TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
UNIVERSITY OF TRANSPOST AND COMMUNICATIONS

BÁO CÁO BÀI TẬP MÔN CÔNG NGHỆ JAVA

ĐỀ TÀI: LÀM GAME BOMBERMAN

Giảng viên hướng dẫn: Vũ Huấn
Họ và tên: Dương Ngọc Huy
Mã SV : 211212006
Lớp : CNTT6 – K62

Năm 2023

LỜI MỞ ĐẦU

Trong thời đại công nghệ 4.0 hiện nay, game đang trở thành một lĩnh vực nhanh chóng phát triển và thu hút sự quan quan tâm cua của nhiều người. Game không chỉ mang tính chất giải trí mà còn mang lại khả năng giải quyết vấn đề cho chúng ta.

Trong đề tài này, tôi đã tập trung vào phát triển một phần mềm game bằng ngôn ngữ lập trình Java. Java là một ngôn ngữ lập trình đa nền tảng, cho phép chúng ta phát triển các ứng dụng trên nhiều hệ điều hành và các thiết bị khác nhau.

Hy vọng bài cáo sau sẽ giúp các bạn có một cái nhìn tổng quát về quá trình phát triển một game bằng ngôn ngữ java.

1.Mô tả yêu cầu bài toán

- Nhân vật được điều khiển trên môi trường chơi game, tránh né các đối thủ, nếu va vào đối thủ sẽ kết thúc và quay trở lại ban đầu.
- Nhân vật có thể đặt bom để phá tường và tiêu diệt đối thủ.
- Thể hiện thông báo khi người chơi kết thúc game.
- Cho phép người chơi bắt đầu lại game và kết thúc game.

2.Phân tích thiết kế chi tiết

A.

Lớp AbstractCharacter

Lớp có bốn thuộc tính SIZE, x, y, và pixelsPerStep.

- Trường SIZE là một số nguyên hằng số có giá trị là 30.
- Trường x và y là các số nguyên đại diện cho vị trí của ký tự trên màn hình.
- Trường pixelsPerStep là một số nguyên đại diện cho số lượng pixel ký tự di chuyển mỗi bước.

Lớp này có hai hàm tạo: một hàm tạo mặc định và một hàm tạo có ba đối số: x, y, và pixelsPerStep.

Lớp cũng có bốn phương thức: move(), moveBack(), getSize(), và getX().

- Phương thức move() nhận đối số kiểu Move là một enum đại diện cho hướng ký tự sẽ di chuyển.
- Phương thức cập nhật vị trí của ký tự dựa trên hướng được truyền vào.
- Phương thức moveBack() nhận đối số kiểu Move là một enum đại diện cho hướng hiện tại của ký tự.
- Phương thức di chuyển ký tự lại một bước theo hướng ngược lại.
- Phương thức getSize() trả về giá trị của hằng số nguyên SIZE.
- Phương thức getX() trả về giá trị của trường riêng tư nguyên x.

B.

Lớp AbstractPowerup

- Thuộc tính:
  - POWERUP_SIZE: một hằng số nguyên tĩnh có giá trị là 30.
  - x: một số nguyên đại diện cho vị trí của powerup trên trục x.
  - y: một số nguyên đại diện cho vị trí của powerup trên trục y.
  - name: một chuỗi đại diện cho tên của powerup.
- Phương thức:
  - AbstractPowerup(int x, int y): hàm tạo của lớp với hai đối số là x và y.
  - addToPlayer(Player player): phương thức không trả về giá trị, được sử dụng để thêm powerup vào người chơi.
  - getPowerupSize(): phương thức trả về giá trị của hằng số POWERUP_SIZE.
  - getX(): phương thức trả về giá trị của thuộc tính x.
  - getY(): phương thức trả về giá trị của thuộc tính y.
  - getName(): phương thức trả về giá trị của thuộc tính name.

C.

Lớp Bomb

- Thuộc tính:
  - BOMBSIZE: một hằng số nguyên tĩnh có giá trị là 30.
  - STARTCOUNTDOWN: một hằng số nguyên tĩnh có giá trị là 100.
  - timeToExplosion: một số nguyên đại diện cho thời gian còn lại cho đến khi bom nổ.

    - rowIndex: một số nguyên đại diện cho chỉ số hàng của bom.
    - colIndex: một số nguyên đại diện cho chỉ số cột của bom.
    - explosionRadius: một số nguyên đại diện cho bán kính của vùng bị nổ của bom.
    - playerLeft: một giá trị boolean đại diện cho việc người chơi đã rời khỏi vùng bị nổ hay chưa.
- Phương thức:
    - Bomb(final int rowIndex, final int colIndex, int explosionRadius): hàm tạo của lớp với ba đối số là rowIndex, colIndex và explosionRadius.
    - getRowIndex(): phương thức trả về giá trị của thuộc tính rowIndex.
    - getColIndex(): phương thức trả về giá trị của thuộc tính colIndex.
    - getBOMBSIZE(): phương thức tĩnh trả về giá trị của hằng số BOMBSIZE.
    - getTimeToExplosion(): phương thức trả về giá trị của thuộc tính timeToExplosion.
    - setTimeToExplosion(final int timeToExplosion): phương thức không trả về giá trị, được sử dụng để thiết lập giá trị cho timeToExplosion.
    - getExplosionRadius(): phương thức trả về giá trị của thuộc tính explosionRadius.
    - isPlayerLeft(): phương thức trả về giá trị của thuộc tính playerLeft.
    - setPlayerLeft(final boolean playerLeft): phương thức không trả về giá trị, được sử dụng để thiết lập giá trị cho playerLeft.

D*. Lớp BombCounterPU* có các thuộc tính và phương thức sau:

- Thuộc tính:
    - Không có thuộc tính nào được khai báo.
- Phương thức:
    - BombCounterPU(int rowIndex, int colIndex): hàm tạo của lớp với hai đối số là rowIndex và colIndex.
    - addToPlayer(Player player): phương thức không trả về giá trị, được sử dụng để điều chỉnh bombCount của người chơi.
    - getName(): phương thức trả về tên của lớp.

E.

Lớp BombermanComponent có các thuộc tính và phương thức sau:

- Thuộc tính:
    - SQUARE_SIZE: hằng số kiểu int.
    - CHARACTER_ADJUSTMENT_FOR_PAINT: hằng số kiểu int.
    - SQUARE_MIDDLE: hằng số kiểu int.
    - BOMB_ADJUSTMENT_1: hằng số kiểu int.
    - BOMB_ADJUSTMENT_2: hằng số kiểu int.
    - PAINT_PARAMETER_13: hằng số kiểu int.
    - PAINT_PARAMETER_15: hằng số kiểu int.
    - PAINT_PARAMETER_17: hằng số kiểu int.
    - PAINT_PARAMETER_18: hằng số kiểu int.
    - PAINT_PARAMETER_19: hằng số kiểu int.
    - PAINT_PARAMETER_20: hằng số kiểu int.
    - PAINT_PARAMETER_24: hằng số kiểu int.
- Phương thức:
    - BombermanComponent(Floor floor): hàm tạo của lớp với đối số là floor.
    - getSquareSize(): phương thức tĩnh trả về kích thước của mỗi ô vuông.

- getSquareMiddle(): phương thức tĩnh trả về giá trị giữa của mỗi ô vuông.
- getPreferredSize(): phương thức trả về kích thước được đề xuất của component.
- floorChanged(): phương thức không trả về giá trị, được sử dụng để vẽ lại component.
- Phương thức paintComponent() được sử dụng để vẽ đồ họa trên một thành phần. Nó được gọi tự động bởi phương thức paint() (trong lớp java.awt.Component) và được ghi đè từ lớp JPanel. Bất kỳ mã nào bạn sử dụng để vẽ một cái gì đó nên được đặt trong phương thức này.Phương thức paintComponent() được sử dụng để vẽ bảng trò chơi Bomberman. Phương thức này sử dụng một đối tượng Graphics để vẽ bảng trò chơi và các thành phần của nó như người chơi, kẻ thù, bom, powerup và explosions. Phương thức cũng sử dụng một bản đồ màu để thiết lập màu của mỗi ô trên bảng trò chơi.
- Phương thức paintPlayer(Player player, Graphics g2d) được sử dụng để vẽ người chơi trên bảng trò chơi Bomberman. Nó nhận một đối tượng Player và một đối tượng Graphics làm đối số. Đối tượng Player chứa thông tin về vị trí của người chơi trên bảng trò chơi và đối tượng Graphics được sử dụng để vẽ người chơi.Trong phương thức này, bạn sẽ sử dụng đối tượng Graphics để vẽ người chơi. Bạn có thể sử dụng các phương thức của đối tượng Graphics như drawImage() hoặc fillRect() để vẽ người chơi.
- Phương thức vẽ powerup. Vòng lặp for được sử dụng để lặp qua danh sách các đối tượng AbstractPowerup trong danh sách powerupList của đối tượng floor. Đối tượng AbstractPowerup chứa thông tin về vị trí và loại của các powerup trên bảng trò chơi.Trong phương thức này, bạn sẽ sử dụng đối tượng Graphics để vẽ các powerup. Nếu tên của powerup là “BombCounter”, màu đen sẽ được sử dụng để vẽ powerup. Nếu tên của powerup là “BombRadius”, màu đỏ sẽ được sử dụng để vẽ powerup. Bạn có thể sử dụng phương thức fillOval() của đối tượng Graphics để vẽ các powerup.
- Phương thức vẽ các quả bom trên bảng trò chơi Bomberman. Vòng lặp for được sử dụng để lặp qua danh sách các đối tượng Bomb trong danh sách bombList của đối tượng floor. Đối tượng Bomb chứa thông tin về vị trí và thời gian nổ của các quả bom trên bảng trò chơi.Trong phương thức này, bạn sẽ sử dụng đối tượng Graphics để vẽ các quả bom. Màu đỏ sẽ được sử dụng để vẽ quả bom. Bạn có thể sử dụng phương thức fillOval() của đối tượng Graphics để vẽ các quả bom.
- PPrivate void paintFloor(): phướng thức để vẽ sàn
- PPrivate void paintEnemy(): phương thức để vẽ Enemy

