Бруцкая Анастасия, 251001

package com.example.lapa12.heros;

import java.util.ArrayList;

import java.util.List;

public class Hilichurls {

public List<Hilichurl> hilichurls = new ArrayList<>();

}

package com.example.lapa12.heros;

import com.fasterxml.jackson.annotation.JsonIgnore;

import com.fasterxml.jackson.annotation.JsonSubTypes;

import com.fasterxml.jackson.annotation.JsonTypeInfo;

import javafx.scene.image.Image;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.Serializable;

@JsonTypeInfo(use = JsonTypeInfo.Id.NAME, property="@type")

@JsonSubTypes({

@JsonSubTypes.Type(value=HilichurlFighter.class, name="fighter"),

@JsonSubTypes.Type(value=HilichurlGrenadier.class, name="grenadier"),

@JsonSubTypes.Type(value=HilichurlGuard.class, name="guard"),

@JsonSubTypes.Type(value=HilichurlShooter.class, name="shooter"),

@JsonSubTypes.Type(value=Lawachurl.class, name="lawachurl"),

@JsonSubTypes.Type(value=Mitachurl.class, name="mitachurl")

})

public class Hilichurl implements Serializable {

private int level;

private int XP;

int maxXP;

private String name;

@JsonIgnore

transient private Image image;

private String imagePath;

private double x;

private double y;

public Hilichurl(int level) throws FileNotFoundException {

this.level = level;

if (level <= 0) {

this.maxXP = 100;

} else if (level > 0 && level < 30) {

this.maxXP = 1000;

} else if (level >= 30 && level < 70) {

this.maxXP = 5000;

} else if (level >= 70 && level < 100) {

this.maxXP = 10000;

} else {

this.maxXP = 15000;

}

this.XP= this.maxXP;

this.name = "Hilichurl";

this.imagePath = "src/main/java/com/example/lapa12/images/"+ this.name +".jpg";

this.image = new Image(new FileInputStream(imagePath), 150, 140, true, true);

}

public Hilichurl() {

}

public String getImagePath() {

return imagePath;

}

public void setImagePath(String imagePath) {

this.imagePath = imagePath;

}

public double getX() {

return x;

}

public void setX(double x) {

this.x = x;

}

public double getY() {

return y;

}

public void setY(double y) {

this.y = y;

}

public String printInfo() {

return "This is " + this.getName() + " of " + this.getLevel() + " level with " + this.getXP() + " XP";

}

public void recovery(){

this.XP=this.maxXP;

}

public int getLevel() {

return level;

}

public void setLevel(int level) {

this.level = level;

}

public int getXP() {

return XP;

}

public Image getImage() {

return image;

}

public void setImage(Image image) {

this.image = image;

}

public void setXP(int XP) {

this.XP = XP;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

@Override

public String toString() {

return "Hilichurl{" +

"level=" + level +

", XP=" + XP +

", maxXP=" + maxXP +

", name='" + name + '\'' +

", imagePath='" + imagePath + '\'' +

", x=" + x +

", y=" + y +

'}';

}

}

package com.example.lapa12.heros;

import com.example.lapa12.Element;

import javafx.scene.image.Image;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.Serializable;

public class HilichurlFighter extends Hilichurl implements Serializable {

private Element club;

public HilichurlFighter() {

}

@Override

public String

toString() {

return "HilichurlFighter{" +

"club=" + club +

"} " + super.toString();

}

public HilichurlFighter(int level, Element club) throws FileNotFoundException {

super(level);

this.club = club;

this.setName("Fighter");

this.setImagePath("src/main/java/com/example/lapa12/images/" + this.getName() + ".jpg");

this.setImage(new Image(new FileInputStream(getImagePath()), 150, 140, true, true));

}

public Element getClub() {

return club;

}

public void setClub(Element club) {

this.club = club;

}

@Override

public String printInfo(){

return super.printInfo() +"with " + this.getClub() + " club ";

