**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

--🙢🕮🙠--



**BÁO CÁO**

**CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI:**

|  |  |  |
| --- | --- | --- |
| Nhóm thực hiện | : | Nhóm 30 |
| Lớp | : | CNTT6-K62 |
| Giáo viên hướng dẫn | : | Vũ Huấn |

**QUẢN LÝ BÁN MÁY TÍNH LAPTOP**

|  |
| --- |
| *Hà Nội*, *ngày 25 tháng 4 năm 2023* |

# Lời nói đầu

Đầu tiên chúng em xin gửi lời cảm ơn chân thành đến Trường Đại học Giao thông vận tải đã Đưa môn học “*Công nghệ Java*” vào chương trình giảng dạy. Đặc biệt, chúng em xin chân thành cảm ơn sâu sắc đến thầy Vũ Huấn – giảng viên bộ môn *“Công nghệ Java”* đã dạy dỗ, truyền đạt, tận tình hướng dẫn và giúp đỡ chúng em trong suốt quá trình thực hiện đề tài này. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để chúng em có thể vững bước sau này.

*“Công nghệ Java”* là môn học thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên.

Tuy nhiên với vốn kiến thức, kinh nghiệm còn hạn chế và khả năng tiếp thu thực tế còn nhiều khiếm khuyết. Mặc dù đã rất cố gắng nhưng trong bài tập lớn này khó có thể tránh khỏi những thiếu sót và còn nhiều điểm chưa chính xác, chúng em rất mong nhận được đóng góp ý kiến từ quý thầy cô để bài tập lớn của chúng em được hoàn thiện hơn.

*Chúng em xin chân thành cảm ơn!*

# Mục lục

[Lời nói đầu 1](#_Toc134045232)

[Mục lục 2](#_Toc134045233)

[1. Giới thiệu đề tài: 2](#_Toc134045234)

[2. Tổng quan: 3](#_Toc134045235)

[2.1. Giao diện của trò chơi: 4](#_Toc134045236)

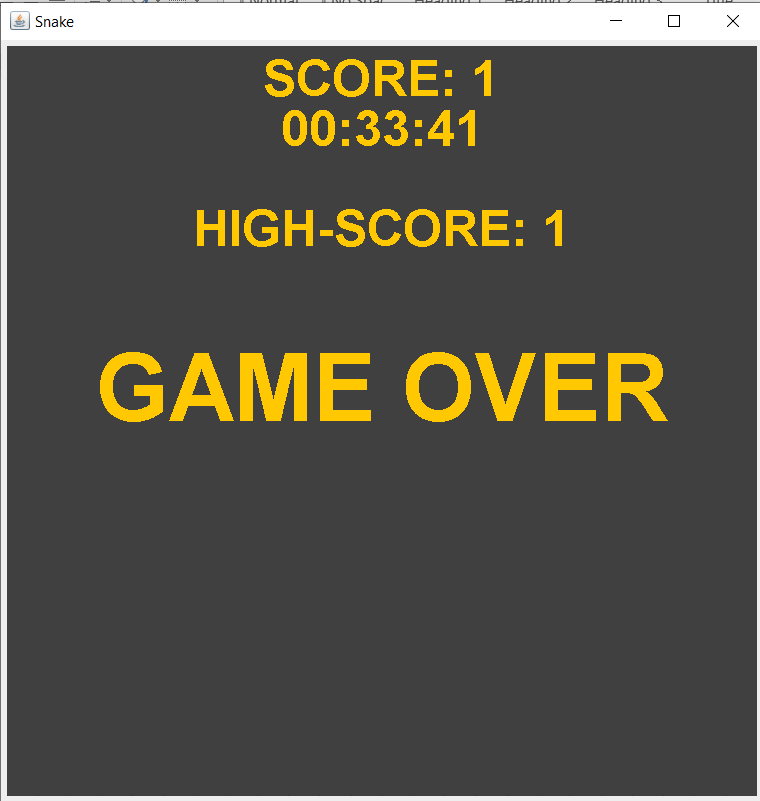
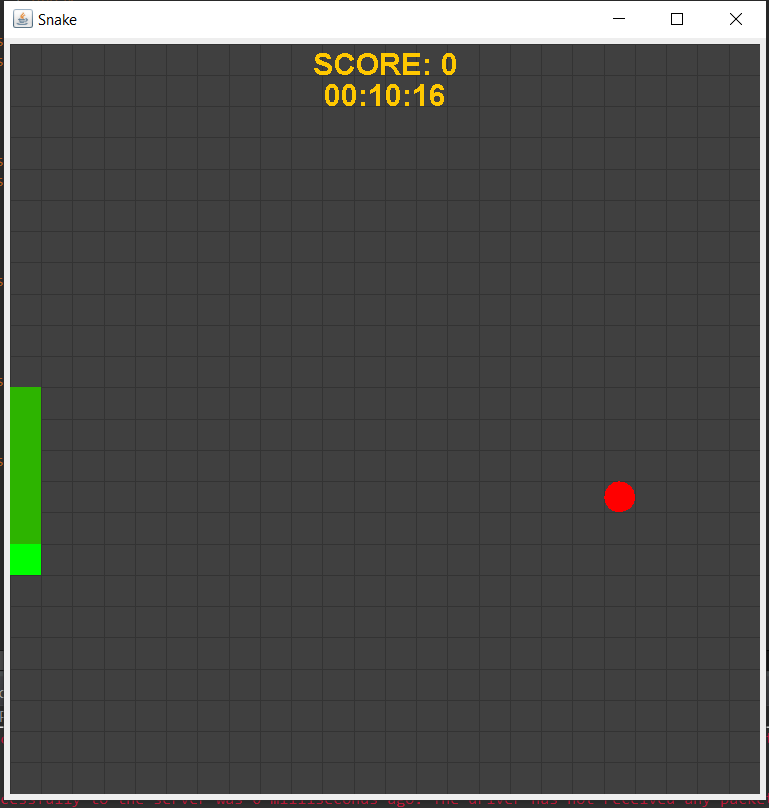
[2.2. Xây dựng code viết chương trình: 4](#_Toc134045237)

