

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

BÀI TẬP LỚN

MÔN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

ĐỀ TÀI: TRÒ CHƠI FLAPPY BIRDS

Sinh viên thực hiện : Nguyễn Viết Tuấn
Hà Tuấn Mạnh
Lê Ngọc Anh Tuấn
Nguyễn Đình Quyền
Lớp : Kỹ thuật điện tử và tin học – Khóa QH.2022.T.CQ
Giáo viên hướng dẫn : Bùi Việt Hà

Hà Nội, tháng 11 năm 2023

MỤC LỤC

LỜI NÓI ĐẦU	3
PHÂN CÔNG THÀNH VIÊN TRONG NHÓM	4
CHƯƠNG 1. KHẢO SÁT, ĐẶC TẢ BÀI TOÁN.....	5
1. Mô tả yêu cầu bài toán	5
2. Biểu đồ use case	5
3. Đặc tả use case	5
CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ BÀI TOÁN	6
1. Biểu đồ lớp.....	6
2. package <code>pkd2dgamesframework</code>	6
3. Package <code>flappybirds</code>	7
CHƯƠNG 3: CÔNG NGHỆ VÀ THUẬT TOÁN SỬ DỤNG	10
Sử dụng các thư viện:.....	10
CHƯƠNG 4. XÂY DỰNG CHƯƠNG TRÌNH MINH HỌA.....	11
1. Kết quả chương trình minh họa.....	11
2. Giao diện chương trình.....	11
3. Kết luận	12
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	13
1. Kết luận:	13
2. Hướng phát triển:	14
TÀI LIỆU THAM KHẢO	15

LỜI NÓI ĐẦU

Lý do chúng em chọn đề tài Flappy Birds.

+ Ấn tượng đầu tiên để bọn em nghĩ đến chọn Flappy Birds vì là tựa game của Việt Nam sản xuất.

+ Giao diện đơn giản: Flappy Bird có rất ít nút chức năng. Nếu tính luôn các chức năng chia sẻ cộng đồng, thì tổng cộng chỉ có 7 nút bấm mà chỉ cần 3 trong số đó là đủ để chơi game trọn vẹn. Đây là lựa chọn thông minh, khi hiểu cảm giác choáng ngợp của người chơi khi phải đối mặt với hàng chục tùy chọn mà các game khác.

+ Khó ngay từ đầu: Ấn tượng đầu tiên khi chơi Flappy Bird chính là độ khó kinh dị của nó. Rất nhiều người dù là các game thủ "gạo cội" vẫn không thể qua khỏi cái ống cống chướng ngại vật ngay lần chơi đầu tiên. Độ khó "khủng bố" này khiến người chơi bức bối, nhưng lại rất hài hước. Nếu lướt qua các bình luận về Flappy Bird, chúng ta có thể thấy được có vô số người vừa yêu vừa ghét game này. Có lẽ nguyên nhân lớn nhất khiến nó thành công là: Người ta thích cảm giác "thử thách và chinh phục" giới hạn của bản thân mình vì độ khó của nó và cùng chia sẻ với các người chơi khác.

+ Cách chơi đơn giản: Cách chơi của Flappy Bird vô cùng đơn giản. Chúng ta chỉ cần đưa tay ra "Tap tap tap" vào màn hình là có thể di chuyển được chú chim hay là click chuột trên máy tính, chứ không có nhiều kiểu combo phức tạp như những tựa game hiện đại. Tuy cách chơi đơn giản nhưng để chơi hay được hay Flappy Bird thì người chơi phải bỏ ra nhiều thời gian luyện tập để vượt qua những cây cột. Điều này là một sự phối hợp hoàn mỹ với độ khó cao của trò chơi: dễ chơi nhưng khó chơi hay.

Những yếu tố này đã tạo cảm hứng để nhóm tạo ra trò chơi này.