}

}

package com.example.lapa12.heros;

import com.example.lapa12.Element;

import javafx.scene.image.Image;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.Serializable;

public class HilichurlGrenadier extends Hilichurl implements Serializable {

private Element slime;

public HilichurlGrenadier() {

}

public HilichurlGrenadier(int level, Element slime) throws FileNotFoundException {

super(level);

this.slime = slime;

this.setName("Grenadier");

this.setImagePath("src/main/java/com/example/lapa12/images/" + this.getName() + ".jpg");

this.setImage(new Image(new FileInputStream(getImagePath()), 150, 140, true, true));

}

public Element getSlime() {

return slime;

}

public void setSlime(Element slime) {

this.slime = slime;

}

@Override

public String printInfo(){

return super.printInfo()+ " with " + this.getSlime()+ " slime";

}

@Override

public String toString() {

return "HilichurlGrenadier{" +

"slime=" + slime +

"} " + super.toString();

}

}

package com.example.lapa12.heros;

import com.example.lapa12.Element;

import javafx.scene.image.Image;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.Serializable;

public class HilichurlGuard extends HilichurlFighter implements Serializable {

private Element shield;

public HilichurlGuard() {

}

public HilichurlGuard(int level, Element club, Element shield) throws FileNotFoundException {

super(level, club);

this.shield = shield;

this.setName("Guard");

this.setImagePath("src/main/java/com/example/lapa12/images/" + this.getName() + ".jpg");

this.setImage(new Image(new FileInputStream(getImagePath()), 150, 140, true, true));

}

public Element getShield() {

return shield;

}

public void setShield(Element shield) {

this.shield = shield;

}

@Override

public String printInfo(){

return super.printInfo() + " and " + this.getShield() + " shield";

}

@Override

public String toString() {

return "HilichurlGuard{" +

"shield=" + shield +

"} " + super.toString();

}

}

package com.example.lapa12.heros;

import com.example.lapa12.Element;

import javafx.scene.image.Image;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.Serializable;

public class HilichurlShooter extends Hilichurl implements Serializable {

private Element crossbow;

public HilichurlShooter() {

}

public HilichurlShooter(int level, Element crossbow) throws FileNotFoundException {

super(level);

this.crossbow = crossbow;

this.setName("Shooter");

this.setImagePath("src/main/java/com/example/lapa12/images/" + this.getName() + ".jpg");

this.setImage(new Image(new FileInputStream(getImagePath()), 150, 140, true, true));

}

public Element getCrossbow() {

return crossbow;

}

public void setCrossbow(Element crossbow) {

this.crossbow = crossbow;

}

@Override

public String printInfo(){

return super.printInfo() + " with " + this.getCrossbow() + " crossbow";

}

@Override

public String toString() {

return "HilichurlShooter{" +

"crossbow=" + crossbow +

"} " + super.toString();

}

}

package com.example.lapa12.heros;

import com.example.lapa12.Element;

import javafx.scene.image.Image;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.Serializable;

public class Lawachurl extends Mitachurl implements Serializable {

private Element shell;

public Lawachurl() {

}

public Lawachurl(int level, Element axe, Element shield, Element shell) throws FileNotFoundException {

super(level, axe, shield);

this.shell = shell;

this.setName("Lawachurl");

this.setImagePath("src/main/java/com/example/lapa12/images/" + this.getName() + ".jpg");

this.setImage(new Image(new FileInputStream(getImagePath()), 150, 140, true, true));

}

@Override

public String printInfo(){

return super.printInfo() + " and " + this.getShell() + " shell";

}

public Element getShell() {

return shell;

}

public void setShell(Element shell) {

this.shell = shell;

}

@Override

public String toString() {

return "Lawachurl{" +

"shell=" + shell +

"} " + super.toString();

}

}

package com.example.lapa12.heros;

import com.example.lapa12.Element;

import com.fasterxml.jackson.annotation.JsonProperty;

import javafx.scene.image.Image;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.Serializable;

public class Mitachurl extends Hilichurl implements Serializable {

@JsonProperty("axe")

private Element axe;

