Бруцкая Анастасия, 251001

package com.example.lapa12;

import java.util.List;

import java.util.ServiceLoader;

import java.util.ServiceLoader.Provider;

import java.util.stream.Collectors;

public interface PluginImplementation {

void doSmth();

static List<PluginImplementation> getServices(ModuleLayer layer) {

return ServiceLoader

.load(layer, PluginImplementation.class)

.stream()

.map(Provider::get)

.collect(Collectors.toList());

}

}

package com.example.lapa12;

public class BasicPluginRealisation implements PluginImplementation {

@Override

public void doSmth() {

System.out.println("Basic realisation");

}

}

public static void loadPlugins(){

Path pluginsDir = Paths.get("plugins");

// Будем искать плагины в папке plugins

ModuleFinder pluginsFinder = ModuleFinder.of(pluginsDir);

List<String> plugins = pluginsFinder

.findAll()

.stream()

.map(ModuleReference::descriptor)

.map(ModuleDescriptor::name)

.toList();

// Создадим конфигурацию, которая выполнит резолюцию указанных модулей (проверит корректность графа зависимостей)

Configuration pluginsConfiguration = ModuleLayer

.boot()

.configuration()

.resolve(pluginsFinder, ModuleFinder.of(), plugins);

// Создадим слой модулей для плагинов

ModuleLayer layer = ModuleLayer

.boot()

.defineModulesWithOneLoader(pluginsConfiguration, ClassLoader.getSystemClassLoader());

// Найдём все реализации сервиса IService в слое плагинов и в слое Boot

List<PluginImplementation> services = PluginImplementation.getServices(layer);

for (PluginImplementation service : services) {

service.doSmth();

}

}

package com.example.hierarchyextension.factories;

import com.example.hierarchyextension.landscapes.Landscape;

import java.io.FileNotFoundException;

public abstract class Factory {

public abstract Landscape create(String imagePath) throws FileNotFoundException;

}

package com.example.hierarchyextension.factories;

import com.example.hierarchyextension.landscapes.Bush;

import com.example.hierarchyextension.landscapes.Landscape;

import java.io.FileNotFoundException;

public class BushFactory extends Factory{

@Override

public Landscape create(String imagePath) throws FileNotFoundException {

return new Bush(imagePath);

}

}

package com.example.hierarchyextension.factories;

import com.example.hierarchyextension.landscapes.Bonfire;

import com.example.hierarchyextension.landscapes.Landscape;

import java.io.FileNotFoundException;

public class BonfireFactory extends Factory {

@Override

public Landscape create(String imagePath) throws FileNotFoundException {

return new Bonfire(imagePath);

}

}

package com.example.hierarchyextension.factories;

import com.example.hierarchyextension.heros.Shamachurl;

import com.example.lapa12.Element;

import com.example.lapa12.factories.Factory;

import com.example.lapa12.heros.Hilichurl;

import javafx.collections.FXCollections;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.HBox;

import javafx.scene.layout.VBox;

import java.io.FileNotFoundException;

public class ShamachurlFactory extends Factory {

@Override

public Hilichurl create(String imagePath) throws FileNotFoundException {

return new Shamachurl(99, imagePath, Element.ANEMO);

}

@Override

public VBox createWindow(Hilichurl hilichurl) {

Label lXP = new Label("XP");

TextField tfXP = new TextField(String.valueOf(hilichurl.getXP()));

tfXP.setDisable(true);

HBox hbXP = new HBox(5,lXP,tfXP);

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, hbXP);

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

Label lStaff = new Label("Staff");

cbStaff.setValue(((Shamachurl) hilichurl).getStaff());

HBox hbStaff = new HBox(5,lStaff,cbStaff);

root.getChildren().add(hbStaff);

return root;

}

}

package com.example.hierarchyextension.factories;

import com.example.hierarchyextension.heros.AbyssMage;

import com.example.hierarchyextension.heros.Shamachurl;

import com.example.lapa12.Element;

import com.example.lapa12.factories.Factory;

import com.example.lapa12.heros.Hilichurl;

import javafx.collections.FXCollections;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.HBox;

import javafx.scene.layout.VBox;

import java.io.FileNotFoundException;

public class AbyssMageFactory extends Factory {

@Override

public Hilichurl create(String imagePath) throws FileNotFoundException {

return new AbyssMage(23, imagePath, Element.CRIO, Element.ELECTRO);