PHÂN CÔNG THÀNH VIÊN TRONG NHÓM

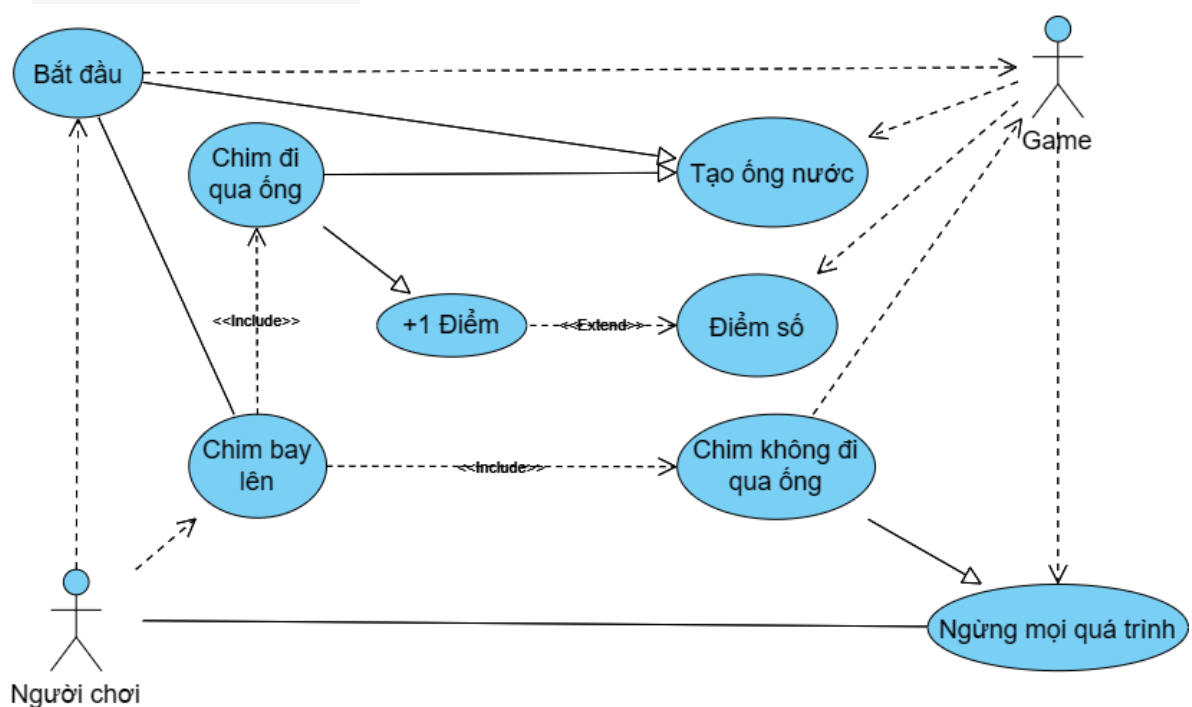
Họ và tên	Email	Công việc thực hiện	Đánh giá
Nguyễn Viết Tuấn	nguyenviettuang_t67@hus.edu.vn	Lên ý tưởng, thuyết trình, viết báo cáo	Tốt
Hà Tuấn Mạnh	hatuanmanh_t67@hus.edu.vn	Viết code, đóng góp ý tưởng thiết kế	Tốt
Lê Ngọc Anh Tuấn	lengocanhtuang_t67@hus.edu.vn	Sửa lỗi, cải thiện đồ họa	Tốt
Nguyễn Đình Quyền	nguyendinhquyeng_t67@hus.edu.vn	Tìm tài liệu tham khảo, đóng góp ý tưởng thiết kế	Tốt

CHƯƠNG 1. KHẢO SÁT, ĐẶC TẢ BÀI TOÁN

1. Mô tả yêu cầu bài toán

- Tạo cửa sổ game và vẽ nền cho game
- Tạo con chim và điều khiển nó bay lên
- Chuyển động rơi tự do của chim
- Tạo cột và chuyển động của ống nước
- Xử lý va chạm của chim và ống nước
- Tính điểm
- Tạo hiệu ứng âm thanh trong các quá trình