@JsonProperty("shield")

private Element shield;

public Mitachurl() {

}

public Mitachurl(int level, Element axe, Element shield) throws FileNotFoundException {

super(level);

this.axe = axe;

this.shield = shield;

this.setName("Mitachurl");

this.setImagePath("src/main/java/com/example/lapa12/images/" + this.getName() + ".jpg");

this.setImage(new Image(new FileInputStream(getImagePath()), 150, 140, true, true));

}

@Override

public String printInfo(){

return super.printInfo() + " with " + this.getAxe() + " axe and " + this.getShield()+ " shield";

}

@JsonProperty("axe")

public Element getAxe() {

return axe;

}

public void setAxe(Element axe) {

this.axe = axe;

}

@JsonProperty("shield")

public Element getShield() {

return shield;

}

public void setShield(Element shield) {

this.shield = shield;

}

@Override

public String toString() {

return "Mitachurl{" +

"axe=" + axe +

", shield=" + shield +

"} " + super.toString();

}

}

package com.example.lapa12.factories;

import com.example.lapa12.heros.Hilichurl;

import java.io.FileNotFoundException;

public abstract class Factory {

public abstract Hilichurl create() throws FileNotFoundException;

}

package com.example.lapa12.factories;

import com.example.lapa12.heros.Hilichurl;

import java.io.FileNotFoundException;

public class HilichurlFactory extends Factory{

@Override

public Hilichurl create() throws FileNotFoundException {

return new Hilichurl(100);

}

}

package com.example.lapa12.factories;

import com.example.lapa12.Element;

import com.example.lapa12.heros.Hilichurl;

import com.example.lapa12.heros.HilichurlFighter;

import java.io.FileNotFoundException;

public class HilichurlFighterFactory extends Factory{

@Override

public Hilichurl create() throws FileNotFoundException {

return new HilichurlFighter(56, Element.GEO);

}

}

package com.example.lapa12.factories;

import com.example.lapa12.Element;

import com.example.lapa12.heros.Hilichurl;

import com.example.lapa12.heros.HilichurlGrenadier;

import java.io.FileNotFoundException;

public class HilichurlGrenadierFactory extends Factory{

@Override

public Hilichurl create() throws FileNotFoundException {

return new HilichurlGrenadier(99, Element.HYDRO);

}

}

package com.example.lapa12.factories;

import com.example.lapa12.Element;

import com.example.lapa12.heros.Hilichurl;

import com.example.lapa12.heros.HilichurlGuard;

import java.io.FileNotFoundException;

public class HIlichurlGuardFactory extends Factory{

@Override

public Hilichurl create() throws FileNotFoundException {

return new HilichurlGuard(69, Element.FIRE, Element.DENDRO);

}

}

package com.example.lapa12.factories;

import com.example.lapa12.Element;

import com.example.lapa12.heros.Hilichurl;

import com.example.lapa12.heros.HilichurlShooter;

import java.io.FileNotFoundException;

public class HilichurlShooterFactory extends Factory{

@Override

public Hilichurl create() throws FileNotFoundException {

return new HilichurlShooter(55, Element.ANEMO);

}

}

package com.example.lapa12.factories;

import com.example.lapa12.Element;

import com.example.lapa12.heros.Hilichurl;

import com.example.lapa12.heros.Lawachurl;

import java.io.FileNotFoundException;

public class LawachurlFactory extends Factory{

@Override

public Hilichurl create() throws FileNotFoundException {

return new Lawachurl(80, Element.ELECTRO, Element.GEO, Element.FIRE);

}

}

package com.example.lapa12.factories;

import com.example.lapa12.Element;

import com.example.lapa12.heros.Hilichurl;

import com.example.lapa12.heros.Mitachurl;

import java.io.FileNotFoundException;

public class MitachurlFactory extends Factory{

@Override

public Hilichurl create() throws FileNotFoundException {

return new Mitachurl(79, Element.HYDRO, Element.CRIO);