[3. Tổng kết: 11](#_Toc134045238)

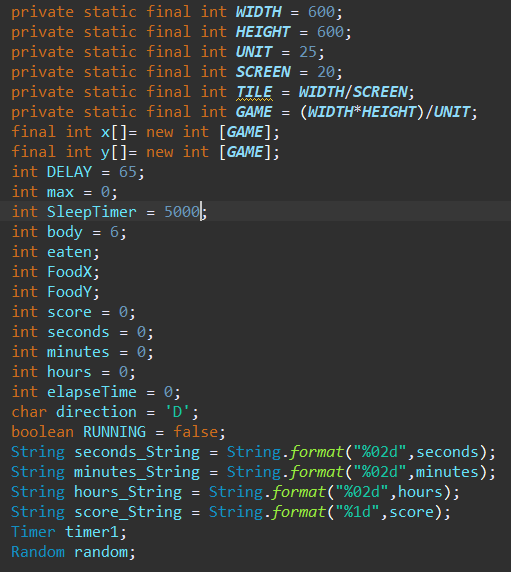
[4. Tài liệu tham khảo: 11](#_Toc134045239)

# Giới thiệu đề tài:

* Mô tả:
* Snake là một trò chơi cổ điển. Lần đầu tiên xuất hiện trong thập  niên 70, sau đó nó đã được đưa lên máy tính. Trong trò chơi này người chơi điều khiển một con rắn, mục tiêu là ăn nhiều càng nhiều chấm càng tốt.
* Khi con rắn ăn một chấm, số đốt thân của nó tăng lên. Con rắn phải tránh những bức tường và cơ thể của chính mình.
* Con rắn được điều khiển bằng phím mũi tên hay các phím W/A/S/D. Ban đầu con rắn có sáu khớp, trò chơi được bắt đầu bằng cách
* bấm một trong các phím mũi tên.
* Nếu trò chơi kết thúc, chúng ta sẽ hiển thị thông báo Game Over ở giữa của màn hình.
* Cách chơi:
* Người chơi dùng các phím mũi tên lên, xuống, trái, phải để điều khiển con rắn ăn được mồi.
* Điều kiện thất bại (GameOver): Con rắn đụng vào vật cản trên đường đi hay cắn phải thân sẽ kết thúc trò chơi.
* Chức năng:
* reset(): tạo một trò chơi mới đồng thời kết thúc trò chơi cũ.
* start(): khởi động trò chơi.
* Timer(): tạo bộ đếm thời gian cho trò chơi.
* draw(): hiện thị con rắn, đồ ăn, điểm … lên màn hình.
* move(): làm cho con rắn di chuyển.
* checkFood(): xem con rắn đã ăn đôg ăn chưa.
* checkCollision(): xem con rắn có bị chạm tường hay thân không.
* gameOver(): hiện thị màn hình kết thúc.
* Yêu cầu:
* Cấu trúc Thread.
* Câu lệnh Try-Catch.
* Kết nối MySQL.

