Shape, square

Description automatically generated

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**UNIVERSITY OF TRANSPOST AND COMMUNICATIONS**

Logo

Description automatically generated

**BÁO CÁO BÀI TẬP LỚN**

**MÔN CÔNG NGHỆ JAVA**

***ĐỀ TÀI: Thiết kế game brick breaker***

**Giảng viên hướng dẫn: Vũ Huấn**

**Sinh viên thực hiện:**

**Họ và tên: Ngô Minh Mạnh**

**Mã sinh viên: 211202236**

**Lớp: Công nghệ thông tin 6-k62**

**Năm 2023**

**Lời nói đầu**

Trong những năm gần đây, công nghệ thông tin đã phát triển mạnh mẽ trên mọi lĩnh vực trong cuộc sống và đặc biệt là trong lĩnh vực giải trí. Game là một trong những sản phẩm của công nghệ thông tin, giúp con người giải trí và phát triển nhiều kỹ năng quan trọng như tư duy, khả năng tập trung cao độ, tính kiên trì, quyết tâm.

Để lập trình một trò chơi như game cờ Caro, ngôn ngữ Java là một lựa chọn tốt. Với Java, việc thiết kế game trở nên dễ dàng hơn, mã chương trình rõ ràng, dễ hiểu và cô đọng hơn. Ngoài ra, sử dụng ngôn ngữ lập trình Java còn giúp tăng tính mềm dẻo cho chương trình và đồng thời cũng giúp tái sử dụng tài nguyên một cách hiệu quả.

Contents

[I. Giới thiệu 1](#_Toc134020040)

[1. Ngôn ngữ java 1](#_Toc134020041)

[a. Lịch sử 1](#_Toc134020042)

[b. Đặc điểm 1](#_Toc134020043)

[2. Mysql 1](#_Toc134020044)

[3. Game brick breaker 2](#_Toc134020045)

[a) Lịch sử 2](#_Toc134020046)

[b) Yêu cầu bài toán 2](#_Toc134020047)

[II. Thiết kế game 2](#_Toc134020048)

[1. Mô tả 2](#_Toc134020049)

[2. Các chức năng chính và chi tiết 2](#_Toc134020050)

[III. Các class 3](#_Toc134020051)

[1. Sơ đồ các class 3](#_Toc134020052)

[2. Các class 3](#_Toc134020053)

[a) Class Main: 3](#_Toc134020054)

[b) Class gamePanel: 3](#_Toc134020055)

[c) Class Brick: 4](#_Toc134020056)

[d) Class Ball: 4](#_Toc134020057)

[e) Class Paddle: 5](#_Toc134020058)

[f) Class connectJdbc 5](#_Toc134020059)

[IV. Giao diện game 6](#_Toc134020060)

[1. Bắt đầu và báo lỗi khi không nhập tên 6](#_Toc134020061)

[2. Game đang chơi 7](#_Toc134020062)

[3. Kết thúc Game 8](#_Toc134020063)

[V. Kết luận 8](#_Toc134020064)

[1. Những vấn đề đạt được 8](#_Toc134020065)

[2. Những vấn đề còn hạn chế 8](#_Toc134020066)

[VI. Tài liệu tham khảo 8](#_Toc134020067)

# Giới thiệu

## Ngôn ngữ java

1. **Lịch sử**

Java là một ngôn ngữ lập trình hiện đại và phổ biến được phát triển bởi Sun Microsystems (nay là Oracle Corporation). Java được phát hành vào những năm 1995 và đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới.

Java được sử dụng rộng rãi trong lĩnh vực phát triển ứng dụng di động, phát triển web, phát triển ứng dụng máy tính, trò chơi, và các ứng dụng doanh nghiệp. Java là một ngôn ngữ đa nền tảng, có nghĩa là chương trình viết bằng Java có thể chạy trên nhiều hệ điều hành khác nhau, bao gồm Windows, macOS, Linux và nhiều hệ điều hành khác.

1. **Đặc điểm**

Java có nhiều đặc điểm nổi bật, trong đó phải kể đến tính đa nền tảng, mã nguồn mở và an toàn. Tính đa nền tảng cho phép chương trình được phát triển trên một nền tảng và chạy trên nhiều nền tảng khác nhau mà không cần sửa đổi mã nguồn. Mã nguồn mở của Java cho phép người dùng chỉnh sửa và phát triển tiếp trên mã nguồn gốc mà không cần phải mua bản quyền. Về mặt an toàn, Java có tính năng bảo mật cao, cho phép ngăn chặn việc thực thi mã độc và các lỗ hổng bảo mật.