2. Biểu đồ use case

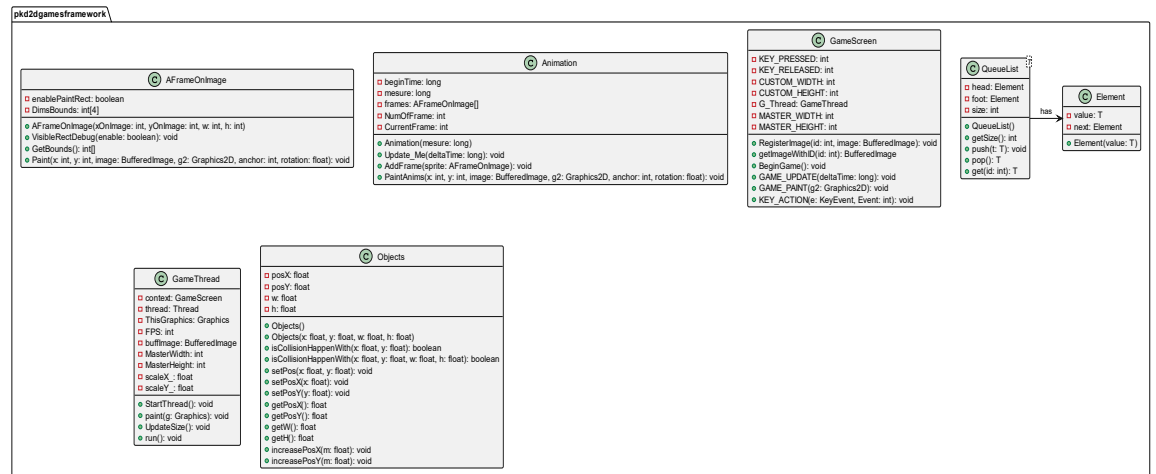


3. Đặc tả use case

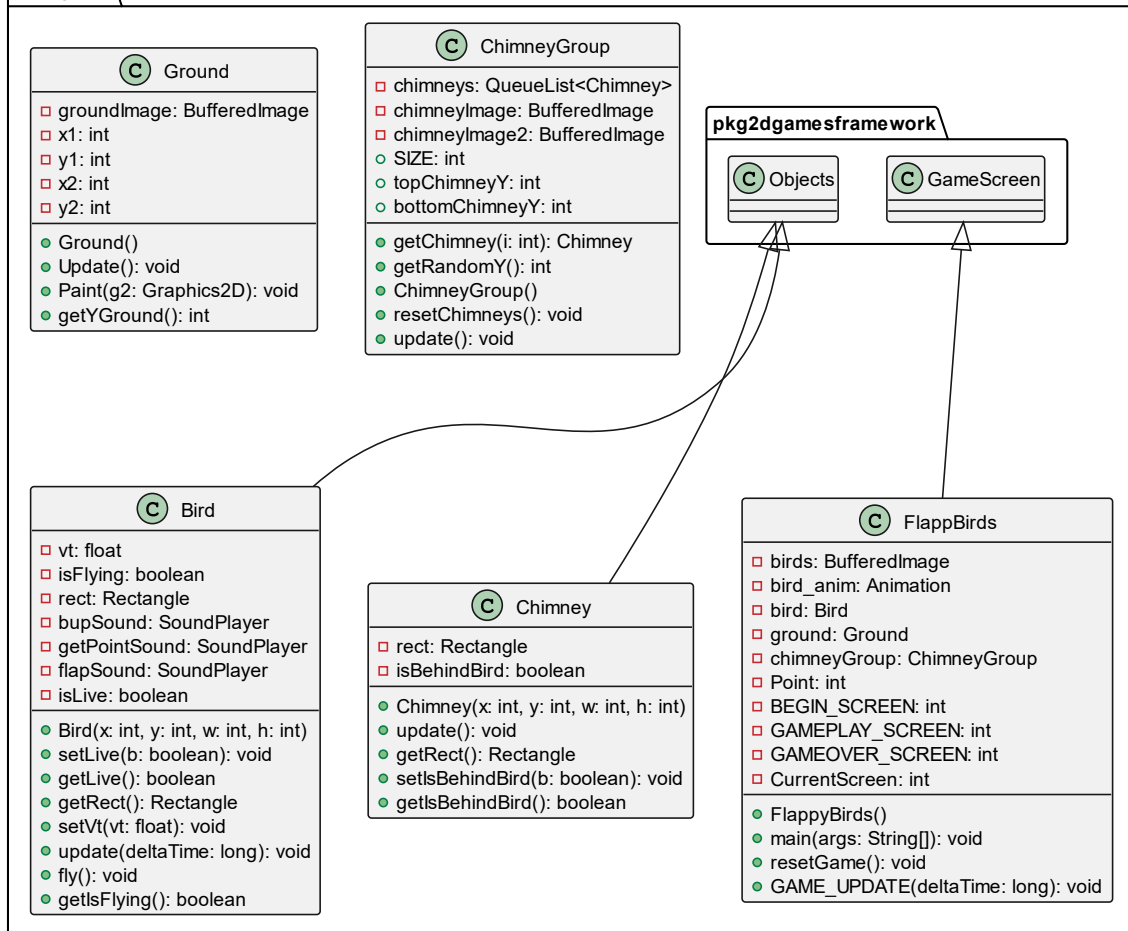
- Khi người chơi bắt đầu game, game sẽ tạo ống nước và đặt điểm số là 0, và con chim rơi tự do.
- Khi người chơi bấm Space, con chim bay lên một khoảng cách nhất định
- Khi con chim đi qua khe hở giữa 2 cột ống nước, game sẽ cộng 1 điểm vào điểm số
- Game sẽ tạo ống nước liên tục, khoảng cách giữa 2 ống nước là cố định
- Khi con chim không đi qua ống nước, mọi quá trình sẽ dừng lại và cho phép người chơi tạo ván mới.

