# Exp7 GUI program design and Event-Driven Programming

**1 Experiment purpose**

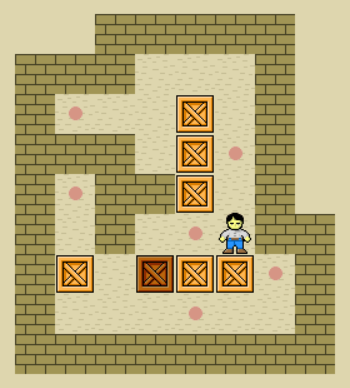
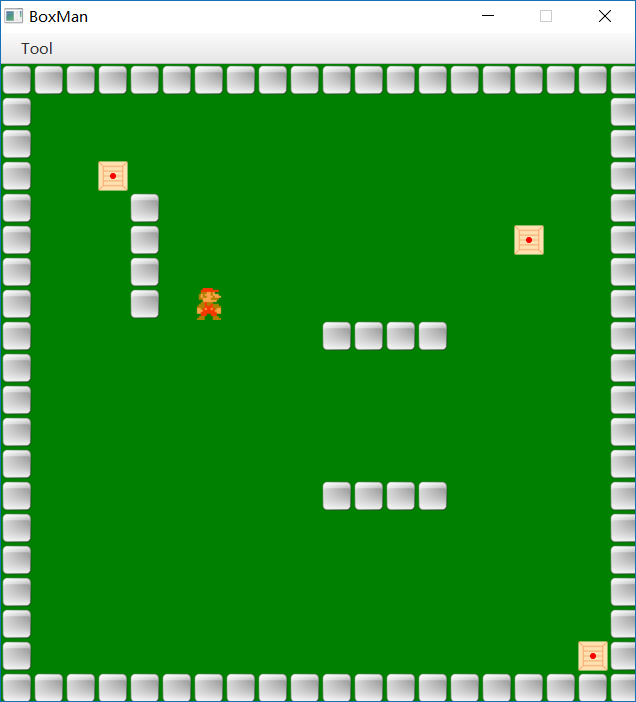
Through this training, the students should master GUI design using JavaFX and event handling.

**2 Development Environment**

1. Notepad++
2. JDK 1.8+

**3 Content and Steps**

**Problem 1:** Sokoban Game.



**Steps:**

1. Define a class to hold URLs

**public** **class** Paths {

**public** **final** **static** String ***MAIN*** = "file:D:/main.png";

**public** **final** **static** String ***BOX*** = "file:D:/box.png";

**public** **final** **static** String ***STONE*** = "file:D:/stone.png";

**public** **final** **static** String ***MAN*** = "file:D:/man.png";

}

1. Define the main interface of the game

**public** **class** MainProgram **extends** Application {

@Override

**public** **void** start(Stage primaryStage) **throws** Exception {

VBox root = **new** VBox();

ImageView img = **new** ImageView(Paths.***MAIN***);

img.setFitWidth(400);

img.setFitHeight(200);

Button playBtn = **new** Button("开始游戏");

playBtn.setPrefWidth(400);

playBtn.setPrefHeight(100);

playBtn.setOnAction(e ->{**new** PlayStage().show();});

Button editBtn = **new** Button("编辑地图");

editBtn.setPrefWidth(400);

editBtn.setPrefHeight(100);

editBtn.setOnAction(e ->{**new** MapEdit().show();});

root.getChildren().addAll(img, playBtn, editBtn);

root.setAlignment(Pos.***CENTER***);

Scene scene = **new** Scene(root, 400, 400);

primaryStage.setScene(scene);

primaryStage.setTitle("推箱子");

primaryStage.getIcons().add(**new** Image(Paths.***BOX***));

primaryStage.setResizable(**false**);

primaryStage.show();

}

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

// **TODO** Auto-generated method stub

*launch*(args);

}

}



1. Map Editing Program.
2. Initialization

**public** **class** MapEdit **extends** Stage {

**int** mapw = 32 \* 20;

**int** maph = 32 \* 20;

Node[][] elements = **new** Node[20][20];

**int** index = 0;

**public** MapEdit(){

VBox root = **new** VBox();

Pane p = **new** Pane();

ImageView man = **new** ImageView(Paths.***MAN***);

man.setX(32 \* 10);

man.setY(32 \* 10);

MenuBar menuBar = **new** MenuBar();

Menu menu1 = **new** Menu("菜单");

menuBar.getMenus().add(menu1);

MenuItem menu11 = **new** MenuItem("保存地图");

menu1.getItems().add(menu11);

menu11.setOnAction(e ->{//(4)});

MenuItem menu12 = **new** MenuItem("重置地图");

menu1.getItems().add(menu12);

menu1.setOnAction(e -> {//**(2)**});

root.getChildren().addAll(menuBar, p);

Scene scene = **new** Scene(root, mapw, maph + 32);

scene.setOnKeyPressed(e ->{//(3)});