1. **Tổng quan:**
   1. ******Giao diện của trò chơi:**
   2. **Xây dựng code viết chương trình:**

* Khai báo những hằng (constant) sử dụng trong trò chơi:



WIDTH và HEIGHT xác định kích thước của Panel.

UNIT xác định kich thước của các phần tử có trong Panel.

SCREEN xác định độ rộng màn hình.

GAME xác định kích thước của không gian trong Panel.

x[] và y[] xác định vị trí của con rắn.

DELAY là thời gian làm mới hay là tốc độ của trò chơi của trò chơi.

SleepTimer thời gian tạm dừng trò chơi.

body số thân bắt đầu của con rắn.

eaten số táo mà con rắn đã ăn được.

FoodX và FoodY vị trí cuat táo.

Score là điểm.

Eclapse time, hours, minutes, seconds cấu trúc bộ đếm thời gian.

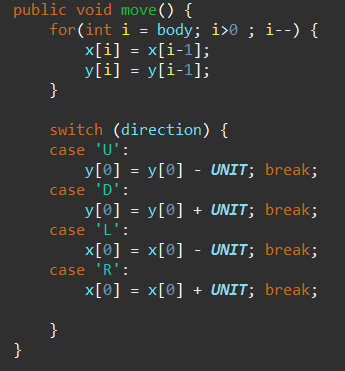
‘D’ hướng di chuỷen ban đầu của con rắn.

RUNNING trạng thái chạy.

Timer bộ đếm thời gian.

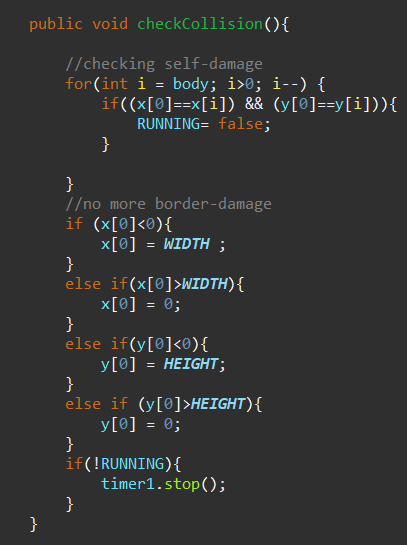
Random tính năng ngẫu nhiên.

* Hàm move() thuật giải chính của trò chơi:



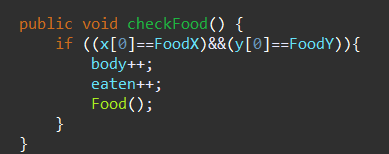
Kiểm soát đầu của con rắn, có thể thay đổi hướng của nó với các phím mũi tên. Phần còn lại của các khớp di chuyển lên một vị trí trong mảng. Khớp thứ 2 di chuyển đến vị trí vừa rồi của khớp thứ nhất, khớp thứ 3 di chuyển đến vị trí vừa rồi của khớp thứ 2...

* Hàm checkCollision()



Trong trường hợp nay con rắn có thể đi xuyên qua khung của màn hình.

* Hàm checkFood()