Java cũng có rất nhiều thư viện và framework hỗ trợ phát triển ứng dụng, giúp cho việc phát triển ứng dụng trở nên nhanh chóng và tiện lợi hơn. Các ứng dụng sử dụng Java được sử dụng rộng rãi trên các lĩnh vực khác nhau như phát triển web, game, ứng dụng di động, dịch vụ trực tuyến, và nhiều lĩnh vực khác nữa.

## Mysql

MySQL là một hệ quản trị cơ sở dữ liệu (DBMS) phổ biến, được sử dụng rộng rãi trên toàn thế giới. MySQL là một phần mềm mã nguồn mở, được phát triển bởi tập đoàn Oracle Corporation.

MySQL có tính năng vượt trội về tốc độ, hiệu suất, độ tin cậy và khả năng mở rộng. Nó hỗ trợ nhiều loại dữ liệu, bao gồm các kiểu dữ liệu số, chuỗi, ngày tháng và định dạng đặc biệt. MySQL có khả năng quản lý dữ liệu lớn, cho phép lưu trữ và truy xuất hàng triệu dòng dữ liệu một cách nhanh chóng và hiệu quả.

MySQL cũng được tích hợp với nhiều ngôn ngữ lập trình khác nhau, cho phép phát triển ứng dụng trên nhiều nền tảng khác nhau. Nó hỗ trợ các giao thức kết nối phổ biến như TCP/IP và các giao thức web như HTTP và FTP.

MySQL cũng được sử dụng rộng rãi trong các ứng dụng web, cho phép lưu trữ và quản lý dữ liệu trên các trang web và ứng dụng web. Nó cũng được sử dụng trong các ứng dụng di động và desktop, cho phép lưu trữ và quản lý dữ liệu trên các thiết bị khác nhau.

Với tính năng mạnh mẽ và sự phổ biến trên toàn thế giới, MySQL là một lựa chọn tốt cho các nhà phát triển và các doanh nghiệp để quản lý và lưu trữ dữ liệu của mình.

## Game brick breaker

* + 1. **Lịch sử**

Brick Breaker là một trò chơi điện tử cổ điển, được phát triển từ những năm 1970 và trở nên phổ biến trong những năm 1980 và 1990. Trò chơi này được đưa vào các máy trò chơi cầm tay và sau đó được phát triển trên nhiều nền tảng khác nhau như PC, điện thoại di động và máy tính bảng.

Trò chơi Brick Breaker đơn giản nhưng hấp dẫn, và đã trở thành một biểu tượng của thời đại điện tử cổ điển. Nhiều phiên bản của trò chơi đã được phát triển, với nhiều tính năng mới và đồ họa tốt hơn để đáp ứng với nhu cầu của người chơi hiện đại.

* + 1. **Yêu cầu bài toán**
* Yêu cầu chức năng
* Nhập thông tin người chơi
* Có chức năng tạm dừng game
* Hiển thị tên người chơi,điểm số,điểm cao nhất,…
* Kết nối cơ sở dữ liệu
* Yêu cầu sử dụng
* Chương trình có giao diện
* Giao diện đơn giản và dễ sử dụng
* Có khả năng nâng cấp

# Công nghệ được sử dụng

# Thiết kế game

## Mô tả

Mục tiêu của trò chơi Brick Breaker là phá hủy tất cả các viên gạch bằng cách điều khiển một thanh chắn và một quả bóng. Khi bóng va chạm với viên gạch, viên gạch sẽ bị phá hủy và điểm số của người chơi sẽ được tăng lên. Nếu bóng rơi xuống đất mà không được bắt bởi thanh chắn, người chơi sẽ mất một lượt chơi. Nếu người chơi phá hủy được tất cả các viên gạch, họ sẽ chiến thắng cấp độ và tiếp tục chơi ở cấp độ tiếp theo.

## Các chức năng chính và chi tiết

**Bắt đầu game:**Để bắt đầu game thì người chơi cần nhập tên.Nếu không nhập gì chương trình sẽ báo lỗi và yêu cầu nhập tên.

**Dừng trò chơi:**Người chơi có thể tạm dừng game bằng cách bấm space,hoặc tiếp tục chơi bằng cách bấm space.

**Hiển thị:**

+ Điểm số: Hiển thị số điểm mà người chơi đạt được.

+ Mạng: Hiển thị số mạng mà người chơi còn lại.

+ Level: Hiển thị level mà người chơi đang chơi.

