**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

###### KHOA CÔNG NGHỆ THÔNG TIN

**🙠  🙢**

****

**BÁO CÁO BÀI TẬP LỚN JAVA**

***Đề tài: Lập trình game nông trại 2D***

***Nhóm thực hiện: Nhóm 36***

***Thành viên:***

***Nguyễn Đăng Quang – 211204147***

***An Đức Thanh – 211214128***

***Lớp : Công nghệ thông tin 6 – Khóa 62***

***Giảng viên hướng dẫn : Vũ Huấn***

**HÀ NỘI – 2023**

**🙠  🙢**

**LỜI MỞ ĐẦU**

Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động. Nó được coi là công nghệ mang tính cách mạng và khả thi nhất trong việc tạo ra các ứng dụng có khả năng chạy thống nhất trên nhiều nền tảng mà chỉ cần biên dịch một lần. Ngày nay, với sự phát triển nhanh của kinh tế, xã hội, công nghệ, đồng nghĩa với việc con người có nhiều khó khăn, áp lực hơn trong đời sống và công việc. Vì vậy, nhu cầu giải trí là một trong những nhu cầu cấp thiết và chính đáng của mọi người, đây là cơ sở để con người cân bằng cuộc sống, tái tạo sức lao động. Bài tập lớn của nhóm chúng em lựa chọn nhằm mục đích để đáp ứng nhu cầu trên, áp dụng ngôn ngữ Java, Cơ sở dữ liệu để xây dựng và thiết kế một chương trình trò chơi đơn giản. Từ đó, giúp chúng em hiểu rõ hơn về ngôn ngữ này, bổ sung thêm kiến thức về hướng đối tượng, cơ sở dữ liệu, đa luồng… để sau này có thể áp dụng tốt vào thực tế.

**Mục lục:**

[**I.** **Giới thiệu các gói của bài tập lớn.** 4](#_Toc134028083)

[**1.** **Gói database.** 4](#_Toc134028084)

[**2.** **Gói Key\_Handler.** 4](#_Toc134028085)

[**3.** **Gói Main.** 4](#_Toc134028086)

[**4.** **Gói map.** 4](#_Toc134028087)

[**5.** **Gói Menu.** 4](#_Toc134028088)

[**6.** **Gói object.** 4](#_Toc134028089)

[**7.** **Panel.** 4](#_Toc134028090)

[**8.** **Image:** 4](#_Toc134028091)

[**II.** **Giới thiệu các lớp của bài tập lớn.** 4](#_Toc134028092)

[**1.** **Gói Panel.** 4](#_Toc134028093)

[**2.** **Gói Menu.** 6](#_Toc134028094)

[**3.** **Gói Main.** 8](#_Toc134028095)

[**4.** **Gói object.** 8](#_Toc134028096)

[**5.** **Gói map.** 14](#_Toc134028097)

[**6.** **Gói Key\_Handler.** 14](#_Toc134028098)

[**7.** **Gói database.** 15](#_Toc134028099)

[**III.Giới thiệu và mô tả chức năng của trò chơi.** 15](#_Toc134028100)

[**1.** **Giới thiệu:** 15](#_Toc134028101)

[**2.** **Chức năng của trò chơi:** 17](#_Toc134028102)

[**IV.Hướng dẫn chạy mã lệnh trên máy tính.** 22](#_Toc134028103)

[**V.** **Phần kết.** 23](#_Toc134028104)

1. **Giới thiệu các gói của bài tập lớn.**
   * + 1. **Gói database.**

* **Chức năng:** Được sử dụng để chứa các lớp liên quan tới kết nối cơ sở dữ liệu.
  + - 1. **Gói Key\_Handler.**
* **Chức năng:** Được sử dụng để chứa các lớp nhận sự kiện nhập phím của người dùng.
  + - 1. **Gói Main.**
* **Chức năng:** Sử dụng để chứa hàm main và một lớp View kế thừa lớp JFrame để chứa các thành phần hiển thị khác của trò chơi.
  + - 1. **Gói map.**
* **Chức năng:** Sử dụng để chứa các lớp liên quan đến tạo bản đồ.
  + - 1. **Gói Menu.**
* **Chức năng:** Sử dụng để chứa lớp Menu và các lớp con của nó giúp hiển thị các bảng chọn cho người dùng.
  + - 1. **Gói object.**
* **Chức năng:** Sử dụng để chứa các lớp đối và thực thể trong trò chơi
  + - 1. **Panel.**
* **Chức năng:** Ngoài lớp Panel thì gói này còn chứa các lớp liên quan đến âm thanh, khởi tạo đối tượng, kiểm tra va chạm và hiển thị giao diện người dùng.
  + - 1. **Image:**
* **Chức năng:** Lưu bản đồ ma trận và các gói chứa hình ảnh của từng đối tượng,

1. **Giới thiệu các lớp của bài tập lớn.**
   1. **Gói Panel.**
      * 1. **Lớp GamePanel.**

