
953         Stage stage = (Stage) rootVBox.getScene().getWindow();
954         stage.setIconified(true);
955     }
956 
957     // Metodo para expandir/ restaurar la ventana
958     @FXML
959     private void handleExpandButton(MouseEvent event) {
960         Stage stage = (Stage) rootVBox.getScene().getWindow();
961         if (stage.isMaximized()) {
962             stage.setMaximized(false);
963         } else {
964             stage.setMaximized(true);
965         }
966     }
967 
968 }
```

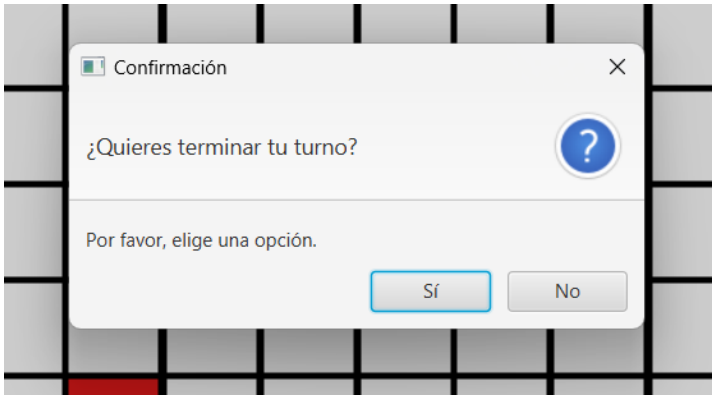
- En este código, tengo el método onButtonClick, que lo que hace es primero buscar si hay un soldado en esa posición, (usando de referencia el mapa) una vez que detecta un soldado, revisa si es del ejército respectivo al turno actual (si es turno del ejercito 1 o 2) para que solo puedas mover soldados cuando te toca a ti.
- Luego de eso, pregunto al usuario a que coordenadas lo quiere mover, revisando que sea una coordenada a 1 casilla de distancia.
- Luego de eso ve si hay un soldado en la casilla de destino y si lo hay, revisa que no sea del mismo ejército. Si no lo es realiza una batalla y coloca ahí al ganador, para terminar.
- Esa es la lógica principal, también está el método runGame que trabaja en un hilo, constantemente refrescando los gráficos. Y el resto de métodos son simplemente para que la interfaz funcione, como los métodos para minimizar o cerrar usando los botones de la barra de tareas.
- También está el método para determinar si el juego se ha terminado o no y se cierra la pestaña.
- Al final el tablero queda de esta forma al iniciar el juego.



- Y con algunas capturas del funcionamiento del videojuego.







- Estos cambios fueron subidos al respositorio, a través de commits.

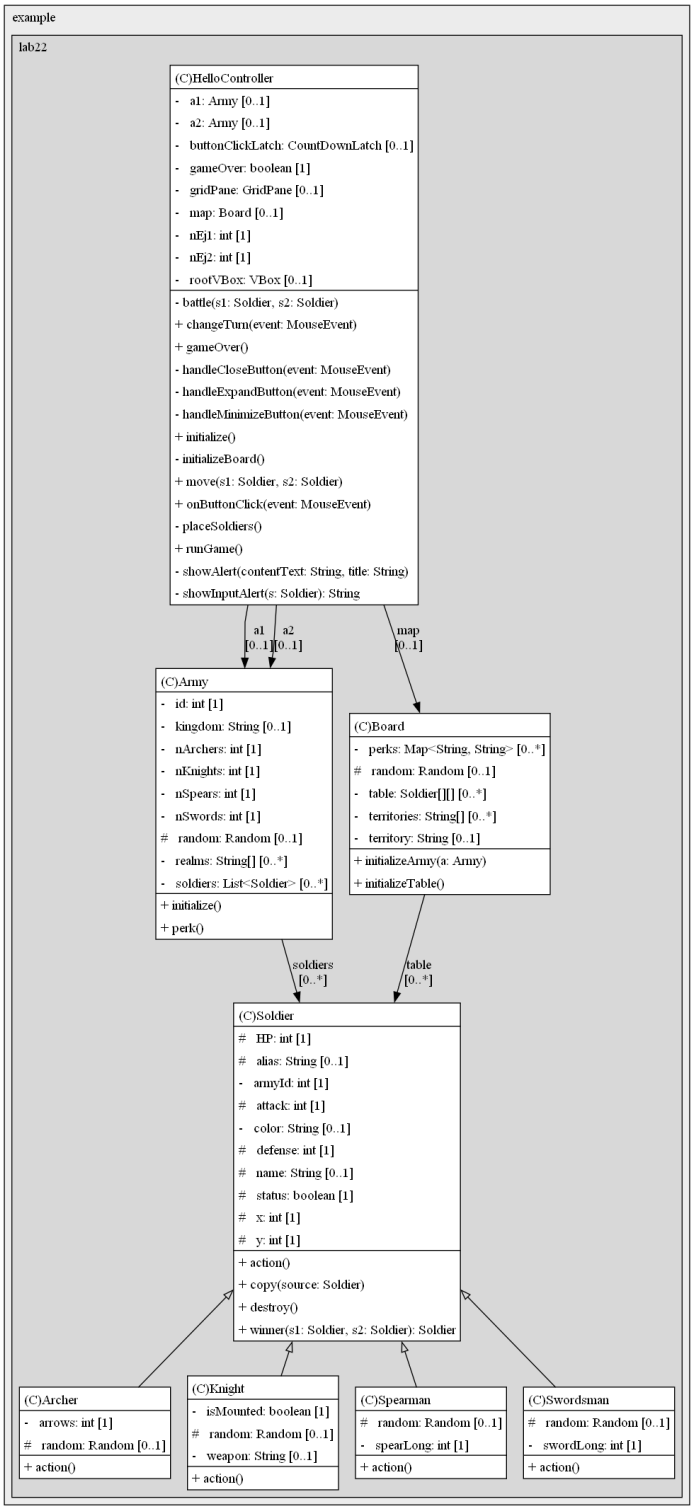
Listing 8: Commit

```
969   $commit feec5bf606898afbaaaadd1a80b1f0c0c356a83f (HEAD -> main, origin/main)
970 Author: RYAN VALDIVIA <rvaldiviase@unsa.edu.pe>
971 Date:   Sat Jan 20 11:48:31 2024 -0500
972
973     Estableciendo las clases necesarias y realizando pruebas de JavaFx, usando IntelliJ Idea
```

Listing 9: Commit

```
974   $commit 44f6eaa51af046b665943d901d3ac65b04040216 (HEAD -> main, origin/main)
975 Author: RYAN VALDIVIA <rvaldiviase@unsa.edu.pe>
976 Date:  Mon Jan 22 13:38:31 2024 -0500
977
978    Trabajo terminado, el juego es 100% funcional
```

- Además, aquí está el diagrama UML de todo el proyecto.