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ BÀI TOÁN

1. Biểu đồ lớp



flappybird



2. package pkd2dgamesframework

Đây là thư viện thiết lập các yếu tố cơ bản để nhanh chóng gọi các lệnh hành động khi lập trình package **flappybird**

- + **AFrameOnImage**: xuất hình ảnh đơn, nhiều hình đơn gộp lại sẽ thành hiệu ứng
- + **Animation**: tạo hiệu ứng từ việc ghép hình ảnh đơn
- + **GameScreen**: tạo giao diện hình ảnh
- + **GameThread**: tạo vòng lặp cho các quá trình trong game như tạo ống nước, phần nền đất,...
- + **QueueList**: lưu trữ một mảng chứa các đối tượng cùng loại như ống nước,...
- + **SoundPlayer**: tạo âm thanh

3. Package **flappybird**

- a. lớp **Bird**: tạo và vận hành con chim

Thuộc tính:

- **vt**: float: gán độ cao của chim
- **isFlying**: boolean: gán chim có bay hay không
- **rect**: Rectangle: gán vùng cấm bay của chim
- **bupSound**: SoundPlayer: âm thanh khi chim vượt qua giới hạn bay
- **getPointSound**: SoundPlayer: âm thanh khi được cộng điểm
- **flapSound**: SoundPlayer: âm thanh khi chim bay lên
- **isLive**: boolean: gán chim còn sống hay không
- + Constructor **Bird(x: int, y: int, w: int, h: int)**: thiết lập chim và các thuộc tính

Phương thức:

- + **setLive(b: boolean)**: thiết lập việc sống và chết của chim
- + **getLive()**: boolean: trả về xem chim còn sống không
- + **getRect()**: Rectangle: trả về vùng cấm bay
- + **setVt(vt: float)**: thiết lập độ cao của chim
- + **update(deltaTime: long)**: gán độ cao mới vào thuộc tính vt, isFlying sau một khoảng thời gian.
- + **fly()**: tăng độ cao của chim

+ **getIsFlying()**: boolean: xem chim có bay không

b. Lớp **Chimney**:

Thuộc tính

- **isBehindBird**: cho biết liệu ống nước có bị con chim đâm vào hay không.

+ Constructor **Chimney**: khởi tạo vị trí và kích thước của ống nước.

Phương thức:

+ **update()**: Cập nhật vị trí của ống nước bằng cách di chuyển nó sang trái (với vận tốc 2 pixel mỗi khung hình).

+ **getRect()**: Trả về vùng cấm bay bao quanh ống nước.

+ **setIsBehindBird(boolean b)**: Thiết lập để chỉ ra liệu ống nước có bị sau con chim đâm vào không.