+ Name: Hiển thị tên người chơi

+ Max Score: Hiển thị điểm người chơi cao nhất

+ Brick: Hiển thị số gạch còn lại cần phá hủy

**Di chuyển thanh paddle:** Người chơi sử dụng phím điều hướng để di chuyển

Dịch sang trái:

Dịch sang phải:

**Tính điểm:** Người chơi cần điều hướng bóng chạm gạch.Sau khi chạm 3 lần gạch sẽ biến mất,và sẽ được cộng 50 điểm

**Level:** Có 3 level,mỗi level thì tốc độ bóng sẽ bay nhanh hơn

**Game over:**

* Khi live bằng 0 game sẽ kết thúc.
* Khi level =3 và gạch biến mất hoàn toàn

## Các class

1. **Sơ đồ các class**
2. **Các class**

**Class Main:**

Mô tả: Để chạy ứng dụng. Nó sử dụng giao diện đồ họa Swing để tạo một cửa sổ JFrame. Cửa sổ này có kích thước là 700 x 500 pixel và không thể thay đổi kích thước. Tên của cửa sổ được đặt là "game Brick breaker". Khi người dùng đóng cửa sổ, ứng dụng sẽ kết thúc với lệnh "EXIT\_ON\_CLOSE".

Đối tượng "gamePanel" được tạo ra và thêm vào JFrame để hiển thị trò chơi. Phương thức "requestFocusInWindow()" được sử dụng để yêu cầu trường hợp được tập trung vào panel gamePanel để đảm bảo rằng các sự kiện phím được xử lý đúng.

**Class gamePanel:**

Mô tả: lớp gamePanel được thừa kế từ lớp JPanel và triển khai giao diện Runnable. Trong lớp này, có các biến và phương thức để quản lý các thông số trò chơi như điểm số, mức độ, số mạng còn lại, tốc độ bóng, độ rộng và chiều cao của gạch và nhiều hơn nữa.

Lớp gamePanel cũng bao gồm một số phương thức để xử lý các tình huống khác nhau trong trò chơi.

Một số phương thức

* setGiaTri() để thiết lập các giá trị ban đầu cho thanh chắn và quả bóng,
* makeBricks() để tạo gạch,
* startGame() để bắt đầu trò chơi
* pauseGame() để tạm dừng trò chơi.
* playMusic() để chơi nhạc nền
* run() được triển khai để xử lý các tình huống di chuyển của bóng và xử lý va chạm với các phần tử khác trong trò chơi, chẳng hạn như thanh chắn, gạch và tường. Ngoài ra, nó cũng kiểm tra số mạng và mức độ của người chơi và đặt lại bóng, gạch và thanh chắn tùy thuộc vào kết quả của trò chơi.

**Class Brick:**

Mô tả: Chứa các thuộc tính và phương thức để quản lý và vẽ các viên gạch trong trò chơi

Các thuộc tính của lớp bao gồm:

* brickX: tọa độ X của viên gạch trên màn hình
* brickY: tọa độ Y của viên gạch trên màn hình
* brick\_width: chiều rộng của viên gạch
* brick\_height: chiều cao của viên gạch
* destroyed: biến boolean để xác định liệu viên gạch đã bị phá hủy hay chưa
* color: màu sắc của viên gạch
* hit: số lần va chạm của quả bóng với viên gạch

Các phương thức của lớp bao gồm:

* draw(Graphics g): phương thức để vẽ viên gạch trên màn hình
* addhit(): phương thức để tăng biến hit lên 1 và thay đổi màu sắc của viên gạch theo số lần va chạm của quả bóng
* hitBottom(int x, int y): phương thức để xác định xem quả bóng có va chạm vào phía dưới của viên gạch không
* hitTop(int x, int y): phương thức để xác định xem quả bóng có va chạm vào phía trên của viên gạch không
* hitLeft(int x, int y): phương thức để xác định xem quả bóng có va chạm vào phía trái của viên gạch không
* hitRight(int x, int y): phương thức để xác định xem quả bóng có va chạm vào phía phải của viên gạch không
* Ngoài ra còn có 1 số phương thức getter và setter

**Class Ball:**

**Mô tả:**Lớp này chứa các thuộc tính và phương thức của bóng

Các thuộc tính của lớp gồm:

* BallX: tọa độ x của quả bóng trên màn hình.
* BallY: tọa độ y của quả bóng trên màn hình.
* BALL\_WIDTH: chiều rộng của quả bóng.
* BALL\_HEIGHT: chiều cao của quả bóng.
* xDir: hướng di chuyển của quả bóng theo trục x.
* yDir: hướng di chuyển của quả bóng theo trục y.
* color: màu sắc của quả bóng.
* onScreen: biến boolean xác định xem quả bóng có hiển thị trên màn hình hay không.