Khi con rắn chạm vào thức ăn , cơ thể sẽ dài ra và số táo đã ăn sẽ tăng lên

* Giới thiệu một số đoạn code xây dựng chương trình
* public GamePanel() {
* random = new Random();
* this.setPreferredSize(new Dimension(***WIDTH***,***HEIGHT***));
* this.setBackground(Color.***darkGray***);
* this.setFocusable(true);
* this.addKeyListener(new MyKeyAdapter());
* start();
* try {
* String DB\_URL = "jdbc:mysql://localhost:3306/Highscore";
* String USER = "root";
* String PASSWORD = "Brrocoins2003";
* Connection conn = DriverManager.*getConnection*(DB\_URL, USER, PASSWORD);
* String selectSql = "SELECT MAX(Score) FROM Scores";
* java.sql.Statement statement = conn.createStatement();
* ResultSet resultSet = statement.executeQuery(selectSql);
* if (resultSet.next()) {
* eaten = resultSet.getInt(1);
* }
* conn.close();
* }
* catch (SQLException e) {
* System.***err***.println("Error fetching max score: " + e.getMessage());
* }
* }
* //giao dien cuaa mot game moi khi reset
* public void newGame() {
* frame.setTitle("Snake");
* panel.add(new GamePanel());
* frame.add(panel);
* frame.pack();
* frame.setLocationRelativeTo(null);
* frame.setVisible(true);
* frame.setDefaultCloseOperation(JFrame.***DISPOSE\_ON\_CLOSE***);
* try {
* String DB\_URL = "jdbc:mysql://localhost:3307/quanlydiem";
* String USER = "root";
* String PASSWORD = "12345678m";
* Connection conn = DriverManager.*getConnection*(DB\_URL, USER, PASSWORD);
* String selectSql = "SELECT MAX(score) FROM scores";
* java.sql.Statement statement = conn.createStatement();
* ResultSet resultSet = statement.executeQuery(selectSql);
* if (resultSet.next()) {
* eaten = resultSet.getInt(1);
* }
* conn.close();
* }
* catch (SQLException e) {
* System.***err***.println("Error fetching max score: " + e.getMessage());
* }
* }
* //Tro choi bat dau chay, bo dem gio bat dau
* public void start() {
* Food();
* RUNNING = true;
* timer1 = new Timer(DELAY,this);
* timer1.start();
* }
* //bo dem thoi gian
* public void Timer() {
* elapseTime = elapseTime + 1000;
* hours = (elapseTime / 3600000);
* minutes = (elapseTime / 60000) % 60;
* seconds = (elapseTime / 1000) % 60;
* seconds\_String = String.*format*("%02d",seconds);
* minutes\_String = String.*format*("%02d",minutes);
* hours\_String = String.*format*("%02d",hours);
* }
* public void paintComponent(Graphics g) {
* super.paintComponent(g);
* draw(g);
* }
* public void draw (Graphics g) {
* if (RUNNING) {
* //Draw lines
* for(int i = 0; i < ***HEIGHT***/***UNIT***; i++) {
* g.drawLine(i\****UNIT***, 0, i\****UNIT***, ***HEIGHT***);
* g.drawLine(0, i\****UNIT***, ***WIDTH***, i\****UNIT***);
* }
* //Draw Food
* g.setColor(Color.***red***);
* g.fillOval(FoodX, FoodY, ***UNIT***, ***UNIT***);
* //Draw Snake
* for(int i = 0; i < body; i++) {
* if(i==0) {
* g.setColor(Color.***green***);
* g.fillRect(x[i], y[i], ***UNIT***,***UNIT***);
* }
* else {
* g.setColor(new Color(45,180,0));
* g.fillRect(x[i], y[i], ***UNIT***,***UNIT***);
* }
* }
* //Score
* g.setColor(Color.***orange***);
* g.setFont(new Font("CALIBRE", Font.***BOLD***, 25));
* FontMetrics metrics1 = getFontMetrics(g.getFont());
* g.drawString("SCORE: "+eaten,(***WIDTH***-metrics1.stringWidth("SCORE: "+eaten))/2,g.getFont().getSize());
* // Timer
* g.setColor(Color.***orange***);
* g.setFont(new Font("CALIBRE", Font.***BOLD***, 25));
* FontMetrics metrics0 = getFontMetrics(g.getFont());
* g.drawString(hours\_String+":"+minutes\_String+":"+seconds\_String,(***WIDTH***-metrics0.stringWidth(hours\_String+":"+minutes\_String+":"+seconds\_String))/2,g.getFont().getSize()\*2);
* try {
* String DB\_URL = "jdbc:mysql://localhost:3307/quanlydiem";
* String USER = "root";
* String PASSWORD = "12345678m";
* Connection conn = DriverManager.*getConnection*(DB\_URL, USER, PASSWORD);
* String insertSql = "INSERT INTO scores (player\_name, score, levels) VALUES (?, ?, ?)";
* PreparedStatement preparedStatement = conn.prepareStatement(insertSql);
* preparedStatement.setInt(2, eaten);
