NGÀY THỨ HAI:

* TẠO THÊM MỘT THƯ MỤC MỚI thuộc package inpus

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import main.Game;

import main.GamePanel;

public class KeyboardInputs implements KeyListener {

private GamePanel gamePanel;

public KeyboardInputs(GamePanel gamePanel) {

this.gamePanel = gamePanel;

}

@Override

public void keyTyped(KeyEvent e) {

// TODO Auto-generated method stub

}

@Override

public void keyReleased(KeyEvent e) {

// TODO Auto-generated method stub

}

@Override

public void keyPressed(KeyEvent e) {

switch (e.getKeyCode()) {

case KeyEvent.VK\_W:

gamePanel.changeYDelta(-5);

break;

case KeyEvent.VK\_A:

gamePanel.changeXDelta(-5);

break;

case KeyEvent.VK\_S:

gamePanel.changeYDelta(5);

break;

case KeyEvent.VK\_D:

gamePanel.changeXDelta(5);

break;

}

}

}

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.awt.event.MouseMotionListener;

import main.GamePanel;

public class MouseInputs implements MouseListener, MouseMotionListener {

private GamePanel gamePanel;

public MouseInputs(GamePanel gamePanel) {

this.gamePanel= gamePanel;

}

@Override

public void mouseDragged(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseMoved(MouseEvent e) {

gamePanel.setRectPos(e.getX(), e.getY());

}

@Override

public void mouseClicked(MouseEvent e) {

System.out.println("Mouse clicked!");

}

@Override

public void mousePressed(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseReleased(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseEntered(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseExited(MouseEvent e) {

// TODO Auto-generated method stub

}

}

import java.awt.Graphics;

import javax.swing.JPanel;

import inputs.KeyboardInputs;

import inputs.MouseInputs;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

public class GamePanel extends JPanel {

private MouseInputs mouseInputs;

private int xDelta = 100, yDelta = 100;

public GamePanel() {

mouseInputs = new MouseInputs(this);

addKeyListener(new KeyboardInputs(this));

addMouseListener(mouseInputs);

addMouseMotionListener(mouseInputs);

}

public void changeXDelta(int value) {

this.xDelta += value;

repaint();

}

public void changeYDelta(int value) {

this.yDelta += value;

repaint();

}

public void setRectPos(int x, int y) {

this.xDelta = x;

this.yDelta = y;

repaint();

}

public void paintComponent(Graphics g) {

super.paintComponent(g);

g.fillRect(xDelta, yDelta, 200, 50);

}

}

import javax.swing.JFrame;

public class GameWindow {

private JFrame jframe;

public GameWindow(GamePanel gamePanel) {

jframe = new JFrame();

jframe.setSize(400, 400);

jframe.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

jframe.add(gamePanel);

jframe.setLocationRelativeTo(null);

jframe.setVisible(true);

}

}

public class Game {

private GameWindow gameWindow;

private GamePanel gamePanel;

public Game() {

gamePanel = new GamePanel();

gameWindow = new GameWindow(gamePanel);

gamePanel.requestFocus();

}

}

public class MainClass {

public static void main(String[] args) {

new Game();

}

}

* Hai cấu trúc mã nguồn là các lớp java được viết dùng để sử lí sự kiện từ bàn phím và chuột trong một ứng dụng đồ họa sử dụng thư viện Swing của java.
* KeyboardInputs là một lớp implement KeyListener lắng nghe các sự kiện từ bàn phím. Contructor KeyboardInputs(GamePanel gamePanel ) nhận một đối tượng gamePanel và lưu nó để có thể thay đổi vị trí của hình chữ nhật.

@Override

public void keyTyped(KeyEvent e) {

}

@Override

public void keyReleased(KeyEvent e) {

}

@Override

public void keyPressed(KeyEvent e) {

switch (e.getKeyCode()) {

case KeyEvent.VK\_W:

gamePanel.changeYDelta(-5);

break;

case KeyEvent.VK\_A:

gamePanel.changeXDelta(-5);

break;

case KeyEvent.VK\_S:

gamePanel.changeYDelta(5);

break;

case KeyEvent.VK\_D:

gamePanel.changeXDelta(5);

break;

}

}

* Các phương thức trong KeyListenner của java được gọi để xử lí các sự kiện từ bàn phím.

+ keyTyped (KeyEvent e) : Phương thức này được gọi khi một phím đã được nhấn và thả ra(Thường thì không sử dụng nhiều trong việc sử lí các sự kiện từ bàn phím )

+ keyReleased (KeyEnvent e ) : Phương thức này được gọi khi một phím được nhấn và sau đó thả ra

+ keyPressed(KeyEvent e): Phương thức này được gọi khi một phím được nhấn xuống.

@Override

public void mouseDragged(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseMoved(MouseEvent e) {

gamePanel.setRectPos(e.getX(), e.getY());

}

@Override

public void mouseClicked(MouseEvent e) {

System.out.println("Mouse clicked!");

}

@Override

public void mousePressed(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseReleased(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseEntered(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseExited(MouseEvent e) {

// TODO Auto-generated method stub

}

}

+ Đối tượng mousse của bạn triển khai hai giao diện là MouseListenner và MouseMotieonListener đều là phần của API trong java. Dưới đây là giải thích ý nghĩa của từng phương thức mà bạn đã override.

+ mouseClicked(MouseEvent e) : Phương thức này được gọi khi một nút chuột được nhấn và nhả ra 🡺 Nó cung cấp một đối tượng MouseEvent chứa các thông tin về : Sự kiện nhấp chuột, vị trí của chuột( Tọa độ x và y ) và các thông tin khác như nút chuột nào được nhấn.

+ mousePressed (MouseEvent e ) : Được gọi khi một nút chuột được nhấn. Nó cung cấp một đối tượng MouseEvent với thông tin về vị trí và các thông tin liên quan khác.

+ mouseReleased(MouseEvent e) : Khi con trỏ di chuyển vào hoặc ra khỏi khu vực của đối tượng lắng nghe.

+ mouseMoved(MouseEvent e) : Được gọi khi chuột di chuyển trên không gian mà không có bất kì nút nào được bấm. Nó cung cấp thông tin về vị trí mới của chuột.

+ mouseDragged(MouseEvent e) : Được gọi khi một nút chuột được nhấn và di chuyển . Nó cung cấp thông tin về vị trí mới của chuột khi đang kéo một nút chuột.

* Tất cả các phương thức trên đều là các phương thức xử lí sự kiện chuột trong ứng dụng của bạn.