Các phương thức của lớp gồm:

* Phương thức khởi tạo "Ball": khởi tạo các giá trị cho các thuộc tính của lớp.
* Phương thức "draw": vẽ quả bóng trên màn hình với màu sắc được định nghĩa.
* Phương thức "move": di chuyển quả bóng theo hướng xDir và yDir.
* Phương thức "reset": đặt lại vị trí của quả bóng và hướng di chuyển về giá trị mặc định.
* Các phương thức getter và setter cho các thuộc tính của lớp.

**Class Paddle:**

**Mô tả:** Lớp này chứa các thuộc tính và phương thức của paddle

Các thuộc tính của lớp Paddle bao gồm:

* padX: tọa độ X của thanh trên màn hình.
* PadY: tọa độ Y của thanh trên màn hình.
* pad\_width: độ rộng của thanh .
* pad\_height: chiều cao của thanh .
* color: màu sắc của thanh trượt.

Các phương thức của lớp Paddle bao gồm:

* Paddle(): hàm khởi tạo.
* draw(Graphics g): phương thức vẽ thanh .
* reset(): phương thức thiết lập lại vị trí của thanh .
* vaCham(int x, int y): phương thức kiểm tra va chạm giữa thanh và quả bóng.
* Các phương thức getter và setter cho các thuộc tính của lớp.

**Class connectJdbc**

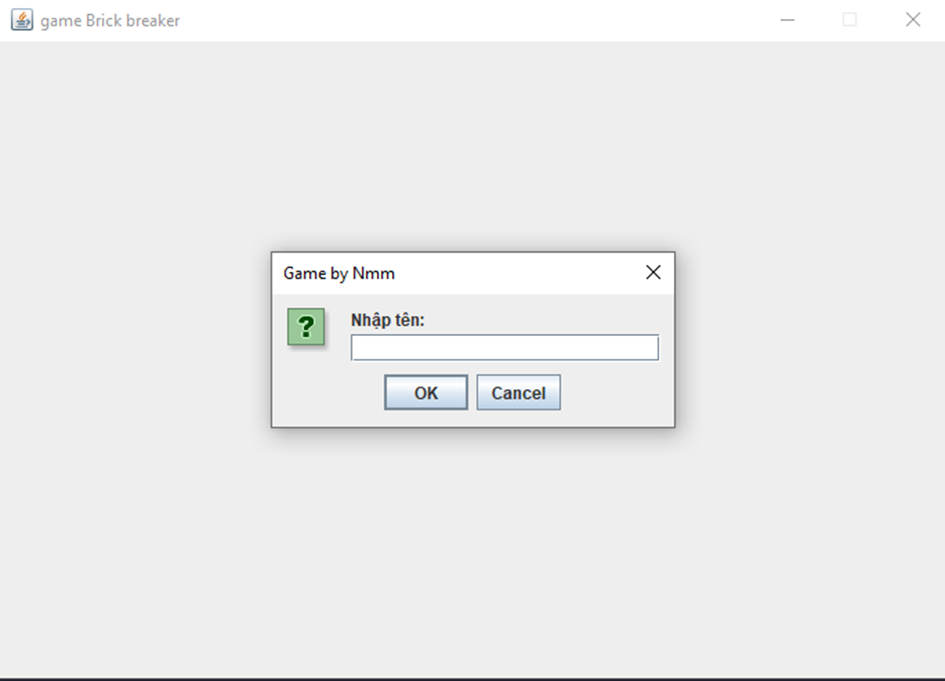
**Mô tả:** Đây là lớp kết nối cơ sở dữ liệu. Để kết nối và thao tác với cơ sở dữ liệu MySQL. Trong constructor của class, chúng ta tạo một kết nối tới cơ sở dữ liệu MySQL sử dụng địa chỉ URL, tên người dùng và mật khẩu. Sau đó, chúng ta tạo một đối tượng Statement để thực thi các lệnh truy vấn SQL.