}

@Override

public VBox createWindow(Hilichurl hilichurl) {

Label lXP = new Label("XP");

TextField tfXP = new TextField(String.valueOf(hilichurl.getXP()));

tfXP.setDisable(true);

HBox hbXP = new HBox(5,lXP,tfXP);

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, hbXP);

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

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

Label lStaff = new Label("Staff");

cbStaff.setValue(((Shamachurl) hilichurl).getStaff());

HBox hbStaff = new HBox(5,lStaff,cbStaff);

root.getChildren().add(hbStaff);

Label lShield = new Label("Shield");

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

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

root.getChildren().add(hbShield);

return root;

}

}

package com.example.hierarchyextension.heros;

import com.example.lapa12.Element;

import com.example.lapa12.heros.Hilichurl;

import java.io.FileNotFoundException;

import java.io.Serializable;

public class Shamachurl extends Hilichurl implements Serializable {

private Element staff;

public Shamachurl() {

}

public Shamachurl(int level, String imagePath, Element staff) throws FileNotFoundException {

super(level, imagePath);

this.staff = staff;

this.setName("Shamachurl");

}

public Element getStaff() {

return staff;

}

public void setStaff(Element staff) {

this.staff = staff;

}

@Override

public String toString() {

return "Shamachurl{" +

"staff=" + staff +

"} " + super.toString();

}

@Override

public String printInfo(){

return super.printInfo() +"with " + this.getStaff() + " staff ";

}

}

package com.example.hierarchyextension.heros;

import com.example.lapa12.Element;

import java.io.FileNotFoundException;

import java.io.Serializable;

public class AbyssMage extends Shamachurl implements Serializable {

private Element shield;

public AbyssMage() {

}

public AbyssMage(int level, String imagePath, Element staff, Element shield) throws FileNotFoundException {

super(level, imagePath, staff);

this.shield = shield;

this.setName("Abyss Mage");

}

@Override

public String printInfo(){

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

}

public Element getShield() {

return shield;

}

public void setShield(Element shield) {

this.shield = shield;

}

@Override

public String toString() {

return "AbyssMage{" +

"shield=" + shield +

"} " + super.toString();

}

}

package com.example.hierarchyextension.landscapes;

import com.example.lapa12.Main;

import com.example.lapa12.heros.Hilichurl;

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;

import java.util.ArrayList;

import static com.example.hierarchyextension.HierarchyExtension.heroes;

import static com.example.hierarchyextension.HierarchyExtension.landscapes;

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

@JsonSubTypes({})

public abstract class Landscape implements Serializable {

private String name;

private String imagePath;

@JsonIgnore

transient private Image image;

private int XP;

private final int maxXP;

private double x;

private double y;

public abstract void hitHilicurl(Hilichurl hilichurl);

public void findNearHilichurls(){

int i = landscapes.landscapes.indexOf(this);

double rightBound = heroes.getChildren().get(i).getBoundsInParent().getMaxX();

double leftBound = heroes.getChildren().get(i).getBoundsInParent().getMinX();

double bottomBound = heroes.getChildren().get(i).getBoundsInParent().getMaxY();

double topBound = heroes.getChildren().get(i).getBoundsInParent().getMinY();

for (Hilichurl hilichurl :

new ArrayList<>(Main.hilichurls.hilichurls)) {

double hilichurlX = hilichurl.getX();

double hilichurlY = hilichurl.getY();

if (hilichurlX >= leftBound - 100 &&

hilichurlX <= rightBound + 100 &&

hilichurlY >= topBound - 100 &&

hilichurlY <= bottomBound + 100) {

System.out.println("ударило кустом");

this.hitHilicurl(hilichurl);

if (hilichurl.getXP() < 10) {

Main.heroes.getChildren().remove(Main.hilichurls.hilichurls.indexOf(hilichurl));

Main.hilichurls.hilichurls.remove(hilichurl);

System.out.println(hilichurl.getName() + " died because of bush");

}

}

}

}

public Landscape(String imagePath) throws FileNotFoundException {

this.imagePath = imagePath;

this.image = new Image(new FileInputStream(imagePath), 100,130, true,true);

this.maxXP=200;

this.XP=this.maxXP;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getImagePath() {

return imagePath;

}

public void setImagePath(String imagePath) {

this.imagePath = imagePath;

}

public Image getImage() {

return image;