}

}

package com.example.lapa12;

public enum Element {

GEO, CRIO, ELECTRO, HYDRO, DENDRO, FIRE, ANEMO

}

package com.example.lapa12;

import com.example.lapa12.factories.\*;

import com.example.lapa12.heros.\*;

import com.fasterxml.jackson.databind.ObjectMapper;

import javafx.scene.control.Label;

import javafx.scene.image.ImageView;

import javafx.scene.layout.GridPane;

import java.io.\*;

import java.util.ArrayList;

import java.util.List;

public class Logic {

private static final ObjectMapper mapper = new ObjectMapper();

private static final File BINSerializeFile = new File("ser.bin");

private static final File JSONSerializeFile = new File("ser.json");

public void BINSerialize(List<Hilichurl> characters) throws IOException {

FileOutputStream fileOutputStream = new FileOutputStream(BINSerializeFile);

ObjectOutputStream objectOutputStream = new ObjectOutputStream(fileOutputStream);

objectOutputStream.writeObject(characters);

System.out.println("Well done");

objectOutputStream.close();

}

public void JSONSerialize(Hilichurls characters) throws IOException {

mapper.writeValue(JSONSerializeFile, characters);

System.out.println("Well done");

}

public ArrayList<Hilichurl> BINDeserialize() throws IOException, ClassNotFoundException {

FileInputStream fileInputStream = new FileInputStream(BINSerializeFile);

ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);

ArrayList<Hilichurl> characters = (ArrayList<Hilichurl>) objectInputStream.readObject();

objectInputStream.close();

System.out.println("Well done");

return characters;

}

public Hilichurls JSONDeserialize() throws IOException {

System.out.println("Well done");

return mapper.readValue(JSONSerializeFile, Hilichurls.class);

}

public Hilichurl[] createHilichurls(Hilichurl[] hilichurls) throws FileNotFoundException {

hilichurls[0] = new Hilichurl(20);

hilichurls[1] = new Mitachurl(44, Element.ANEMO, Element.DENDRO);

hilichurls[2] = new HilichurlGrenadier(67, Element.ANEMO);

hilichurls[3] = new HilichurlShooter(55, Element.ELECTRO);

hilichurls[4] = new HilichurlFighter(30, Element.FIRE);

hilichurls[5] = new Lawachurl(89, Element.HYDRO, Element.GEO, Element.FIRE);

hilichurls[6] = new HilichurlGuard(78, Element.CRIO, Element.GEO);

return hilichurls;

}

public void createHilichurlsFactories(Factory[] factories){

factories[0] = new HilichurlFactory();

factories[1] = new MitachurlFactory();

factories[2] = new HilichurlGrenadierFactory();

factories[3] = new HilichurlShooterFactory();

factories[4] = new HilichurlFighterFactory();

factories[5] = new LawachurlFactory();

factories[6] = new HIlichurlGuardFactory();

}

public Hilichurl[] recoverHilichurls(Hilichurl[] hilichurls){

for (Hilichurl hilichurl :

hilichurls) {

hilichurl.recovery();

}

return hilichurls;

}

public void showHilichurls(Hilichurl[] hilichurls, GridPane gridPane){

Label label = new Label();

label.setWrapText(true);

gridPane.add(new ImageView(hilichurls[0].getImage()), 2,0);

label.setText(hilichurls[0].printInfo());

gridPane.add(label, 2,1);

for (int i = 1; i < 5 ; i++) {

Label label1 = new Label();

label1.setWrapText(true);

label1.setText(hilichurls[i].printInfo());

if (i<=2){

gridPane.add(new ImageView(hilichurls[i].getImage()), i-1,2);

gridPane.add(label1, i-1,3);

}

else {

gridPane.add(new ImageView(hilichurls[i].getImage()), i, 2);

gridPane.add(label1, i,3);

