**package** game.model;

**Board**

**import** java.util.ArrayList;

**import** java.util.Random;

**public** **class** Board {

**private** **int** size;

**private** ArrayList<Square> squares;

**private** **int** FREEZE\_COUNT = 10;

**private** **int** HOME\_COUNT = 10;

**private** **int** LADDER\_COUNT = 10;

**public** Board(**int** size){

**this**.size = size;

**this**.initSquares();

}

**public** **void** initSquares(){

squares = **new** ArrayList<Square>();

**for**(**int** i=1 ; i<= size ; i++){

Square square = **new** SquareNormal(i-1);

squares.add(square);

}

Random random = **new** Random();

**for**(**int** i=0 ; i < HOME\_COUNT ; i++){

**int** index = random.nextInt(size);

**while**(index == 0) index = random.nextInt(size);

squares.set(index, **new** SquareHome(index));

}

**for**(**int** i=0 ; i < FREEZE\_COUNT ; i++){

**int** index = random.nextInt(size);

**while**(index == 0) index = random.nextInt(size);

squares.set(index, **new** SquareFreeze(index));

}

System.***out***.println(">>>>> Generate the terminal of each ladder <<<<<");

**for**(**int** i=0 ; i < LADDER\_COUNT ; i++){

**int** index = random.nextInt(size);

**while**(index == 0) index = random.nextInt(size);

squares.set(index, **new** SquareLadder(index, size));

}

}

**public** Square getInititalSquare(){

**return** **this**.squares.get(0);

}

**public** Square getSquare(**int** i){

**return** squares.get(i);

}

**public** Square changeSquare(**int** value){

**return** **this**.squares.get(value);

}

**public** **int** getSize(){

**return** size;

}

}

**package** game.model;

**Square**

**public** **abstract** **class** Square {

**private** **int** position;

**protected** String name;

**public** Square (**int** position){

**this**.position = position;

}

**public** **int** getPosition(){

**return** position;

}

**public** String getName() {

**return** name + (position+1);

}

**public** **void** land(Player player){

// do something

}

}

**package** game.model;

**SquareFreeze**

**public** **class** SquareFreeze **extends** Square {

**public** SquareFreeze(**int** position) {

**super**(position);

name = "FZ";

}

**public** **void** land(Player player){

System.***out***.println("FREEZE");

player.freeze();

}

}

**package** game.model;

**SquareHome**

**public** **class** SquareHome **extends** Square{

**public** SquareHome(**int** position) {

**super**(position);

name = "HOME";

}

**public** **void** land(Player player){

System.***out***.println("HOME");

player.home();

}

}

**package** game.model;

**SquareLadder**

**import** java.util.Random;

**public** **class** SquareLadder **extends** Square {

**private** **int** terminal;

**public** SquareLadder(**int** position, **int** boardSize) {

**super**(position);

name = "LD";

terminal = determineTerminal(position, boardSize);

System.***out***.println("ladder at = "+(position+1));

System.***out***.println("terminal = "+(terminal+1));

System.***out***.println("--------------------------------------");

}

**public** **int** determineTerminal(**int** position, **int** boardSize){

Random random = **new** Random();

**int** temp = random.nextInt(boardSize);

**while**(temp <= position) temp = random.nextInt(boardSize);

**return** temp;

}

**public** **void** land(Player player){

System.***out***.println("LADDER");

player.ladder(terminal);

}

}

**package** game.model;

**SquareNormal**

**public** **class** SquareNormal **extends** Square {

**public** SquareNormal(**int** position) {

**super**(position);

name = "s";

}

**public** **void** land(Player player){

System.***out***.println("NORMAL");

}

}

**Player**

**package** game.model;

**public** **class** Player {

**private** String name;

**private** Square square;

**private** Board board;

**private** Die die;

**private** **boolean** isFreeze;

**public** Player(String name, Board board, Die die){

**this**.name = name;

**this**.board = board;

**this**.die = die;

**this**.square = board.getInititalSquare();

**this**.isFreeze = **false**;

}

**public** **int** roll(){

die.randomizeValue();