+ **getIsBehindBird()**: Trả về giá trị của setIsBehindBird

c. Lớp **ChimneyGroup**:

Thuộc tính

- **chimneys**: QueueList<Chimney>: tạo sách ống nước

- **chimneyImage**: BufferedImage: gán hình ảnh ống nước ở dưới

- **chimneyImage2**: BufferedImage: gán hình ảnh ống nước ở trên

- **topChimneyY**: gán ống nước ở trên

- **bottomChimneyY**: gán ống nước ở dưới

Phương thức

+ **getChimney(i: int)**: Chimney: trả về danh sách ống nước đang hiển thị

+ **getRandomY()**: int: trả về khe hở giữa 2 ống nước trên dưới

+ Constructor: **ChimneyGroup()**: khởi tạo nhóm ống nước

+ **resetChimneys()**: thiết lập lại nhóm ống nước

+ **update()**: cập nhật vị trí ống nước sau một khoảng thời gian

+ **paint()**: hiển thị ống nước mới lên màn hình

d. Lớp **Ground**:

Thuộc tính

- **groundImage** là hình ảnh nền đất.
 - **x1, y1, x2, y2** là tọa độ của hai phần nền đất (để tạo hiệu ứng di chuyển).
- + Constructor **Ground()**: khởi tạo nền đất

Phương thức

- + **Update()** cập nhật vị trí của nền đất.
 - + Phương thức **Paint(Graphics2D g2)** vẽ nền đất lên màn hình.
 - + **getYGround()** trả về tọa độ Y của nền đất.
- e. Lớp **FlappyBirds**:

Thuộc tính:

- **birds**: BufferedImage: gán hình ảnh tĩnh của chim
- **bird_anim**: Animation: gán chuyển động bay của chim
- **bird**: Bird: gán một con chim
- **ground**: Ground: gán phần nền đất
- **chimneyGroup**: ChimneyGroup: gán tập hợp các ống
- **Point**: int: gán điểm số
- **BEGIN_SCREEN**: int: gán màn hình bắt đầu
- **GAMEPLAY_SCREEN**: int: gán màn hình khi đang chơi
- **GAMEOVER_SCREEN**: int: gán màn hình khi trò chơi kết thúc
- **CurrentScreen**: int: gán màn hình khi trò chơi đang vận hành

Phương thức

- + **resetGame()**: đưa game về trạng thái bắt đầu
- + **FlappyBirds()**: tạo cửa sổ giao diện khi game khởi động
- + **main(String[] args)**: khởi động game

CHƯƠNG 3: CÔNG NGHỆ VÀ THUẬT TOÁN SỬ DỤNG

Sử dụng các thư viện:

1. `java.awt.*;`
2. `javax.swing.*`
3. `java.io.File`
4. `java.io.IOException;`
5. `javax.imageio.ImageIO`
6. `javax.sound.sampled.*`
7. `java.util.Random`

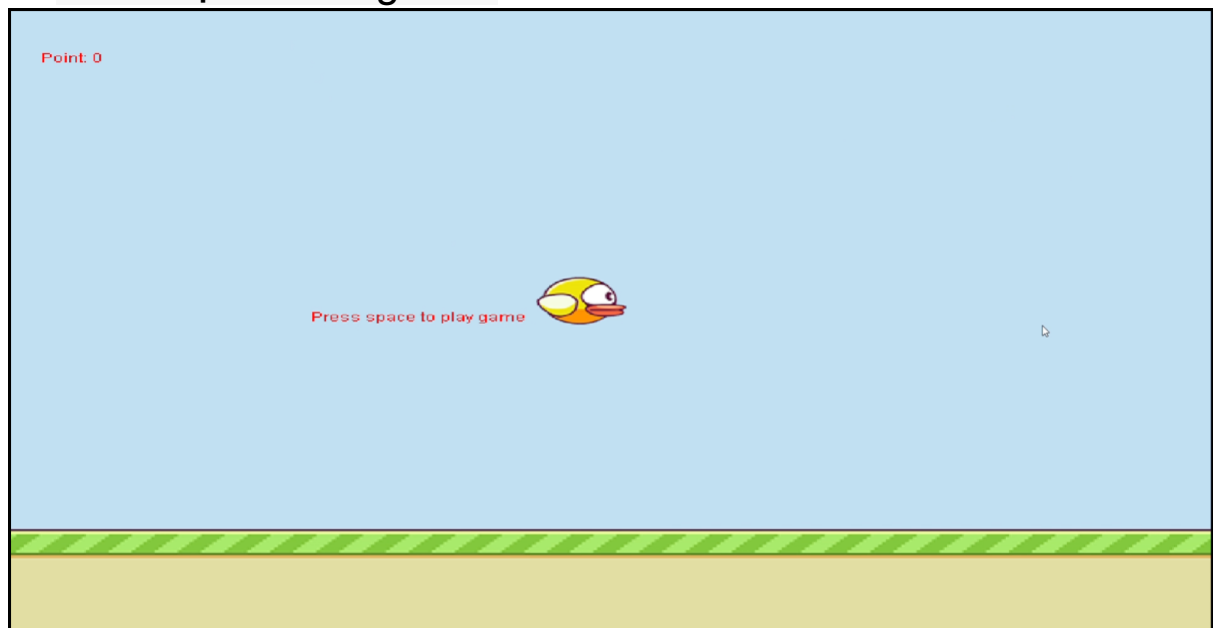
CHƯƠNG 4. XÂY DỰNG CHƯƠNG TRÌNH MINH HỌA

