|  |  |
| --- | --- |
| package main;  import java.awt.Graphics;  import entities.Player; 🡺 DÙNG ĐỂ ĐIỀU KHIỂN MỘT THỰC THỂ CỤ THỂ Ở ĐÂY LÀ NHÂN VẬT  public class Game implements Runnable {  private GameWindow gameWindow;    private GamePanel gamePanel;    private Thread gameThread;      /\*- Tạo ra FPS ổn định cho game - \*/  private final int FPS\_SET = 120;      /\*- Update cho trò chơi của mình - \*/  private final int UPS\_S = 200;    /\* - Tạo một thực thể người chơi - \*/  private Player player;      public Game() {  initClasses();  gamePanel = new GamePanel(this);  gameWindow = new GameWindow(gamePanel);      /\* -Nhận được sự kiện từ bàn phím -\*/  gamePanel.requestFocus();  /\*- Khởi chạy luồng - \*/  StartGameLoop();      }  private void initClasses() {  /\* -Chỗ này là khởi tạo nhân vật ở vị trí 200 ,200 \*/  player = new Player(200, 200);    }  private void StartGameLoop() {  gameThread = new Thread(this);  gameThread.start();  }  private void update() {  player.update();  }    public void render(Graphics g) {  player.render(g);  }  @Override  public void run() {    /\* - Biến lưu tru thoi luong moi khung hinh sẽ kéo dài - \*/  double timePerFrame = 1000000000.0/ FPS\_SET;  double timePerUpdate = 1000000000.0/ UPS\_S;    long previousTime = System.nanoTime();  long lastCheck = System.currentTimeMillis();    int frames =0;  int updates =0;    double deltaU =0;  double deltaF =0;    while (true){  long currentTime = System.nanoTime();    deltaU += (currentTime - previousTime) / timePerUpdate;  deltaF += (currentTime - previousTime) / timePerFrame;  previousTime = currentTime;    if (deltaU >= 1) {  update();  updates++;  deltaU--;  }  if (deltaF >= 1) {  gamePanel.repaint();  frames++;  deltaF--;  }    if(System.currentTimeMillis() - lastCheck >= 1000) {  lastCheck = System.currentTimeMillis();  System.out.println("FPS: " + frames+ " | UPS: " + updates);  frames = 0;  updates =0;  }  }  }  /\* -Các hàm ở di chuyển đều nằm trong thực thể "Player" - \*/  /\*- Chúng ta điều khiển hàm game này trong hàm GamePanel -\*/    /\* -Chúng ta dùng hàm này lấy ra player rồi từ đối tượng Game này  gọi ra các hàm nằm trong đối tượng Player từ class GamePanel  \*/  public Player getPlayer() {  return player;  }    public void WindowFocusLost() {  player.resetDirBooleans();  } | * Cái class game này sẽ chứa thực thể của một player mà cái thực thể game này lại được chứa trong một cái gamePanel   gamePanel chứa Game() ----- Game() chứa Player()   * Nhắc lại một số kiến thức : FPS = SỐ KHUNG HÌNH VẼ RA TRONG 1 GIÂY   Số khung hình mà chương trình mong muốn vẽ trong mỗi giây |