**int** value = die.getValue();

**return** value;

}

**public** **void** walk(**int** value){

**if**(square.getPosition() + value >= board.getSize()){

**int** temp = (square.getPosition() + value) - (board.getSize()-1);

**int** newValue = (board.getSize()-1) - temp;

square = board.changeSquare(newValue);

System.***out***.println(**this**.getName() + " walks to: " + (square.getPosition()+1));

}

**else** {

System.***out***.println(**this**.getName() + " walks to: " + (square.getPosition()+value+1));

square = board.changeSquare(square.getPosition()+value);

}

System.***out***.print("Square Type: ");

square.land(**this**);

**if**(square.getPosition() == board.getSize()-1){

System.***out***.println(">>>>> YOU WON <<<<<");

}

}

**public** Square getSquare(){

**return** square;

}

**public** String getName(){

**return** name;

}

**public** **void** home(){

square = board.changeSquare(board.getInititalSquare().getPosition());

}

**public** **void** freeze(){

**if**(isFreeze == **false**) {

isFreeze = **true**;

}

}

**public** **void** ladder(**int** terminal){

square = board.changeSquare(terminal);

}

**public** **void** setFreeze(**boolean** freeze){

isFreeze = freeze;

}

**public** **boolean** getIsFreeze(){

**return** isFreeze;

}

}

**Die**

**package** game.model;

**import** java.util.Random;

**public** **class** Die {

**private** **int** value;

**public** **void** randomizeValue(){

Random r = **new** Random();

**this**.value= r.nextInt(6) + 1;

}

**public** **int** getValue(){

**return** **this**.value;

}

}

**package** game.model;

**Game**

/\*\*

\* Game model which acts as a controller. I knows about everything inside the

\* game such as Board, Die, Player1, Player2 and delegates most of the work to

\* those objects.

\*

\* **@author** Keeratipong <kee@fameworks.co>

\*

\*/

**public** **class** Game {

// **TODO**: Declare all you models here

**private** Board board;

**private** Die die;

**private** Player player1;

**private** Player player2;

**private** **int** turn;

**public** Game() {

// **TODO** : Initialize all objects here

board = **new** Board(getBoardSize());

die = **new** Die();

player1 = **new** Player("A" , board , die);

player2 = **new** Player("B" , board , die);

turn = 1; // player1 starts

}

/\*\*

\* Make the current player take his/her turn Taking turn can be roll a dice

\* + move to that square

\*/

**public** **void** currentPlayerTakeTurn() {

**if**(turn==1) {

**if**(player1.getIsFreeze()==**false**) {

**int** value = player1.roll();

System.***out***.println(player1.getName() + " rolls: " + value);

player1.walk(value);

}

}

**else** {

**if**(player2.getIsFreeze()==**false**) {

**int** value = player2.roll();

System.***out***.println(player2.getName() + " rolls: " + value);

player2.walk(value);

}

}

System.***out***.println("======================================");

}

/\*\*

\* Change the state of the game to the next turn

\*/

**public** **void** nextTurn() {

**if**(turn == 1){

**if**(player2.getIsFreeze() == **false**) turn = 2;

**else** **if**(player2.getIsFreeze() == **true**) {

turn = 1;

player2.setFreeze(**false**);

}

}

**else** **if**(turn ==2){

**if**(player1.getIsFreeze() == **false**) turn = 1;

**else** **if**(player1.getIsFreeze() == **true**) {

turn = 2;

player1.setFreeze(**false**);