* preparedStatement.setInt(3, elapseTime);
* preparedStatement.executeUpdate();
* conn.close();
* } catch (SQLException e) {
* System.***err***.println("Error" + e.getMessage());
* }
* }
* else {
* GameOver(g);
* }
* }
* public void move() {
* for(int i = body; i>0 ; i--) {
* x[i] = x[i-1];
* y[i] = y[i-1];
* }
* switch (direction) {
* case 'U':
* y[0] = y[0] - ***UNIT***; break;
* case 'D':
* y[0] = y[0] + ***UNIT***; break;
* case 'L':
* x[0] = x[0] - ***UNIT***; break;
* case 'R':
* x[0] = x[0] + ***UNIT***; break;
* }
* }
* //hien thi thuc an o mot vi tri bat ki
* public void Food() {
* FoodX= random.nextInt((int)(***WIDTH***/***UNIT***))\****UNIT***;
* FoodY= random.nextInt((int)(***HEIGHT***/***UNIT***))\****UNIT***;
* }
* public void checkFood() {
* if ((x[0]==FoodX)&&(y[0]==FoodY)){
* body++;
* eaten++;
* Food();
* }
* }
* public void checkScore() {
* if(eaten>max) {
* max = eaten;
* }
* }
* public void checkCollision(){
* //checking self-damage
* for(int i = body; i>0; i--) {
* if((x[0]==x[i]) && (y[0]==y[i])){
* RUNNING= false;
* }
* }
* //no more border-damage
* if (x[0]<0){
* x[0] = ***WIDTH*** ;
* }
* else if(x[0]>***WIDTH***){
* x[0] = 0;
* }
* else if(y[0]<0){
* y[0] = ***HEIGHT***;
* }
* else if (y[0]>***HEIGHT***){
* y[0] = 0;
* }
* if(!RUNNING){
* timer1.stop();
* }
* }
* //hien thi khi game ket thuc
* public void GameOver(Graphics g) {
* //Score
* g.setColor(Color.***orange***);
* g.setFont(new Font("CALIBRE", Font.***BOLD***, 40));
* FontMetrics metrics1 = getFontMetrics(g.getFont());
* g.drawString("SCORE: "+eaten,(***WIDTH***-metrics1.stringWidth("SCORE: "+eaten))/2,g.getFont().getSize());
* //High-score
* g.setColor(Color.***orange***);
* g.setFont(new Font("CALIBRE", Font.***BOLD***, 40));
* FontMetrics metrics4 = getFontMetrics(g.getFont());
* g.drawString("HIGH-SCORE: "+max,(***WIDTH***-metrics4.stringWidth("HIGH-SCORE: "+max))/2,g.getFont().getSize()\*4);
* //Timer
* g.setColor(Color.***orange***);
* g.setFont(new Font("CALIBRE", Font.***BOLD***, 40));
* FontMetrics metrics2 = getFontMetrics(g.getFont());
* g.drawString(hours\_String+":"+minutes\_String+":"+seconds\_String,(***WIDTH***-metrics2.stringWidth(hours\_String+":"+minutes\_String+":"+seconds\_String))/2,g.getFont().getSize()\*2);
* //Game Over
* g.setColor(Color.***orange***);
* g.setFont(new Font("CALIBRE", Font.***BOLD***, 75));
* FontMetrics metrics3 = getFontMetrics(g.getFont());
* g.drawString("GAME OVER",(***WIDTH***-metrics3.stringWidth("GAME OVER"))/2, ***HEIGHT***/2);
* }
* *@Override*
* public void actionPerformed(ActionEvent e) {
* if(RUNNING) {
* move();
* checkFood();
* checkCollision();
* checkScore();
* }
* repaint();
* Timer();
* }
* //tam dung tro choi
* public void pause() {
* try {
* Thread.*sleep*(SleepTimer);
* }
* catch (InterruptedException e) {
* e.getStackTrace();
* }
* }
* public void resume() {
* Thread.*currentThread*().interrupt();
* }
* public void CloseGame(KeyEvent e) {
* JComponent comp = (JComponent) e.getSource();
* Window win = SwingUtilities.*getWindowAncestor*(comp);
* win.dispose();
* }
* public void reset(){
* EventQueue.*invokeLater*(()->
* {
* GamePanel game = new GamePanel();
* game.newGame();
* });
* }

1. **Tổng kết:**

* Trong thời gian nghiên cứu và vận dụng những kiến thức đã được trang bị ở nhà

trường và các lĩnh vực khác.

* Chúng em đã viết được game rắn săn mồi nhằm mục đích giải trí lành mạnh.
* Chúng em hy vọng rằng những khái niệm này là tiền đề giúp chúng em phát triển mạnh hơn về những thể loại game khác chất lượng hơn, hay hơn để phục vụ cho nhu cầu trong cuộc sống chúng ta.
* Trong quá trình thực hiên đề tài, chúng em đã nhận được sự giúp đỡ tận tình của Vũ Huấn giúp chúng em hoàn thành tốt bài báo cáo này.
* Vì thời gian làm báo cáo hạn hẹp, kinh nghiệm thực tế chưa nhiều nên báo cáo không tránh khỏi những sai sót, rất mong sự góp ý của thầy và các bạn.

1. **Tài liệu tham khảo:**

Powerpoint, slide bài giảng của thầy Vũ Huấn.

Youtube,

Stackoverflow.