1. Kết quả chương trình minh họa

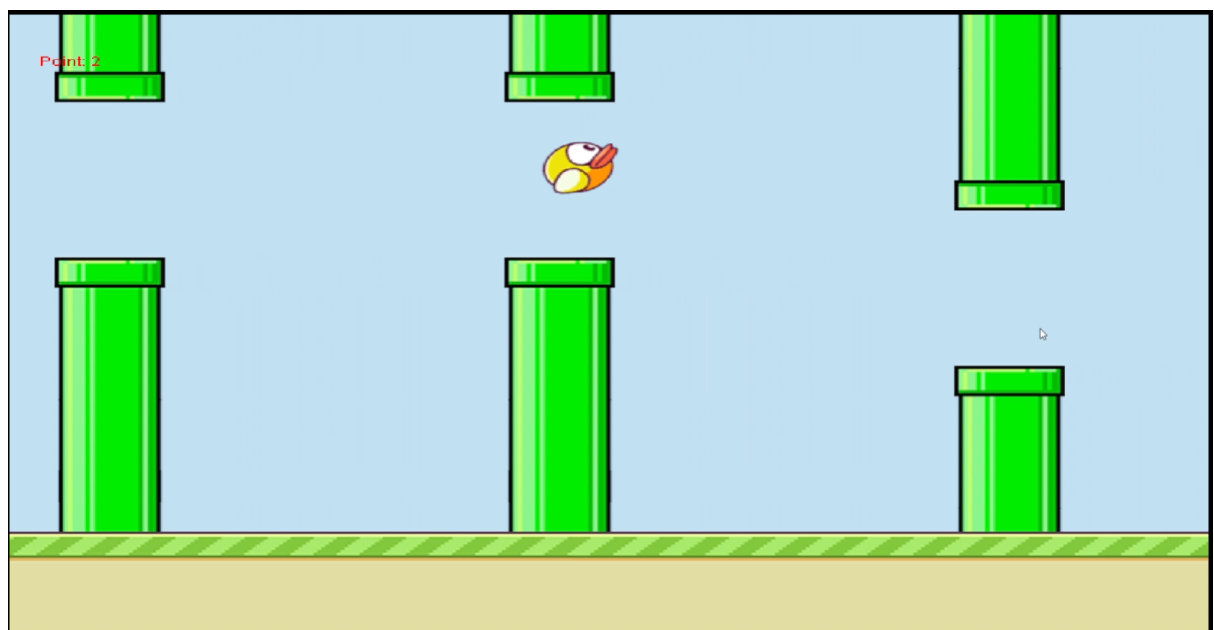
Chương trình đã chạy được các tính năng cơ bản đã được nêu ở phần 1.1

- + Hiển thị giao diện game
- + Điều khiển chim đi qua ống nước
- + Có hiệu ứng âm thanh trong các quá trình