}

public void setImage(Image image) {

this.image = image;

}

public int getXP() {

return XP;

}

public void setXP(int XP) {

this.XP = XP;

}

public int getMaxXP() {

return maxXP;

}

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;

}

@Override

public String toString() {

return "Landscape{" +

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

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

", XP=" + XP +

", maxXP=" + maxXP +

", x=" + x +

", y=" + y +

'}';

}

}

package com.example.hierarchyextension.landscapes;

import com.example.lapa12.Element;

import com.example.lapa12.heros.Hilichurl;

import java.io.FileNotFoundException;

public class Bush extends Landscape{

private final Element element=Element.DENDRO;

public Bush(String imagePath) throws FileNotFoundException {

super(imagePath);

this.setName("Bush");

}

public Element getElement() {

return element;

}

@Override

public String toString() {

return "Bush{" +

"element=" + element +

"} " + super.toString();

}

@Override

public void hitHilicurl(Hilichurl hilichurl) {

hilichurl.setXP(hilichurl.getXP() - 10);

}

}

package com.example.hierarchyextension.landscapes;

import com.example.lapa12.Element;

import com.example.lapa12.heros.Hilichurl;

import java.io.FileNotFoundException;

public class Bonfire extends Landscape{

private final Element element=Element.FIRE;

public Bonfire(String imagePath) throws FileNotFoundException {

super(imagePath);

this.setName("Bonfire");

}

@Override

public void hitHilicurl(Hilichurl hilichurl) {

hilichurl.setXP(hilichurl.getXP() - 30);

}

public Element getElement() {

return element;

}

@Override

public String toString() {

return "Bonfire{" +

"element=" + element +

"} " + super.toString();

}

}

package com.example.hierarchyextension.landscapes;

import java.util.ArrayList;

import java.util.List;

public class Landscapes {

public List<Landscape> landscapes = new ArrayList<>();

}

package com.example.hierarchyextension;

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

import com.example.hierarchyextension.landscapes.Landscape;

import com.example.lapa12.Main;

import javafx.animation.AnimationTimer;

import java.io.File;

import static com.example.hierarchyextension.HierarchyExtension.imagePaths;

import static com.example.hierarchyextension.HierarchyExtension.landscapes;

public class Logic extends com.example.lapa12.Logic {

AnimationTimer animationTimer;

public void hitHilichurl() {

double desiredFrameRate = 0.3; // Желаемая частота кадров в кадрах в секунду

long oneFrameTime = (long) (1\_000\_000\_000 / desiredFrameRate); // Время одного кадра в наносекундах

final long[] lastUpdateTime = {0};

animationTimer = new AnimationTimer() {

@Override

public void handle(long currentTime) {

long elapsedTime = currentTime - lastUpdateTime[0];

if (elapsedTime >= oneFrameTime) {

for (Landscape landscape :

landscapes.landscapes) {

landscape.findNearHilichurls();

}

lastUpdateTime[0] = currentTime;

}

}

}; animationTimer.start();

}

public void loadLandscapeImages(){

String folderPath = "../HierarchyExtension/images/landscape";

File folder = new File(folderPath);

File[] listOfFiles = folder.listFiles();

if (listOfFiles != null) {

for (File file : listOfFiles) {

if (file.isFile()) {

imagePaths.add(file.getAbsolutePath());

}

}

}

}

public void loadHilichurlImages(){

String folderPath = "../HierarchyExtension/images/hilichurl";

File folder = new File(folderPath);

File[] listOfFiles = folder.listFiles();

if (listOfFiles != null) {

for (File file : listOfFiles) {

if (file.isFile()) {

Main.imagePaths.add(file.getAbsolutePath());

}

}

}

}

public void createLandscapesFactories(Factory[] factories){

factories[0] = new BonfireFactory();

factories[1] = new BushFactory();

}

public void createHilichurlFactories(){

Main.factories.add(new ShamachurlFactory());

Main.factories.add(new AbyssMageFactory());

}

}

package com.example.hierarchyextension;

import com.example.hierarchyextension.factories.Factory;

import com.example.hierarchyextension.landscapes.Landscape;

import com.example.hierarchyextension.landscapes.Landscapes;

import com.example.lapa12.PluginImplementation;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.geometry.Pos;

import javafx.scene.Group;

import javafx.scene.Node;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.image.ImageView;

import javafx.scene.layout.HBox;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

import java.io.FileNotFoundException;

import java.util.ArrayList;

import java.util.List;

import static com.example.lapa12.Main.\*;

public class HierarchyExtension implements PluginImplementation {

public static void main(String[] args) {

System.out.println("Plugin was built");

}

public static Group heroes;

public static Landscapes landscapes;

Factory[] landscapeFactories;

com.example.lapa12.factories.Factory[] hilichurlFactories;

public static ArrayList<String> landscapeNames = new ArrayList<>(List.of(new String[]{"Bonfire", "Bush"}));

public static ArrayList<String> hilichurlNames = new ArrayList<>(List.of(new String[]{"Shamachurl", "Abyss Mage"}));

public static ArrayList<String> imagePaths = new ArrayList<>();

Logic logic = new Logic();

@Override

public void doSmth() {

System.out.println("extension");

addNewInterface();

extendHierarchy();

logic.hitHilichurl();

}

public void addNewInterface(){

logic.loadLandscapeImages();

logic.loadHilichurlImages();

typeNames.addAll(hilichurlNames);

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

MenuItem[] menuItems = new MenuItem[landscapeNames.size()];

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

menuItems[i] = new MenuItem(landscapeNames.get(i));

}

for (MenuItem item : menuItems) {

menu.getItems().add(item);

}

MenuBar menuBar = new MenuBar(menu);

controls.getChildren().add(menuBar);

landscapeFactories = new Factory[2];

landscapes = new Landscapes();

logic.createLandscapesFactories(landscapeFactories);

heroes = new Group();

root.getChildren().add(heroes);

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

Landscape landscape;

int index = 0;

if (e.getSource() instanceof MenuItem) {

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

}

double x = mouseEvent.getX();

double y = mouseEvent.getY();

int i = isShowData(heroes, x,y);

if (i<0) {

Landscape currentLandscape;

try {

currentLandscape = landscapeFactories[index].create(imagePaths.get(index));

} catch (FileNotFoundException ex) {

throw new RuntimeException(ex);

}

currentLandscape.setX(x);

currentLandscape.setY(y);

try {

landscapes.landscapes.add(currentLandscape);

Node hero = new ImageView(currentLandscape.getImage());

heroes.getChildren().add(hero);

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

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

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

} catch (Exception exception) {

System.out.println("низя так");

}

} else {

landscape = landscapes.landscapes.get(i);

createNewWindow(heroes, landscape);

}

});

for (MenuItem menuItem : menuItems) {

menuItem.setOnAction(event);

}

}

public void extendHierarchy(){

hilichurlFactories = new com.example.lapa12.factories.Factory[2];

for (String typeName : hilichurlNames) {

menuItems.add(new MenuItem(typeName));

}

for (int i = menuItems.size()-hilichurlNames.size(); i < menuItems.size(); i++) {

MenuItem item = menuItems.get(i);

menu.getItems().add(item);

}

logic.createHilichurlFactories();

}

private int isShowData(Group heroes, double x, double y){

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

double rightBound = heroes.getChildren().get(i).getBoundsInParent().getMaxX();

double leftBound = heroes.getChildren().get(i).getBoundsInParent().getMinX();

double bottomBound = heroes.getChildren().get(i).getBoundsInParent().getMaxY();

double topBound = heroes.getChildren().get(i).getBoundsInParent().getMinY();

if (x >= leftBound &&

x <= rightBound &&

y >= topBound &&

y <= bottomBound) {

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

return i;

}

}

return -1;

}

private void createNewWindow(Group group, Landscape landscape) {

Stage optionsStage = new Stage();

Label lXP = new Label("XP");

TextField tfXP = new TextField(String.valueOf(landscape.getXP()));

HBox hbXP = new HBox(5, lXP,tfXP);

Button btnDelete = new Button("Delete");

HBox hbButtons = new HBox(btnDelete);

hbButtons.setAlignment(Pos.BOTTOM\_CENTER);

VBox root = new VBox(20, hbXP, hbButtons);

btnDelete.setOnAction(actionEvent -> {

group.getChildren().remove(landscapes.landscapes.indexOf(landscape));

landscapes.landscapes.remove(landscape);

optionsStage.close();

}

);

Scene optionsScene = new Scene(root, 200, 100);

optionsStage.setScene(optionsScene);

optionsStage.setTitle("Options");

optionsStage.show();

}

}