}

}

Label label2 = new Label();

label2.setWrapText(true);

gridPane.add(new ImageView(hilichurls[5].getImage()), 0,4);

label2.setText(hilichurls[5].printInfo());

gridPane.add(label2, 0,5);

Label label3 = new Label();

label3.setWrapText(true);

gridPane.add(new ImageView(hilichurls[6].getImage()), 4,4);

label3.setText(hilichurls[6].printInfo());

gridPane.add(label3, 4,5);

}

}

package com.example.lapa12;

import com.example.lapa12.heros.Hilichurl;

import javafx.animation.AnimationTimer;

import javafx.scene.Group;

import javafx.scene.Node;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

public class Movement {

boolean goUp, goDown, goRight, goLeft;

AnimationTimer animationTimer;

public void keyPress(Scene scene, Group root, Hilichurl character, double x, double y) throws FileNotFoundException {

Node hero = new ImageView(new Image(new FileInputStream(character.getImagePath()),150,140, true, true));

root.getChildren().add(hero);

moveHeroTo(x, y, hero);

scene.setOnKeyPressed(keyEvent -> {

switch (keyEvent.getCode()){

case UP -> goUp =true;

case DOWN -> goDown=true;

case LEFT -> goLeft = true;

case RIGHT -> goRight = true;

}

});

scene.setOnKeyReleased(keyEvent -> {

switch (keyEvent.getCode()){

case UP -> goUp =false;

case DOWN -> goDown=false;

case LEFT -> goLeft = false;

case RIGHT -> goRight = false;

}

});

animationTimer = new AnimationTimer() {

@Override

public void handle(long l) {

int dx=0, dy=0;

if (goUp) dy-=2;

if (goDown) dy+=2;

if (goRight) dx+=2;

if (goLeft) dx-=2;

moveHeroBy(dx, dy, hero, character);

}

};

animationTimer.start();

}

private void moveHeroBy(int dx, int dy, Node hero, Hilichurl character){

if (dx==0 && dy==0) return;

double cx = hero.getBoundsInLocal().getWidth()/2;

double cy = hero.getBoundsInLocal().getHeight()/2;

double x = cx + hero.getLayoutX() + dx;

double y = cy + hero.getLayoutY() + dy;

character.setX(x);

character.setY(y);

moveHeroTo(x, y, hero);

}

private void moveHeroTo(double x, double y, Node hero){

double cx = hero.getBoundsInLocal().getWidth()/2;

double cy = hero.getBoundsInLocal().getHeight()/2;

if (x-cx>=0 &&

x+cx<=Main.W &&

y-cy>=0 &&

y+cy<=Main.H){

hero.relocate(x-cx, y-cy);

}

}

public void stopTimer(){

if (this.animationTimer!=null){

animationTimer.stop();

}

}

}

package com.example.lapa12;

import com.example.lapa12.factories.Factory;

import com.example.lapa12.heros.\*;

import javafx.application.Application;

import javafx.collections.FXCollections;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.geometry.Pos;

import javafx.scene.Group;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.layout.HBox;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.Arrays;

public class Main extends Application {

public static final int W = 700;

public static final int H = 700;

Hilichurls hilichurls;

public static void main(String[] args) {

launch(args);

}

@Override

public void start(Stage stage) {

Movement movement = new Movement();

Menu menu = new Menu("Choose character");

MenuItem[] menuItems = new MenuItem[]{new MenuItem("Hilichurl"), new MenuItem("Metachurl"),

new MenuItem("Grenadier"), new MenuItem("Shooter"),new MenuItem( "Fighter"),

new MenuItem("Lawachurl"),new MenuItem( "Guard")};

for (MenuItem item : menuItems) {

menu.getItems().add(item);

}

Factory[] factories = new Factory[7];

hilichurls = new Hilichurls();

Logic logic = new Logic();

logic.createHilichurlsFactories(factories);

MenuBar menuBar = new MenuBar(menu);

Button btnSerialize = new Button("Serialize");

Button btnDeserialize = new Button("Deserialize");