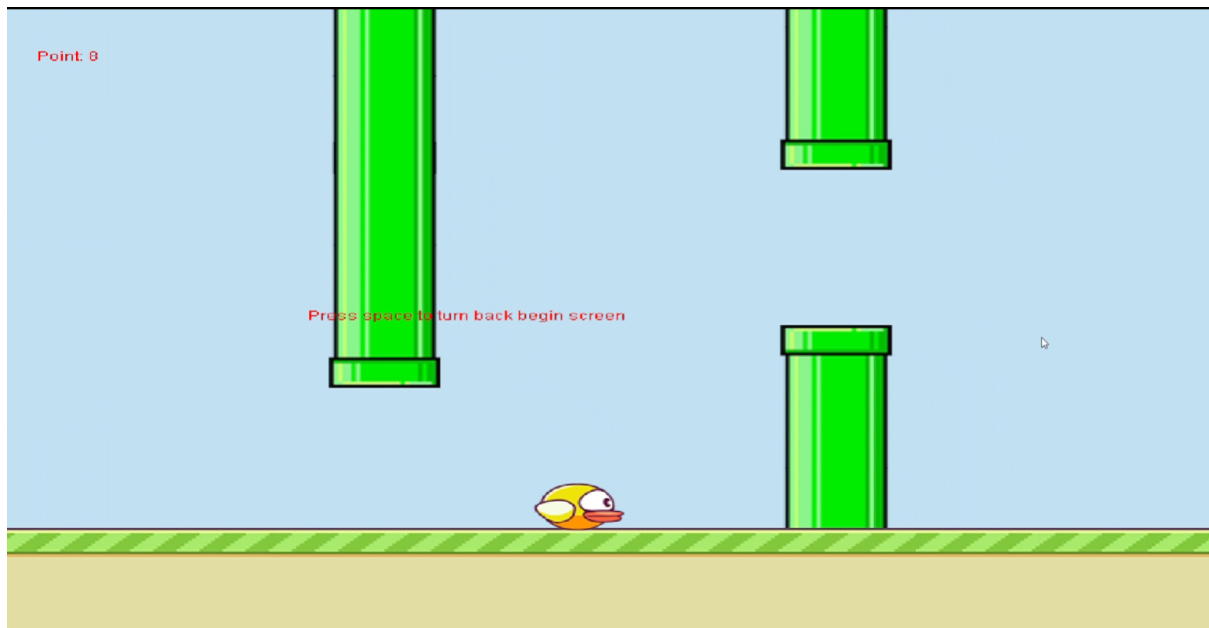
2. Giao diện chương trình



Màn hình bắt đầu



Màn hình khi đang chơi



Màn hình khi kết thúc

3. Kết luận

Chương trình chạy tốt, ổn định, mượt mà, không xảy ra lỗi

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận:

- Những điều đã làm được trong quá trình thực hiện bài tập lớn:
- + Thiết kế và xây dựng cấu trúc chương trình để tạo ra một trò chơi Flappy Birds hoạt động bằng ngôn ngữ Java.
- + Viết mã để điều khiển con chim trong game, cho phép người chơi nhấn phím Space để làm chim bay lên.
- + Xây dựng và hiển thị các vật cản, chẳng hạn như ống, di chuyển từ bên phải sang trái trên màn hình.
- + Triển khai mã xử lý va chạm giữa chim và các vật cản, để kiểm tra xem chim có va chạm với vật cản hay không và xử lý kết quả tương ứng.
- + Tính điểm dựa trên số lần chim vượt qua các vật cản và kiểm tra điều kiện kết thúc trò chơi khi chim va chạm với vật cản.
- + Hiển thị được các thành phần giao diện như điểm số.

- Ưu điểm:

- + Flappy Birds có cách chơi đơn giản, chỉ yêu cầu người chơi nhấn một nút để làm cho con chim nhảy qua các ống. Điều này làm cho trò chơi dễ tiếp cận và thu hút cho mọi đối tượng người chơi, bao gồm cả người mới bắt đầu.
- + Mặc dù Flappy Birds có cách chơi đơn giản, nhưng nó đòi hỏi sự tập trung và phản ứng nhanh.
- + Flappy Birds được biết đến với tính chất gây nghiện của nó. Sự kết hợp giữa tính thử thách và đơn giản, cùng với việc người chơi muốn cố gắng cải thiện điểm số của mình, tạo ra một trạng thái "chơi thêm một lần nữa" liên tục.
- + Mỗi lượt chơi trong Flappy Birds thường diễn ra trong thời gian ngắn, chỉ kéo dài trong vài giây đến vài phút. Điều này làm cho trò chơi lý tưởng để giải trí trong những khoảnh khắc rảnh rỗi hoặc khi bạn muốn một trò chơi nhanh chóng.
- + Với đồ họa đơn giản, Flappy Birds tạo ra một cái nhìn thân thiện và gần gũi. Điều này cũng giúp trò chơi chạy trơn tru trên nhiều thiết bị và thu hút sự quan tâm của người chơi.

- Nhược điểm:

- + Do cách chơi đơn giản và thiếu sự thay đổi trong nội dung, trò chơi Flappy Birds có thể trở nên đơn điệu và dễ chán sau một thời gian chơi dài. Việc lặp đi lặp lại cùng một hành động nhảy qua các ống có thể làm mất đi sự hứng thú và khám phá của người chơi.
- + Flappy Birds không có hệ thống cấp độ hoặc sự phát triển dài hạn, điều này làm giảm yếu tố tiếp thêm hứng thú của người chơi. Sau khi đạt được một điểm số cao, không có mục tiêu hay thách thức mới để tiếp tục.

+ Flappy Birds thiếu tính năng tương tác xã hội như chia sẻ điểm số hoặc thách đấu với bạn bè. Điều này làm giảm khả năng cạnh tranh và giao lưu giữa người chơi.

+ Trò chơi Flappy Birds được thiết kế với mục tiêu đơn giản và sự dễ tiếp cận, dẫn đến việc thiếu tính đa dạng và sự phức tạp trong gameplay. Điều này có thể khiến một số người chơi cảm thấy thiếu hứng thú và không có đủ thách thức để tiếp tục chơi.

2. Hướng phát triển:

- Thay đổi giao diện đồ họa và âm thanh của trò chơi để tạo ra một trải nghiệm mới mẻ. Hãy thử các phong cách nghệ thuật khác nhau, hiệu ứng hình ảnh động, hoặc nhạc nền độc đáo để tạo sự khác biệt.

- Một cách để tăng tính tương tác và cạnh tranh là thêm tính năng đa người chơi. Cho phép người chơi thi đấu trực tiếp với nhau qua mạng hoặc chia sẻ điểm số trên các nền tảng xã hội.

- Thêm các cấp độ khác nhau hoặc hệ thống sự phát triển để làm cho trò chơi trở nên phức tạp hơn và thú vị hơn theo thời gian. Các cấp độ mới có thể đi kèm với các hình thức vật cản khác nhau, tốc độ di chuyển khác nhau hoặc thậm chí đặt trong các môi trường khác nhau.

- Tạo thêm sự động lực cho người chơi bằng cách thêm hệ thống thành tích và thưởng. Người chơi có thể nhận được các thành tích đặc biệt khi đạt được các mục tiêu nhất định, và những thành tích này có thể mở khóa các tính năng hoặc nội dung bổ sung.

- Cho phép người chơi tùy chỉnh chim hoặc vật cản theo ý thích. Cung cấp các tùy chọn để người chơi có thể thay đổi hình dạng, màu sắc hoặc kiểu của chim và vật cản.

- Xem xét việc thêm các yếu tố mới vào trò chơi để làm cho nó thú vị hơn. Ví dụ, có thể thêm hiệu ứng đặc biệt, vật phẩm sức mạnh, hoặc các yếu tố bất ngờ để tạo ra sự đa dạng và hứng thú cho người chơi.

TÀI LIỆU THAM KHẢO

Sách Lập trình hướng đối tượng – Lê Thị Minh Châu

https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_overview.htm

<https://viettuts.vn/java-swing/lop-graphics-trong-java-swing>

<https://viblo.asia/p/bieu-do-lop-uml-Az45bDaVZxY>

<https://viblo.asia/p/phan-tich-thiet-ke-he-thong-thong-tin-su-dung-bieu-do-uml-phan-1-PjxMe6yNG4YL>

<https://vietjack.com/java/>