G.

Lớp ***BombermanFrame*** là một lớp kế thừa từ lớp JFrame.

Nó có hai thuộc tính là floor và bombermanComponent.

- Thuộc tính floor là một đối tượng của lớp Floor, chứa thông tin về bản đồ trò chơi Bomberman.
- Thuộc tính bombermanComponent là một đối tượng của lớp BombermanComponent, chứa thông tin về người chơi và các đối tượng trên bản đồ trò chơi.

- Phương thức khởi tạo của lớp này được sử dụng để thiết lập các thuộc tính và các thành phần giao diện người dùng. Nó cũng sử dụng phương thức setKeyStrokes() để thiết lập các phím tắt cho trò chơi.
- Lớp này cũng có một số phương thức khác như askUser() để hỏi người dùng một câu hỏi và trả về câu trả lời của họ, và quit() để thoát khỏi trò chơi.

H. Lớp ***BombRadiusPU*** là một lớp kế thừa từ lớp AbstractPowerup.
Nó có hai thuộc tính là rowIndex và colIndex, được kế thừa từ lớp cha.
Lớp này cũng có các phương thức như addToPlayer() để điều chỉnh bán kính nổ của người chơi và getName() để trả về tên của đối tượng.

I.
Lớp Enemy là một lớp kế thừa từ lớp AbstractCharacter.
Nó có hai thuộc tính là x và y, được kế thừa từ lớp cha. Lớp này cũng có các phương thức như changeDirection() để thay đổi hướng di chuyển của đối tượng và getCurrentDirection() để trả về hướng di chuyển hiện tại của đối tượng.

Phương thức khởi tạo của lớp này được sử dụng để thiết lập các thuộc tính.

K.

Lớp Engine là một lớp cuối cùng và không thể kế thừa.

Nó có các thuộc tính như TIME_STEP, width, height, nrOfEnemies, clockTimer. Lớp này cũng có các phương thức như main(), startGame(), gameOver(), và tick().Phương thức khởi tạo của lớp này được sử dụng để thiết lập các thuộc tính.

I.

Lớp Explosion là một lớp đơn giản với các thuộc tính như rowIndex, colIndex, và duration. Lớp này cũng có các phương thức như getRowIndex(), getColIndex(), getDuration(), và setDuration().

Phương thức khởi tạo của lớp này được sử dụng để thiết lập các thuộc tính.

M.
Lớp FloorListener là một lớp giao diện với một phương thức là floorChanged(). Phương thức này được triển khai trong lớp BombermanComponent để vẽ lại trò chơi.

Lớp này không có thuộc tính.

N. Enum được gọi là FloorTile. Một enum là một loại kiểu dữ liệu đặc biệt cho phép một biến chỉ có một trong số các giá trị được xác định trước. Trong trường hợp này, enum FloorTile có ba giá trị: FLOOR, UNBREAKABLEBLOCK và BREAKABLEBLOCK. Mỗi giá trị đại diện cho một loại thành phần khác nhau mà sàn có thể có trong một trò chơi. Mục đích của mã này là xác định các giá trị này để chúng có thể được sử dụng trong logic của trò chơi.

O.
Lớp Player. Lớp này kế thừa từ lớp AbstractCharacter và có các thuộc tính explosionRadius, bombCount và floor. Nó cũng có các phương thức up, right, down, left và dropBomb.

> Thuộc tính explosionRadius là bán kính của vùng nổ của bom mà người chơi sẽ thả.
>
> Thuộc tính bombCount là số lượng bom tối đa mà người chơi có thể giữ.