}

}

}

/\*\*

\* Return total number of squares in the board

\*

\* **@return** the number of squares

\*/

**public** **int** getBoardSize() {

// **TODO**: Replace this dummy return value with actual implementation

**return** 36;

}

/\*\*

\* Return the name of the square

\*

\* **@param** i

\* The index of the square in board's collection

\* **@return** The name of the square

\*/

**public** String getSquareNameAtIndex(**int** i) {

// **TODO**: Replace this dummy return value with actual implementation

**return** board.getSquare(i).getName();

}

**public** **int** getSquareColorCodeAtIndex(**int** i) {

**return** -16777216;

}

/\*\*

\* Return the index of the square that player1 is on

\*

\* **@return** The index of the square

\*/

**public** **int** getPlayer1SquareIndex() {

// **TODO**: Replace this dummy return value with actual implementation

**return** player1.getSquare().getPosition();

}

/\*\*

\* Return the name of the square that the player1 is on

\*

\* **@return** The index of the square

\*/

**public** String getPlayer1SquareName() {

// **TODO**: Replace this dummy return value with actual implementation

**return** player1.getSquare().getName();

}

/\*\*

\* Return the index of the square that player2 is on

\*

\* **@return** The index of the square

\*/

**public** **int** getPlayer2SquareIndex() {

// **TODO**: Replace this dummy return value with actual implementation

**return** player2.getSquare().getPosition();

}

/\*\*

\* Return the name of the square that the player1 is on

\*

\* **@return** The index of the square

\*/

**public** String getPlayer2SquareName() {

// **TODO**: Replace this dummy return value with actual implementation

**return** player2.getSquare().getName();

}

/\*\*

\* Return the face value of a die

\*

\* **@return** Die value

\*/

**public** **int** getDieValue() {

// **TODO**: Replace this dummy return value with actual implementation

**return** die.getValue();

}

/\*\*

\* Return the name of the current player who is in this turn

\*

\* **@return** The name of the current player

\*/

**public** String getCurrentPlayerName() {

// **TODO**: Replace this dummy return value with actual implementation

String name = "";

**if**(turn == 1) name = player1.getName();

**else** name = player2.getName();

**return** name;

}

/\*\*

\* Return the name of the square that the current player is on

\*

\* **@return** The name of the square that the player is on

\*/

**public** String getCurrentPlayerSquareName() {

// **TODO**: Replace this dummy return value with actual implementation

String current = "";

**if**(turn == 1) current = getPlayer1SquareName();

**else** current = getPlayer2SquareName();

**return** current;

}

}

**package game.ui;**

**GUI**

**import game.model.Game;**

**import java.awt.BorderLayout;**

**import java.awt.Color;**

**import java.awt.Dimension;**

**import java.awt.Frame;**

**import java.awt.Graphics;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import javax.swing.JButton;**

**import javax.swing.JFrame;**

**import javax.swing.JOptionPane;**

**import javax.swing.JPanel;**

**/\*\***

**\* A stupid & ugly GUI. It depends on only a game object. You can modify it to**

**\* fit your need, although the no modification is needed to make it work.**

**\***

**\* @author Keeratipong <kee@fameworks.co>**

**\***

**\*/**

**public class GUI extends JFrame {**

// Constants

**public** **static** **final** **int** ***BOARD\_SIZE*** = 640;

**public** **static** **final** Color ***BOARD\_COLOR*** = **new** Color(253, 227, 167);

**public** **static** **final** Color ***LINE\_COLOR*** = Color.***WHITE***;

**public** **static** **final** **int** ***PLAYER\_SIZE*** = 20;

**public** **static** **final** Color ***PLAYER\_1\_COLOR*** = Color.***RED***;

**public** **static** **final** Color ***PLAYER\_2\_COLOR*** = Color.***BLUE***;

**public** **static** **final** **int** ***PLAYER\_1\_OFFSET\_X*** = 10;

**public** **static** **final** **int** ***PLAYER\_1\_OFFSET\_Y*** = 15;

**public** **static** **final** **int** ***PLAYER\_2\_OFFSET\_X*** = 10;

**public** **static** **final** **int** ***PLAYER\_2\_OFFSET\_Y*** = 45;

// GUI componenents

**private** JButton rollButton;

**private** BoardUI boardUI;

// Controller

**private** Game game;

**public** GUI() {

**super**("Snake and Ladder Game");

setDefaultCloseOperation(***EXIT\_ON\_CLOSE***);

setResizable(**false**);

setAlwaysOnTop(**true**);

initGame();

initComponents();