RadioButton rbJSON = new RadioButton("JSON");

HBox controls = new HBox(10,menuBar, rbJSON, btnSerialize, btnDeserialize);

Group root = new Group(controls);

Scene scene = new Scene(root, W, H);

scene.getRoot().requestFocus();

stage.setTitle("lapa1,2");

stage.setScene(scene);

stage.show();

rbJSON.setOnAction(actionEvent -> scene.getRoot().requestFocus());

btnSerialize.setOnAction(actionEvent -> {

try {

if (rbJSON.isSelected()){

logic.JSONSerialize(hilichurls);

} else {

logic.BINSerialize(hilichurls.hilichurls);

}

scene.getRoot().requestFocus();

} catch (IOException e) {

throw new RuntimeException(e);

}

});

btnDeserialize.setOnAction(actionEvent -> {

try {

if (rbJSON.isSelected()) {

hilichurls.hilichurls.addAll(logic.JSONDeserialize().hilichurls);

} else {

hilichurls.hilichurls.addAll(logic.BINDeserialize());

}

for (Hilichurl character :

hilichurls.hilichurls) {

movement.keyPress(scene,root, character, character.getX(), character.getY());

movement.stopTimer();

}

scene.getRoot().requestFocus();

} catch (IOException | ClassNotFoundException e) {

throw new RuntimeException(e);

}

});

EventHandler<ActionEvent> event = e -> scene.setOnMouseClicked(mouseEvent -> {

boolean isShowData = false;

Hilichurl hilichurl = new Hilichurl();

movement.stopTimer();

int index=menu.getItems().indexOf((MenuItem)e.getSource());

double x = mouseEvent.getX();

double y = mouseEvent.getY();

for (int i = 0; i < hilichurls.hilichurls.size(); i++) {

double rightBound = root.getChildren().get(i+1).getBoundsInParent().getMaxX();

double leftBound = root.getChildren().get(i+1).getBoundsInParent().getMinX();

double bottomBound = root.getChildren().get(i+1).getBoundsInParent().getMaxY();

double topBound = root.getChildren().get(i+1).getBoundsInParent().getMinY();

if (x>=leftBound &&

x<=rightBound &&

y>=topBound &&

y<=bottomBound){

isShowData = true;

hilichurl = hilichurls.hilichurls.get(i);

System.out.println("нажал на картинку");

break;

}

}

if (!isShowData) {

Hilichurl currentCharacter;

try {

currentCharacter = factories[index].create();

} catch (FileNotFoundException ex) {

throw new RuntimeException(ex);

}

currentCharacter.setX(x);

currentCharacter.setY(y);

try {

hilichurls.hilichurls.add(currentCharacter);

movement.keyPress(scene, root, currentCharacter, currentCharacter.getX(), currentCharacter.getY());

} catch (Exception exception) {

System.out.println("низя так");

}

}else{

createNewWindow(root,hilichurl);

}

});

for (MenuItem menuItem : menuItems) {

menuItem.setOnAction(event);

}

}

private void createNewWindow(Group group, Hilichurl hilichurl){

Stage optionsStage = new Stage();

Label lLevel = new Label("Level");

TextField tfLevel = new TextField(String.valueOf(hilichurl.getLevel()));

HBox hbLevel = new HBox(5,lLevel,tfLevel);

VBox root = new VBox(20, hbLevel);

ComboBox<Element> cbClub = new ComboBox<>(FXCollections.observableArrayList(Element.values()));

ComboBox<Element> cbShield = new ComboBox<>(FXCollections.observableArrayList(Element.values()));

ComboBox<Element> cbAxe = new ComboBox<>(FXCollections.observableArrayList(Element.values()));

ComboBox<Element> cbShell = new ComboBox<>(FXCollections.observableArrayList(Element.values()));

ComboBox<Element> cbSlime = new ComboBox<>(FXCollections.observableArrayList(Element.values()));