example
lab22
(C)HelloController
- a1: Army [0..1]
- a2: Army [0..1]
- buttonClickLatch: CountDownLatch [0..1]
- gameOver: boolean [1]
- gridPane: GridPane [0..1]
- map: Board [0..1]
- nEj1: int [1]
- nEj2: int [1]
- rootVBox: VBox [0..1]
- battle(s1: Soldier, s2: Soldier)
+ changeTurn(event: MouseEvent)
+ gameOver()
- handleCloseButton(event: MouseEvent)
- handleExpandButton(event: MouseEvent)
- handleMinimizeButton(event: MouseEvent)
+ initialize()
- initializeBoard()
+ move(s1: Soldier, s2: Soldier)
+ onButtonClick(event: MouseEvent)
- placeSoldiers()
+ runGame()
- showAlert(contentText: String, title: String)
- showInputAlert(s: Soldier): String
a1 [0..1]
a2 [0..1]
map [0..1]
(C)Army
- id: int [1]
- kingdom: String [0..1]
- nArchers: int [1]
- nKnights: int [1]
- nSpears: int [1]
- nSwords: int [1]
random: Random [0..1]
- realms: String[] [0..*]
- soldiers: List<Soldier> [0..*]
+ initialize()
+ perk()
(C)Board
- perks: Map<String, String> [0..*]
random: Random [0..1]
- table: Soldier[][] [0..*]
- territories: String[] [0..*]
- territory: String [0..1]
+ initializeArmy(a: Army)
+ initializeTable()
soldiers [0..*]
table [0..*]
(C)Soldier
HP: int [1]
alias: String [0..1]
- armyId: int [1]
attack: int [1]
- color: String [0..1]
defense: int [1]
name: String [0..1]
status: boolean [1]
x: int [1]
y: int [1]
+ action()
+ copy(source: Soldier)
+ destroy()
+ winner(s1: Soldier, s2: Soldier): Soldier
(C)Archer
- arrows: int [1]
random: Random [0..1]
+ action()
(C)Knight
- isMounted: boolean [1]
random: Random [0..1]
- weapon: String [0..1]
+ action()
(C)Spearman
random: Random [0..1]
- spearLong: int [1]
+ action()
(C)Swordsman
random: Random [0..1]
- swordLong: int [1]
+ action()

5. Rúbricas

5.1. Entregable Informe

Tabla 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y facil de leer.

5.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumplio con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos lo items.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 2: Niveles de desempeño

	Nivel			
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 3: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	2	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente estan dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	4	
Total		20		18	

6. Referencias

- Fundamentos de la programación 2 - Tópicos de la programación Orientada a Objetos (Marco Aedo)