TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO BÀI TẬP LỚN**

*Đề tài:* Trò chơi Flappy Bird

Người thực hiện: Nguyễn Nho Hùng – Hoàng Tuấn Đạt

Lớp: Công nghệ thông tin 6 – K62

Bộ môn: Công nghệ JAVA

Hà Nội, ngày 3 tháng 5 năm 2023

**1, Giới thiệu và cách chơi game**

* Tên game: Flappy Bird
* Cách chơi: Click chuột để bắt đầu và di chuyển chim. Mục tiêu của trò chơi là click chuột để chim bay qua các chướng ngại vật.
* Ưu điểm trò chơi: nhẹ, dễ chơi.
* Nhược điểm: còn đơn giản, chưa có hiệu ứng âm thanh, đồ họa chưa đẹp.

**2, Quá trình làm game**

* Tạo 3 class FlappyBird , Renderer và Main
* Class FlappyBird **implements** ActionListener, MouseListener

+ Thuộc tính:

FlappyBird *flappyBird*

**int** WIDTH = 800, HEIGHT = 800 -> chiều rộng và cao của màn hình game

Rectangle bird -> chim là hình chữ nhật

ArrayList<Rectangle> columns -> 1 dãy các chướng ngại vật là hcn

**int** ticks, yMotion, score; -> tích tắc và chuyển động của tọa độ y, điểm

**boolean** gameOver, started

Renderer renderer -> ket xuat

Random rand

+ Các phương thức:

* addColumn(**boolean** start), paintColumn(Graphics g, Rectangle column) là 2 phương thức để thêm và tô màu cho các chướng ngại vật là các hình chữ nhật có màu green.darker() và width = 100, height = 50 + rand.nextInt(300), khoảng cách giữa các chướng ngại vật = 300
* hàm tạo FlappyBird()

{

JFrame jframe = **new** JFrame();

Timer timer = **new** Timer(20, **this**);// Timer(int delay, ActionListener listener) -> tốc độ di chuyển của trò chơi

renderer = **new** Renderer();

rand = **new** Random();

jframe.add(renderer);

jframe.setDefaultCloseOperation(jframe.***EXIT\_ON\_CLOSE***); -> khi thoát trò chơi thì chương trình cũng thoát

jframe.setSize(WIDTH, HEIGHT); -> set chiều dài và rộng của game

jframe.setTitle("Flappy Bird"); -> set tiêu đề

jframe.setVisible(**true**);

jframe.setResizable(**false**); -> ko thể phóng to thu nhỏ

jframe.addMouseListener(**this**); -> các sự kiện về chuột

bird = **new** Rectangle(WIDTH / 2 - 10, HEIGHT / 2 - 10, 20, 20); // tạo tọa độ, width, height cho bird

columns = **new** ArrayList<Rectangle>(); // tạo các chướng ngại vật ngay sau bird

addColumn(**true**);

addColumn(**true**);

addColumn(**true**);

addColumn(**true**); -> tạo 4 chướng ngại vật 1 lúc

timer.start();  
}

* jump() -> tí nữa để bắt sự kiện khi click chuột - mouseClicked(MouseEvent e){ jump(); }

**if** (gameOver) { -> khi game kết thúc -> click chuột để chơi lại

gameOver = **false**;

bird = **new** Rectangle(WIDTH / 2 - 10, HEIGHT / 2 - 10, 20, 20); // tạo tọa độ, width, height cho bird

columns.clear();

yMotion = 0;

score = 0;

addColumn(**true**);

addColumn(**true**);

addColumn(**true**);

addColumn(**true**);

}

**if** (!started) { -> khi chưa bắt đầu -> click để bắt đầu

started = **true**;

} **else** **if** (!gameOver) {

**if** (yMotion > 0) {

yMotion = 0;

}

yMotion -= 10; -> độ cao chim nhảy

}

@Override

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

ticks++;