Thuộc tính floor là một đối tượng kiểu Floor, đại diện cho sàn của trò chơi.

Phương thức up, right, down và left xử lý việc di chuyển người chơi theo hướng tương ứng.

Phương thức dropBomb() kiểm tra xem ô hiện tại có bom hay không và số lượng bom trong danh sách các bom nhỏ có nhỏ hơn số lượng bom tối đa mà người chơi có thể giữ hay không. Nếu điều kiện này được thoả mãn, phương thức sẽ tạo ra một đối tượng kiểu Bomb mới và thêm vào danh sách các bom. Sau khi danh sách các bom đã được cập nhật, phương thức notifyListeners() được gọi để thông báo cho các lớp khác biết rằng danh sách các bom đã được cập nhật.
Phương thức setPlayerButtons(BombermanComponent bombermanComponent) thiết lập các phím mà người chơi có thể sử dụng để di chuyển và thả bom.

Phương thức getBombCount() trả về số lượng bom tối đa mà người chơi có thể giữ.

Phương thức setBombCount(int bombCount) thiết lập số lượng bom tối đa mà người chơi có thể giữ.

Phương thức getExplosionRadius() trả về bán kính của vùng nổ của bom mà người chơi sẽ thả.

Phương thức setExplosionRadius(int explosionRadius) thiết lập bán kính của vùng nổ của bom mà người chơi sẽ thả.

Phương thức movePlayer(Move move) xử lý việc di chuyển người chơi theo hướng tương ứng. Nếu người chơi va chạm với khối hoặc bom, phương thức sẽ di chuyển người chơi trở lại vị trí ban đầu. Nếu người chơi va chạm với kẻ địch, phương thức sẽ đặt cờ isGameOver thành true để kết thúc trò chơi. Sau khi di chuyển, phương thức kiểm tra xem người chơi đã rời khỏi bom hay không và kiểm tra va chạm với các powerup. Cuối cùng, phương thức notifyListeners() được gọi để thông báo cho các lớp khác biết rằng sàn đã được cập nhật.

Y.

Lớp Floor có các thuộc tính sau:

- private final static double CHANCE_FOR_BREAKABLE_BLOCK: xác suất để một khối bị phá hủy.
- private final static double CHANCE_FOR_RADIUS_POWERUP: xác suất để một powerup bán kính xuất hiện.
- private final static double CHANCE_FOR_COUNTER_POWERUP: xác suất để một powerup đếm ngược xuất hiện.
- private final FloorTile[][] tiles: mảng 2 chiều chứa các ô của sàn.
- private int width: chiều rộng của sàn.
- private int height: chiều cao của sàn.
- private Collection<FloorListener> floorListeners: danh sách các listener được đăng ký để theo dõi sàn.
- private Player player: người chơi hiện tại.
- private Collection<Enemy> enemyList: danh sách các kẻ địch trên sàn.
- private List<Bomb> bombList: danh sách các bom trên sàn.

- private Collection<AbstractPowerup> powerupList: danh sách các powerup trên sàn.
- private Collection<Bomb> explosionList: danh sách các bom đã nổ trên sàn.
- private Collection<Explosion> explosionCoords: danh sách tọa độ của vùng nổ của bom trên sàn.
- private boolean isGameOver: biến boolean cho biết trò chơi đã kết thúc hay chưa.

Lớp Floor có các phương thức sau:

- public Floor(int width, int height, int nrOfEnemies): khởi tạo một sàn mới với chiều rộng, chiều cao và số lượng kẻ địch được chỉ định.
- public static int pixelToSquare(int pixelCoord): chuyển đổi tọa độ pixel thành tọa độ ô trên sàn.
- public FloorTile getFloorTile(int rowIndex, int colIndex): trả về ô tại hàng rowIndex và cột colIndex trên sàn.
- public int getWidth(): trả về chiều rộng của sàn.
- public int getHeight(): trả về chiều cao của sàn.
- public Player getPlayer(): trả về người chơi hiện tại.
- public Collection<Enemy> getEnemyList(): trả về danh sách các kẻ địch trên sàn.
- public Iterable<Bomb> getBombList(): trả về danh sách các bom trên sàn dưới dạng Iterable.
- public int getBombListSize(): trả về số lượng bom trên sàn.
- public Iterable<AbstractPowerup> getPowerupList(): trả về danh sách các powerup trên sàn dưới dạng Iterable.
- public Iterable<Explosion> getExplosionCoords(): trả về danh sách tọa độ của vùng nổ của bom trên sàn dưới dạng Iterable.
- public boolean getIsGameOver(): trả về giá trị của biến isGameOver.