* **Chức năng:** Kế thừa lớp Jpanel và implement lớp Runnabel để tạo một panel và một luồng để hiển thị hình ảnh.
* **Thuộc tính:**
* **Public Int:** tileSize, maxSizeCol, maxSizeRow, screenWidth, screenHeight, maxWorldCol, maxWorldRow, pigQuantity, cowQuantity, chickenQuantity, FPS.
* **Public Sound:** sound.
* **Public UI:** ui.
* **Public Thread:** thread.
* **Public keyHandler:** keyHandler.
* **Public collisionChecker:** collisionChecker.
* **Public AssetSetter:** assetSetter.
* **Public Player:** player.
* **Public tileManager:** tileManager.
* **Public Mảng object:** object[].
* **Public Mảng entity:** entity[].
* **Public HouseMenu:** houseMenu.
* **Public AnimalMenu:** AnimalMenu.
* **Public TrayMenu:** trayMenu.
* **Public LandTagMenu:** landTagMenu.
* **Public ShopMenu:** shopMenu.
* **Public JDBCUtil:** jdbcUtil.
* **Phương thức:**
* **Public GamePanel():** Hàm không đối dùng để thiết lập panel.
* **Public void setupGame():** Hàm void dùng để khởi tạo các đối tượng.
* **Public void startThread():** Hàm void sử dụng để khởi tạo và bắt đầu luồng.
* **Public void run():** Hàm ghi đè của lớp thread để thiết lập luồng.
* **Public void update():** Hàm void sử dụng để cập nhật trạng thái, giá trị của các đối tượng.
* **Public void paint():** Hàm void dùng để vẽ hình ảnh lên panel.
  1. **Lớp collisionChecker:**
* **Chức năng:** Có chức năng kiểm tra va chạm của người chơi.
* **Thuộc tính:**
* **Public Gamepanel:** gp.
* **Public Player:** player.
* **Phương thức:**
* **Public void checkTile(entity player):** Kiểm tra va chạm của người chơi đối với ô gạch.
* **Public int checkObject(entity player):** Kiểm tra va chạm của người chơi với đối tượng.
* **Public void checkEntity(Player player):** Kiểm tra va chạm của người chơi với động vật.
  1. **Lớp AssetSetter:**
* **Chức năng:** Dùng để khởi tạo các đối tượng trong mảng object và mảng entity.
* **Thuộc tính:**
* **Public GamePanel:** gp.
* **Phương thức:**
* **Public void setObject():** Khởi tạo đối tượng.
* **Public void setEntity():** Khởi tạo động vật.
  1. **Lớp Sound:**
* **Chức năng:** implement lớp Runnable để tạo âm thanh cho trò chơi, và đây cũng là một luồng.
* **Thuộc tính:**
* **Public Clip:** clip.
* **PublicMảng URL:** soundURL[].
* **Public Thread:** soundThread.
* **Phương thức:**
* **Public Sound():** Hàm tạo không đối để nhận đường dẫn của file âm thanh.
* **Public void setFile():** Hàm void để nhận và mở file âm thanh.
* **Public void play():** Hàm void để bắt đầu chạy âm thanh.
* **Public void loop():** Hàm void để thực hiện vòng lặp âm thanh.
* **Public void soundstop():** Hàm void để dừng âm thanh.
* **Public void soundTrack():** Hàm void để chạy nhạc nền.
* **Public void playMusic():** Hàm void để nhận file âm thanh, bắt đầu và thực hiện vòng lặp của nhạc nền.
* **Public void playSE():** Hàm void để chạy âm thanh.
* **Public void run():** Hàm ghi đè của lớp Thread để thiết lập luồng.
  1. **Lớp UI.**
* **Chức năng:** Đây là lớp để hiển thị giao diện người dùng.
* **Thuộc tính:**
* **Public GamePanel:** gp.
* **Public boolean:** checkUI.
* **Public Int:** Uirow.
* **Public BufferedImage:** image, button.
* **Phương thức:**
* **Public UI( GamePanel gp):** Hàm tạo một đối để khởi tạo đối tượng và lấy hình ảnh.
* **Public void draw( Graphics2D g2):** Hàm void để vẽ giao diện lên panel.
  1. **Gói Menu.**
     + - 1. **Lớp Menu.**
* **Chức năng:** Làm lớp cha của các lớp menu con khác, lưu một số thuộc tính dùng chung.
* **Thuộc tính:**
* **Public Int:** MnX, MnY, width, height, slotCol, slotRow, cursorX, cursorY, cursorWidth, cursorHeight, slotXstart, slotYstart, ifX, ifY, ifWidth, ifHeight, seedTomato, seedPotato, seedCorn, seedCarrot, FootPig, FootCow, FootChicken.
* **Public Boolean:** check, checkEnter.
* **Public BufferedImage:** IseedTomato, IseedPotato, IseedCorn, IseedCarrot, Ipig, ICow, Ichicken, IfoodPig, IfoodCow, IfoodChicken.
* **Public Color:** c.
* **Public GamePanel:** gp.
  + - * 1. **Lớp HouseMenu.**
* **Chức năng:** Hiển thị Menu chứa thông tin nông sản, vật phẩm, số tiền của người chơi.
* **Thuộc tính:**
* **Public Int:** coin, tomato, potato, corn, carrot, Ptomato, Ppotato, Pcorn, Pcarrot.
* **Public BufferedImage:** itomato, ipotato, icorn, icarrot.
* **Phương thức:**
* **Public HouseMenu(GamePanel gp):** Hàm tạo một đối để khởi tạo giá trị cho các đối tượng và hình ảnh.
* **Public void draw(Graphics2D g2):** Đây là hàm kiểu void dùng để vẽ menu cùng các thông tin vật phẩm.
  + - * 1. **Lớp AnimalMenu.**
* **Chức năng:** Hiển thị thông tin khi chạm vào con vật.
* **Thuộc tính:**
* **Public Int:** pig, cow, chicken.
* **Phương thức:**
* **Public AnimalMenu(GamePanel gp):** Hàm tạo một đối khởi tạo đối tượng và giá trị thuộc tính.
* **Public void draw(Graphics2D g2):** Hàm void dùng để vẽ menu này lên màn hình.
  + - * 1. **Lớp LandTagMenu.**
* **Chức năng:** Hiển thị số hạt giống để người chơi lựa chọn khi trồng cây.
* **Thuộc tính:**
* **Public Int:** checkTree.
* **Public BufferedImage:** tomato, corn, potato, carrot.
* **Phương thức:**
* **Public LandTagMenu(GamePanel gp):** Hàm tạo một đối khởi tạo đối tượng và hình ảnh.
* **Public void draw(Graphics2D g2):** Hàm void hiển thị số hạt giống để người chơi lựa chọn.
  + - * 1. **Lớp TrayMenu.**
* **Chức năng:** Hiển thị số lượng thức ăn hiện có trong chuồng.
* **Thuộc tính:**
* **Public Int:** foodPig, foodCow, foodChicken.
* **Public BufferedImage:** pig, cow, chicken.
* **Phương thức:**
* **Public TrayMenu(GamePanel gp):** Hàm tạo một đối khởi tạo đối tượng, giá trị thuộc tính và hình ảnh.
* **Public void draw(Graphics2D g2):** Vẽ số lượng thức ăn hiện có.
  + - * 1. **Lớp ShopMenu.**
* **Chức năng:** Là menu của cửa hàng có chức năng mua vật phẩm.
* **Thuộc tính:**
* **Public Int:** Pig, Cow, Chicken, land, price, PseedTomato, PseedPotato, PseedCorn, PseedCarrot, Ppig, Pcow, PChicken, PfoodPig, PfoodCow, PfoodChicken.
* **Public BufferedImage:** iland.
* **Phương thức:**
* **Public ShopMenu(Gamepanel gp):** Hàm tạo một đối dùng để khởi tạo đối tượng.
* **Public void setprice():** Hàm void khởi tạo giá trị cho các đối tượng.
* **Public draw():** Hàm void để vẽ những vật phẩm có thể mua.
  1. **Gói Main.**
     + - 1. **Lớp Main.**
* **Chức năng:** Chạy 2 luồng âm thanh và hình ảnh.
  + - * 1. **Lớp View.**
* **Chức năng:** Kế thừa lớp JFrame để chứa một Jpanel hiển thị trò chơi.
* **Thuộc tính:**
* **Public GamePanel:** panel.
* **Phương thức:**
* **Public View():** Hàm tạo không đối cài những thuộc tính cho JFrame.
  1. **Gói object.**