**this**.setScene(scene);

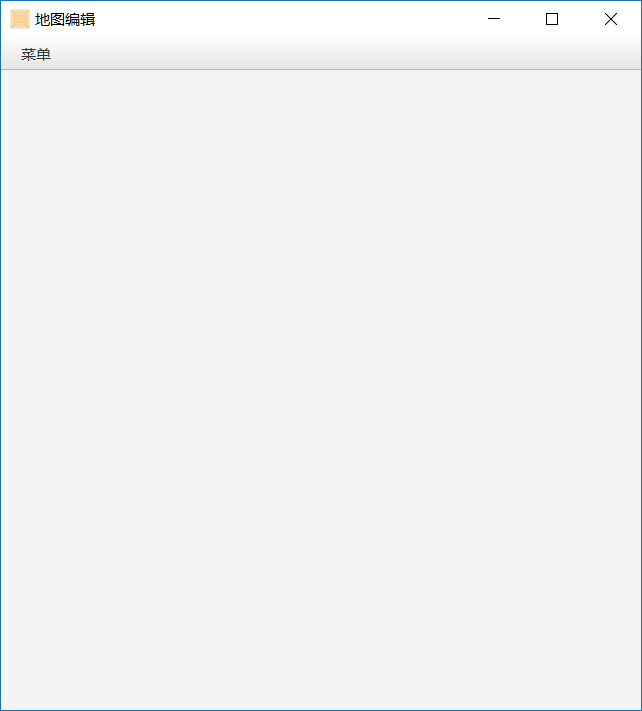
**this**.setTitle("地图编辑");

**this**.getIcons().add(**new** Image(Paths.***BOX***));

//this.setResizable(false);

}

}



(2) Reset map

{

emptyMap();

p.getChildren().clear();

**for**(**int** i=0; i< elements.length; i++){

**for**(**int** j = 0; j < elements[i].length; j++){

**if**(elements[i][j] != **null**){

p.getChildren().add(elements[i][j]);

}

}

}

p.getChildren().add(man);

}

**public** **void** emptyMap(){

**for**(**int** i=0; i< elements.length; i++){

**for**(**int** j = 0; j < elements[i].length; j++){

**if**(**?||?||?||?**){

ImageView stone = **new** ImageView(Paths.***STONE***);

stone.setX(i \* 32);

stone.setY(j \* 32);

elements[i][j] = stone;

}**else**{

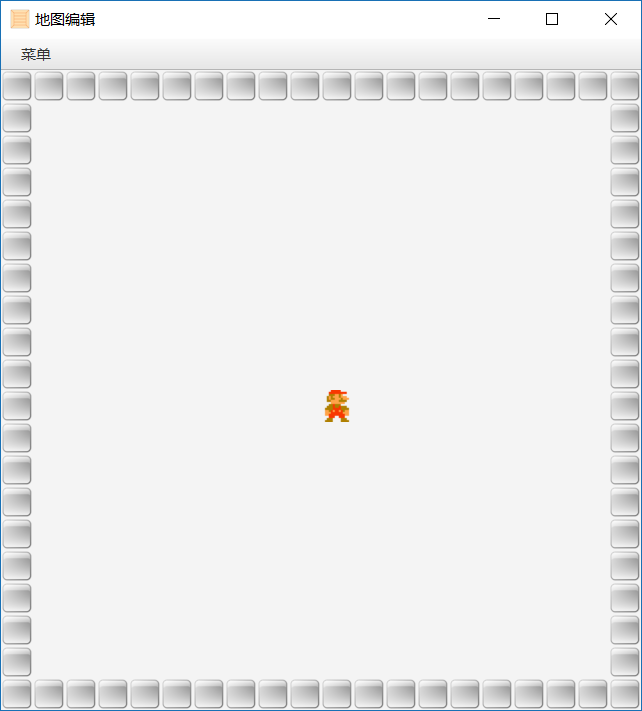
elements[i][j] = **null**;

}

}

}

}



(3) Add elements to pane

{

**switch**(e.getCode()){

**case** ***UP***: man.setY(man.getY() - 32);**break**;

**case** ***DOWN***: man.setY(man.getY() + 32);**break**;

**case** ***LEFT***: man.setX(man.getX() - 32);**break**;

**case** ***RIGHT***: man.setX(man.getX() + 32);**break**;

**case** ***SPACE***: {

Node item = **null**;

**switch**((**int**)(index++ % 3)){

**case** 0: {

ImageView img = **new** ImageView(Paths.***STONE***);

img.setX(man.getX());

img.setY(man.getY());

item = img;

};**break**;

**case** 1: {

**//BOX**

};**break**;

**case** 2: {

**//Circle**

};**break**;

}

**int** manx = (**int**)man.getX()/32;

**int** many = (**int**)man.getY()/32;

**if**(elements[manx][many] != **null**){

p.getChildren().remove(elements[manx][many]);

}

elements[manx][many] = item;

p.getChildren().add(item);

};**break**;

**default**:

**break**;

}

}

(4)Save map.

{

FileChooser directoryChooser=**new** FileChooser();

File file = directoryChooser.showSaveDialog(**this**);

**try** {

PrintWriter pw = **new** PrintWriter(file);

**for**(**int** i=0; i< elements.length; i++){

**for**(**int** j = 0; j < elements[i].length; j++){

**if**(elements[i][j] != **null**){

**if**(elements[i][j] **instanceof** Circle){

**// output i,j,circle**

}**else**{

ImageView img = (ImageView)elements[i][j];

**if**(stone){

**//output i,j,stone;**

}**else**{

**//output i,j,box;**

}

}

}**else**{

pw.print(i + "," + j + ",null\n");

}

}

}

pw.close();

} **catch** (FileNotFoundException e1) {

e1.printStackTrace();

}

}

1. Play game
2. Initialization

**public** **class** PlayStage **extends** Stage {

**int** mapw = 32 \* 20;

**int** maph = 32 \* 20;

Node[][] elements = **new** Node[20][20]; //地图格子

List<Circle> circleList = **new** ArrayList<Circle>();

@SuppressWarnings("incomplete-switch")

**public** PlayStage(){

VBox root = **new** VBox();

Pane p = **new** Pane();

ImageView man = **new** ImageView(Paths.***MAN***);

MenuBar menuBar = **new** MenuBar();

Menu menu1 = **new** Menu("菜单");

menuBar.getMenus().add(menu1);

MenuItem menu11 = **new** MenuItem("加载地图");

menu1.getItems().add(menu11);

menu11.setOnAction(e -> **(2)**);

root.getChildren().addAll(menuBar, p);

Scene scene = **new** Scene(root, mapw, maph + 32);

scene.setOnKeyPressed(e -> **(3)**);

**this**.setScene(scene);

**this**.setTitle("游戏界面");

**this**.getIcons().add(**new** Image(Paths.***BOX***));

}

}

1. Load Map

{

FileChooser directoryChooser=**new** FileChooser();

File file = directoryChooser.showOpenDialog(**this**);

**try** {

Scanner input = **new** Scanner(**new** FileInputStream(file));

p.getChildren().clear();

**while**(input.hasNext()){

String[] items = input.next().split(",");

**int** i = Integer.*parseInt*(items[0]);

**int** j = Integer.*parseInt*(items[1]);

**switch**(items[2]){

**case** "circle": {

Circle c = **new** Circle(i \* 32 + 16, j \* 32 + 16, 5);

c.setFill(Color.***RED***);

circleList.add(c);

elements[i][j] = **null**;

p.getChildren().add(c);

}; **break**;

**case** "box": {

ImageView img = **new** ImageView(Paths.***BOX***);

img.setX(i \* 32);

img.setY(j \* 32);

elements[i][j] = img;

p.getChildren().add(img);

}; **break**;

**case** "stone": {

**//add stone**

};**break**;

**case** "null": elements[i][j] = **null**;**break**;

}

}

//add man

**boolean** manExist = **false**;

**while**(!manExist){

**int** i = (**int**)(Math.*random*() \* 20);

**int** j = (**int**)(Math.*random*() \* 20);

**if**(elements[i][j] == **null**){

man.setX(i \* 32);

man.setY(j \* 32);

p.getChildren().add(man);

manExist = **true**;

}

}

input.close();

} **catch** (IOException e1) {

e1.printStackTrace();

}

}

1. Play and judge

{

**int** x = (**int**)(man.getX() / 32);

**int** y = (**int**)(man.getY() / 32);

**switch**(e.getCode()){

**case** UP:{

//System.out.println(man.getY());

Node node = elements[x][y - 1];

System.out.println(node);

**if**(node **instanceof** ImageView){

ImageView img = (ImageView)node;

**if**(img.getImage().impl\_getUrl().equals(Paths.BOX)){

**if**(elements[x][y - 2] == **null**){

elements[x][y - 2] = img;

elements[x][y - 1] = **null**;

img.setY(img.getY() - 32);

man.setY(man.getY() - 32);

}

}

}**else**{

man.setY(man.getY() - 32);

}

};**break**;

**case** DOWN:{

**//down**

};**break**;

**case** LEFT:{

**//left**

};**break**;

**case** RIGHT:{

**//right**

};**break**;

}

//判断游戏是否完成

**boolean** gameover = **true**;

**for**(Circle c:circleList){

**int** cx = (**int**)((c.getCenterX() - 16) / 32);

**int** cy = (**int**)((c.getCenterY() - 16) / 32);

**if**(elements[cx][cy] == **null**){

gameover = **false**;

}

}

**if**(gameover){

Alert alert = **new** Alert(AlertType.***INFORMATION***);

alert.setTitle("信息提示");

alert.setHeaderText(**null**);

alert.setContentText("游戏完成!");

alert.showAndWait();

}

}