}

**private** **void** initGame() {

game = **new** Game();

}

**private** **void** initComponents() {

setLayout(**new** BorderLayout());

// Create components

rollButton = **new** JButton("Roll a dice!");

rollButton.addActionListener(**new** ActionListener() {

// Current player take a turn when clicked

**public** **void** actionPerformed(ActionEvent e) {

game.currentPlayerTakeTurn();

JOptionPane.*showMessageDialog*(GUI.**this**,

"Player: " + game.getCurrentPlayerName() + "\n"

+ "Dice: " + game.getDieValue() + "\n"

+ "Go to: " + game.getCurrentPlayerSquareName());

boardUI.repaint();

game.nextTurn();

}

});

boardUI = **new** BoardUI();

boardUI.setPreferredSize(**new** Dimension(***BOARD\_SIZE***, ***BOARD\_SIZE***));

// Add components to the window

add(rollButton, BorderLayout.***NORTH***);

add(boardUI, BorderLayout.***CENTER***);

pack();

}

// This BoardUI is just for rendering stuff

// Please feel free to fix it if you want to change looks and feels,

// otherwise just leave it alone.

**class** BoardUI **extends** JPanel {

@Override

**public** **void** paint(Graphics g) {

**super**.paint(g);

// Draw board background

g.setColor(***BOARD\_COLOR***);

g.fillRect(0, 0, ***BOARD\_SIZE***, ***BOARD\_SIZE***);

// Draw lines

g.setColor(***LINE\_COLOR***);

**int** squareXCount = (**int**) Math.*sqrt*(game.getBoardSize());

**int** squareWidth = ***BOARD\_SIZE*** / squareXCount;

**for** (**int** i = 0; i < squareXCount; i++) {

g.drawLine(i \* squareWidth, 0, i \* squareWidth, ***BOARD\_SIZE***);

}

**int** squareYCount = (**int**) Math.*sqrt*(game.getBoardSize());

**int** squareHeight = ***BOARD\_SIZE*** / squareYCount;

**for** (**int** i = 0; i < squareYCount; i++) {

g.drawLine(0, i \* squareHeight, ***BOARD\_SIZE***, i \* squareHeight);

}

// Draw square numbers

**for** (**int** i = 0; i < game.getBoardSize(); i++) {

**int** col = i % squareYCount;

**int** row = i / squareXCount;

g.setColor(**new** Color(game.getSquareColorCodeAtIndex(i)));

g.drawString(game.getSquareNameAtIndex(i), col \* squareWidth

+ squareWidth / 2, row \* squareHeight + squareHeight

/ 2);

}

// Draw players

**int** player1SquareIndex = game.getPlayer1SquareIndex();

**int** player1Col = player1SquareIndex % squareYCount;

**int** player1Row = player1SquareIndex / squareXCount;

g.setColor(***PLAYER\_1\_COLOR***);

g.fillOval(player1Col \* squareWidth + ***PLAYER\_1\_OFFSET\_X***, player1Row

\* squareHeight + ***PLAYER\_1\_OFFSET\_Y***, ***PLAYER\_SIZE***,

***PLAYER\_SIZE***);

**int** player2SquareIndex = game.getPlayer2SquareIndex();

**int** player2Col = player2SquareIndex % squareYCount;

**int** player2Row = player2SquareIndex / squareXCount;

g.setColor(***PLAYER\_2\_COLOR***);

g.fillOval(player2Col \* squareWidth + ***PLAYER\_2\_OFFSET\_X***, player2Row

\* squareHeight + ***PLAYER\_2\_OFFSET\_Y***, ***PLAYER\_SIZE***,

***PLAYER\_SIZE***);

}

}

}

**Main**

**package** game.ui;

**public** **class** Main {

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

GUI gui = **new** GUI();

gui.setVisible(**true**);

}

}