ComboBox<Element> cbCrossbow = new ComboBox<>(FXCollections.observableArrayList(Element.values()));

if (hilichurl.getName().equals("Fighter") || hilichurl.getName().equals("Guard")){

Label lClub = new Label("Club");

cbClub.setValue(((HilichurlFighter) hilichurl).getClub());

HBox hbClub = new HBox(5,lClub,cbClub);

root.getChildren().add(hbClub);

if (hilichurl.getName().equals("Shield")){

Label lShield = new Label("Shield");

cbShield.setValue(((HilichurlGuard) hilichurl).getShield());

HBox hbShield = new HBox(5,lShield,cbShield);

root.getChildren().add(hbShield);

}

}

if (hilichurl.getName().equals("Lawachurl") || hilichurl.getName().equals("Mitachurl")){

Label lAxe = new Label("Axe");

cbAxe.setValue(((Mitachurl) hilichurl).getAxe());

HBox hbAxe = new HBox(5,lAxe,cbAxe);

root.getChildren().add(hbAxe);

Label lShield = new Label("Shield");

cbShield.setValue(((Mitachurl) hilichurl).getShield());

HBox hbShield = new HBox(5,lShield,cbShield);

root.getChildren().add(hbShield);

if (hilichurl.getName().equals("Lawachurl")){

Label lShell = new Label("Shell");

cbShell.setValue(((Lawachurl) hilichurl).getShell());

HBox hbShell = new HBox(5,lShell,cbShell);

root.getChildren().add(hbShell);

}

}

if (hilichurl.getName().equals("Grenadier")){

Label lSlime = new Label("Slime");

cbSlime.setValue(((HilichurlGrenadier) hilichurl).getSlime());

HBox hbSlime = new HBox(5,lSlime,cbSlime);

root.getChildren().add(hbSlime);

}

if (hilichurl.getName().equals("Shooter")){

Label lCrossbow = new Label("Crossbow");

cbCrossbow.setValue(((HilichurlShooter) hilichurl).getCrossbow());

HBox hbCrossbow = new HBox(5,lCrossbow,cbCrossbow);

root.getChildren().add(hbCrossbow);

}

Button btnChange = new Button("Change");

Button btnDelete = new Button("Delete");

HBox hbButtons=new HBox(30,btnChange, btnDelete);

hbButtons.setAlignment(Pos.BOTTOM\_CENTER);

root.getChildren().addAll(hbButtons);

btnDelete.setOnAction(actionEvent -> {

group.getChildren().remove(hilichurls.hilichurls.indexOf(hilichurl)+1);

hilichurls.hilichurls.remove(hilichurl);

optionsStage.close();

}

);

btnChange.setOnAction(actionEvent -> {

hilichurl.setLevel(Integer.parseInt(tfLevel.getText()));

if (hilichurl.getName().equals("Fighter") || hilichurl.getName().equals("Guard")) {

((HilichurlFighter) hilichurl).setClub(cbClub.getValue());

if (hilichurl.getName().equals("Shield")) {

((HilichurlGuard) hilichurl).setShield(cbShield.getValue());

}

}

if (hilichurl.getName().equals("Lawachurl") || hilichurl.getName().equals("Mitachurl")) {

((Mitachurl) hilichurl).setAxe(cbAxe.getValue());

((Mitachurl) hilichurl).setShield(cbShield.getValue());

if (hilichurl.getName().equals("Lawachurl")) {

((Lawachurl) hilichurl).setShell(cbShell.getValue());

}

}

if (hilichurl.getName().equals("Grenadier")) {

((HilichurlGrenadier) hilichurl).setSlime(cbSlime.getValue());

}

if (hilichurl.getName().equals("Shooter")) {

((HilichurlShooter) hilichurl).setCrossbow(cbCrossbow.getValue());

}

optionsStage.close();

});

Scene optionsScene = new Scene(root, 200, 230);

optionsStage.setScene(optionsScene);

optionsStage.setTitle("Options");

optionsStage.show();

}

}