**int** speed = 10;

**if** (started) {

**for** (**int** i = 0; i < columns.size(); i++) {

Rectangle column = columns.get(i);

column.x -= speed; -> tọa độ của các cột giảm đi 10 nghĩa là các cột ngại vật sẽ di chuyển từ phải qua trái

}

**if** (ticks % 2 == 0 && yMotion < 15) {

yMotion += 2; -> tốc độ click chuột

}

**for** (**int** i = 0; i < columns.size(); i++) {

Rectangle column = columns.get(i);

**if** (column.x + column.y < 0) {

// columns.remove(column); // khi các cột đã qua -> xóa cột

**if** (column.y == 0) {

addColumn(**false**);

}

}

}

bird.y += yMotion;

**for** (Rectangle column : columns) {

**if** (column.y == 0 && bird.x + bird.width / 2 > column.x + column.width / 2 - 10

&& bird.x + bird.width / 2 < column.x + column.width / 2 + 10) { -> khi chim ở giữa 2 cột thì điểm + 1

score++;

}

**if** (column.intersects(bird)) // phát hiện điểm va chạm

{

gameOver = **true**;

bird.x = column.x - bird.width; -> chim sẽ rơi xuống và bị cột đẩy đi

}

}

**if** (bird.y >= HEIGHT - bird.height - 120 || bird.y < 0) { -> điều kiện chim chạm cột -> game over

gameOver = **true**;

}

**if** (bird.y + yMotion >= HEIGHT - 120) {

bird.y = HEIGHT - 120 - bird.height;

}

}

renderer.repaint();

* Repaint(Graphics g)

g.setColor(Color.***cyan***);

g.fillRect(0, 0, WIDTH, HEIGHT); // được sử dụng để điền màu mặc định và độ rộng và chiều cao đã cho vào hình

// chữ nhật

g.setColor(Color.***orange***);

g.fillRect(0, HEIGHT - 120, WIDTH, 150); // tô màu cho đất

g.setColor(Color.***green***);

g.fillRect(0, HEIGHT - 120, WIDTH, 20);

g.setColor(Color.***red***);

g.fillRect(bird.x, bird.y, bird.width, bird.height); // tô màu cho bird

**for** (Rectangle column : columns) { // tô màu cho chướng ngại vật

paintColumn(g, column);

}

g.setColor(Color.***white***);

g.setFont(**new** Font("Arial", 1, 100)); // Font(String name, int style, int size)

**if** (!started) {

g.drawString("Click to Start!!", 60, HEIGHT / 2 - 50); // tọa độ Start

}

**if** (gameOver) {

g.drawString("Game Over!", 100, HEIGHT / 2 - 50); // tọa độ Game Over

}

**if** (!gameOver && started) { // hiển thị điểm

g.drawString(String.*valueOf*(score), WIDTH / 2 - 25, 100);

}

g.setColor(Color.***white***);

g.setFont(**new** Font("Arial", 1, 70)); // Font(String name, int style, int size)

**if** (gameOver) {

g.drawString("Score: " + String.*valueOf*(score), WIDTH / 2 - 130, HEIGHT / 2 + 50); // hiển thị điểm của bạn

// khi gameOver

**Class Main**:

**public** **class** Main **extends** FlappyBird {

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

*flappyBird* = **new** FlappyBird();

}

}

**Class Renderer**:

**public** **class** Renderer **extends** JPanel {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

@Override

**protected** **void** paintComponent(Graphics g) {

**super**.paintComponent(g);

FlappyBird.*flappyBird*.repaint(g);

}

}

PHÂN CHIA CÔNG VIỆC TRONG NHÓM

* Nguyễn Nho Hùng viết code, chỉnh sửa
* Hoàng Tuấn Đạt viết code, viết báo cáo

TÀI LIỆU THAM KHẢO

1. Tài liệu lập trình Java - [[PDF] DOWNLOAD 4 tài liệu lập trình Java cơ bản bằng Tiếng Việt (codegym.vn)](https://codegym.vn/blog/2021/02/01/tai-lieu-lap-trinh-java/).
2. Một số nguồn bên ngoài khác.