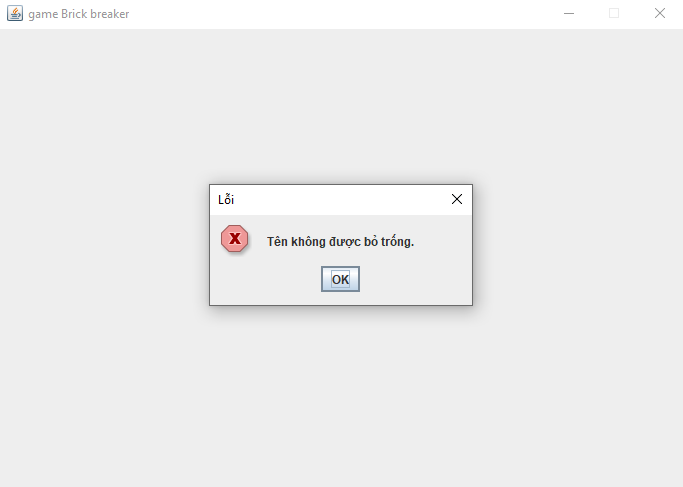
Class này có 3 phương thức:

* addScores: được sử dụng để thêm điểm số vào bảng scores trong cơ sở dữ liệu. Nó lấy tên người chơi, điểm số và mức độ chơi làm tham số và sử dụng đối tượng Statement để thực thi lệnh SQL INSERT vào bảng scores.
* getHighestScore được sử dụng để lấy điểm số cao nhất từ bảng scores. Nó thực hiện một truy vấn SQL SELECT để lấy điểm số cao nhất và trả về giá trị đó. Nếu không tìm thấy kết quả, nó trả về giá trị 0.
* close được sử dụng để đóng kết nối với cơ sở dữ liệu khi không cần sử dụng nữa.

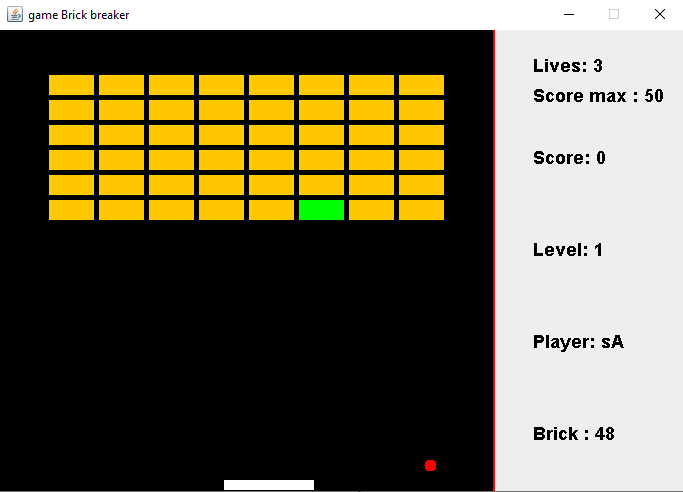
# Giao diện game

## Bắt đầu và báo lỗi khi không nhập tên

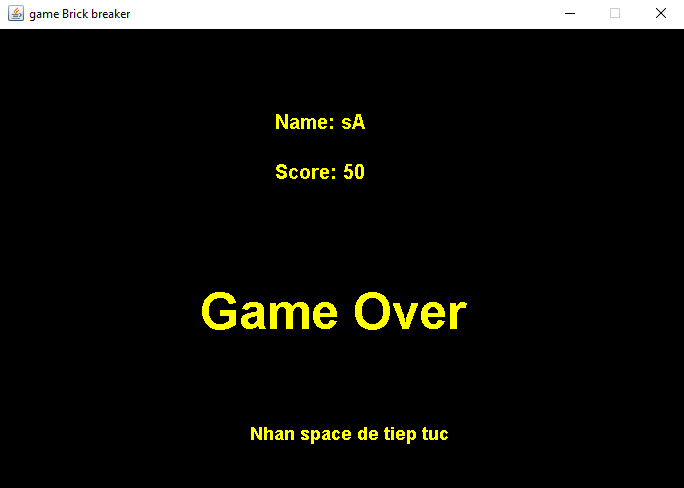




## Game đang chơi



## Kết thúc Game



# Kết luận

## Những vấn đề đạt được

Với kiến thức nền tảng đã được học ở trường và bằng sự nỗ lực của mình,em đã hoàn thành đề tài “Lập trình game brick breaker bằng phương pháp hướng đối tượng”. Tài liệu này giúp ta có cái nhìn toàn vẹn về cơ cấu của phần mềm, cũng như chức năng của phần mềm

## Những vấn đề còn hạn chế

Do thời gian có hạn, nên đề tài không thể hoàn thành hết các chức năng như dự định ban đầu đã đặt ra.Tuy nhiên, những chức năng chưa hoàn thiện sẽ sớm được bổ sung và hoàn chỉnh trong thời gian sớm nhất.Trong quá trình thiết kế và thực hiện không tránh khỏi sai sót, mong thầy và các bạn thông cảm và có những góp ý quí báu nhằm hoàn thiện hơn cho sản phẩm.

# Tài liệu tham khảo

[Java (ngôn ngữ lập trình) – Wikipedia tiếng Việt](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh))