**Vocabulary**

Start the game

Rance

个人属性 attributes

攻击力 ATK

防御力 DEF

财产 property

状态 state

武器 weapon

等级 level

地图:

Select the map

和式旅馆 hotel

小酒馆 bistro

等级商店 level store

武器商店 weapon store

海之家 home of sea

边境出入口 walk the maze

打发时间 kill time

Buy or use Items and weapons

道具：

世色癌50 medicine 50

世色癌100 medicine 100

炒面 Chow mein

绿茶 Green tea

时钟 clock

火腿 ham

商品处理 item processing

武器：

长剑 long sword

游击剑 Swim sword

匕首 dagger

棍棒 stick

魔法剑 magic sword

小型护盾 Small shield

朴素护盾 simple shield

闪光盾 flash shield

锤子 hammer

Enter the maze

Fight or run away

怪物 monster

怪物血量 monster HP

怪物最大血量 monster maxHP

怪物行为 monster behavior

怪物经验 monster EXP

怪物弱点 monster weakness

怪物种类：

果冻 jelly

盗贼 robber

红盗贼 red robber

芋虫 Taro worm

妖精狐 The goblin fox

哈尼 honey

异常状态: abnormal state

陷阱类 trap

迷路 lost

内八 toeIn

兴奋 excitement

结石 lithiasis

乌贼人 cuttlefishman

熬夜 stayUpLate

警报 alarm

接收 reception

锻炼 exercise

临战 battle

休息室 rest

开始游戏 start

Enter the scene

场景

树丛 cross

圣者之碑 The holy one of monuments

迷宫

遭遇敌人 encounter enemy

战斗结算 settlement damage

宝箱 treasure

package sample;  
  
import sample.buttons.\*;  
import sample.itemPart.characterSystem.Person;  
import javafx.scene.control.Label;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.Pane;  
import javafx.scene.text.Font;  
import javafx.scene.text.FontWeight;  
  
  
public class BattlePane extends Pane {  
 ItemPane[] itemList = new ItemPane[21];  
 // int HPTotal = 120;  
// int HPCurrent = 8;  
 int itemPage = 0;  
 GeneralButton nextPage;  
 GeneralButton prePage;  
// Pane HPBackgroundPane = new Pane();  
// Pane HPCurrentImg = new Pane();  
// Pane HPBlackImg = new Pane();  
// Label HPCurrentLabel = new Label("" + HPCurrent);  
// Label HPTotalLabel = new Label("" + HPTotal);  
  
  
 public BattlePane(){  
 Person p = Person.*getInstance*();  
 //要求  
 //变量名指示不清，改变量名  
 Image battleBackground = new Image("file:.\\images\\battleBackground.png");  
 Image battledRance = new Image("file:.\\images\\ranceinbattle.png");  
 Image tentPicture = new Image("file:.\\images\\tent2.png");  
 Image thief1 = new Image("file:.\\images\\thief1.png");  
 Image defenseD = new Image("file:.\\images\\defenseD.png");  
 Image HPBackground = new Image("file:.\\images\\HPBackground.png");  
 Image HPCurrent = new Image("file:.\\images\\HPCurrent.png");  
 //在这里面放backpack  
  
// simpleFactory s1 = new simpleFactory();  
//  
// p.getBackpack().add(s1.buildXIHUXIHU());  
// p.getMedicineBackpack().add(s1.buildXIHUXIHU());  
// System.out.println(p.getBackpack().get(0).getUrl());  
// System.out.println(p.getBackpack().get(0).getPrice());  
// System.out.println(((medicine)p.getBackpack().get(0)).getHeal());  
  
  
 //要求  
 //以下几个组件，如果需要之后修改的话，把它作为BattlePane类的属性，或者直接全部变成BattlePane类的属性，可以参考mainframePane  
 //把非必要的注释删了  
 ImageView tent1 = new ImageView();  
 tent1.setImage(battleBackground);  
 tent1.setFitHeight(battleBackground.getHeight());  
 tent1.setFitWidth(battleBackground.getWidth());  
 int tent1Left = 0;  
 int tent1Top = 0;  
 tent1.setX(tent1Left);  
 tent1.setY(tent1Top);  
  
 ImageView tent2 = new ImageView();  
 tent2.setImage(battledRance);  
 tent2.setFitHeight(battledRance.getHeight());  
 int tent2Left = -100;  
 int tent2Top = 120;  
// tent2.setFitWidth(1150);  
 tent2.setX(tent2Left);  
 tent2.setY(tent2Top);  
  
 ImageView tent3 = new ImageView();  
 tent3.setImage(tentPicture);  
 tent3.setFitHeight(tentPicture.getHeight());  
 tent3.setFitWidth(tentPicture.getWidth());  
 int tent3Left = -50;  
 int tent3Top = 510;  
 tent3.setX(tent3Left);  
 tent3.setY(tent3Top);  
  
 ImageView tent4 = new ImageView();  
 tent4.setImage(thief1);  
 tent4.setFitHeight(thief1.getHeight());  
 int tent4Left = 600;  
 int tent4Top = 30;  
 tent4.setX(tent4Left);  
 tent4.setY(tent4Top);  
  
 ImageView tent5 = new ImageView();  
 tent5.setImage(defenseD);  
 tent5.setFitHeight(defenseD.getHeight());  
 int tent5Left = 700;  
 int tent5Top = 300;  
 tent5.setX(tent5Left);  
 tent5.setY(tent5Top);  
  
 ImageView tent6 = new ImageView();  
 tent6.setImage(HPBackground);  
 tent6.setFitHeight(HPBackground.getHeight());  
 int tent6Left = -50;  
 int tent6Top = 210;  
 tent6.setX(tent6Left);  
 tent6.setY(tent6Top);  
  
 ImageView tent7 = new ImageView();  
 tent7.setImage(HPCurrent);  
 tent7.setFitHeight(HPCurrent.getHeight());  
 int tent7Left = 27;  
 int tent7Top = 281;  
 tent7.setX(tent7Left);  
 tent7.setY(tent7Top);  
  
// Label level = new Label("?");  
// level.setLayoutX(460 -100 + tent1Left);  
// level.setLayoutY(90 - (-300) + tent1Top);  
// level.setFont(Font.font("Timer New Roman", FontWeight.BOLD, 45));  
//  
 Label hp = new Label("160");  
 hp.setLayoutX(50);  
 hp.setLayoutY(280);  
 hp.setFont(Font.*font*("Timer New Roman", FontWeight.*BOLD*, 35));  
//  
 Label currentHp = new Label("125");  
 currentHp.setLayoutX(50);  
 currentHp.setLayoutY(240);  
 currentHp.setFont(Font.*font*("Timer New Roman", FontWeight.*BOLD*, 35));  
//  
// Label exp = new Label("4");  
// exp.setLayoutX(575 -100 + tent1Left);  
// exp.setLayoutY(185 - (-300) + tent1Top);  
// exp.setFont(Font.font("Timer New Roman", FontWeight.BOLD, 45));  
//  
// Label gold = new Label("6");  
// gold.setLayoutX(707 -100 + tent1Left);  
// gold.setLayoutY(185 - (-300) + tent1Top);  
// gold.setFont(Font.font("Timer New Roman", FontWeight.BOLD, 45));  
//  
// Label chip = new Label("7 10");  
// chip.setLayoutX(822 -100 + tent1Left);  
// chip.setLayoutY(185 - (-300) + tent1Top);  
// chip.setFont(Font.font("Timer New Roman", FontWeight.BOLD, 40));  
//  
// Label skill = new Label("8");  
// Label condition = new Label("9");  
// Label note = new Label("10");  
  
 this.getChildren().add(tent1);  
 this.getChildren().add(tent2);  
 this.getChildren().add(tent3);  
 this.getChildren().add(tent4);  
 this.getChildren().add(tent5);  
 this.getChildren().add(tent6);  
 this.getChildren().add(tent7);  
  
// this.getChildren().add(level);  
 this.getChildren().add(hp);  
 this.getChildren().add(currentHp);  
// this.getChildren().add(exp);  
// this.getChildren().add(gold);  
// this.getChildren().add(chip);  
  
 CloseTentButton closeTentButton = new CloseTentButton("快点回去城下町吧","200",10,720);  
 this.getChildren().add(closeTentButton);  
  
 StartBattleButton startBattleButton = new StartBattleButton(" 战斗开始！","200",350,520);  
 this.getChildren().add(startBattleButton);  
  
 SelectAllButton selectAllButton = new SelectAllButton(" 全选择","200",570,520);  
 this.getChildren().add(selectAllButton);  
  
 CancelAllButton cancelAllButton = new CancelAllButton(" 全选择解除","200",800,520);  
 this.getChildren().add(cancelAllButton);  
  
 //以下是物品系统加的东西  
 //bet的位置大小还要调一下  
 nextPage = new GeneralButton(".\\images\\nextItemPage\_hover.png",".\\images\\nextItemPage\_preparing.png",".\\images\\nextItemPage\_pressable.png",".\\images\\nextItemPage\_pressed.png");  
 //图片还没有放进去  
 nextPage.setOnMouseReleased(e->{  
 if (getItemPage() != 3){  
 setItemPage(getItemPage()+1);  
 }  
 });  
 getChildren().add(nextPage);  
 nextPage.setLayoutX(0);//需要调整位置  
 nextPage.setLayoutY(0);  
  
 prePage = new GeneralButton(".\\images\\preItemPage\_hover.png",".\\images\\preItemPage\_preparing.png",".\\images\\preItemPage\_pressable.png",".\\images\\preItemPage\_pressed.png");  
 //图片还没有放进去  
 prePage.setOnMouseReleased(e->{  
 if (getItemPage() != 0){  
 setItemPage(getItemPage()-1);  
 }  
 });  
 getChildren().add(prePage);  
 prePage.setLayoutX(0);//需要调整位置  
 prePage.setLayoutY(0);  
  
 for (int i = 0; i < itemList.length; i++) {  
 itemList[i] = new ItemPane(i);  
 getChildren().add(itemList[i]);  
 }  
 refreshItems();  
 changeToItemPage(0);  
 }  
  
 public Pane[] getItemList() {  
 return itemList;  
 }  
  
 public void refreshItems(){  
 Person person = Person.*getInstance*();  
 for (int i = 0; i < person.getItemList().size(); i++) {  
 itemList[i].setImage(person.getItemList().get(i).getUrl());  
 }  
 }  
  
 //page start from 0  
 public void changeToItemPage(int page){  
 for (ItemPane item :  
 itemList) {  
 item.setVisible(false);  
 }  
 for (int i = 7\*page; i < 7\*page + 7; i++) {  
 itemList[i].setVisible(true);  
 }  
 }  
  
 public void setItemPage(int itemPage) {  
 this.itemPage = itemPage;  
 changeToItemPage(itemPage);  
 }  
  
 public int getItemPage() {  
 return itemPage;  
 }