1. **Lớp object.**

* **Chức năng**: Là lớp cha của tất cả các lớp đối tượng vật thể ( cánh cửa, đất ,củ quả ... ) trong gói.
* **Thuộc tính:**
  + - **Public int:** price, worldX, worldY, solidAreaDefaultX, solidAreaDefaultY, checkTree.
    - **Public boolean**: landKey, collision, checkRipen.
    - **Public BufferedImage:** image.
    - **Public String:** name.
    - **Public Rectangle:** solidArea.
    - **Public long**: time1, time2, time3, time4.
* **Phương thức:**
* **Public void draw(Graphics2D g2, GamePanel panel) :** Vẽ đối tượng.

1. **Lớp entity.**

* **Chức năng**:

Là lớp cha của tất cả các lớp đối tượng thực thể (con vật, người chơi...) trong gói.

* **Thuộc tính:**
  + - **public int:** food, worldX, worldY, speed, spriteCount, spriteNum, solidAreaDefaultX, solidAreaDefaultY.
    - **public long:** time1, time2, time3.
    - **public BufferedImage**: up1, up2, up3, up4 , down1, down2, down3, down4, right1, right2, right3, right4, left1, left2, left3, left4.
    - **public String:** direction, name.
    - **public boolean:** collisionOn, check.
    - **public Rectangle:** solidArea.
* **Phương thức:**
* **public void draw(Graphics2D g2).**
* **public void update().**

1. **Lớp buttterfly.**

* **Chức năng**: Lớp thiết lập thực thể butterfly (con bướm) kế thừa từ lớp entity.
* **Thuộc tính:**
* **Public int:** x.
* **Public GamePanel:** gp.
* **Phương thức:**
* **public void update():** Lớp ghi đè cập nhật trang thái thực thể.
* **public void setAction():** Ngẫu nhiên lựa chọn hướng đi của thực thể.
* **public void draw(Graphics2D g2):** Lớp ghi đè để hiển thị thực thể.

1. **Lớp chicken.**

* **Chức năng**:Thiết lập thực thể chicken thừa từ lớp entity.
* **Thuộc tính:**
* **public int**: x.
* **public GamePanel**: gp.
* **Phương thức:**
  + - **public chicken(GamePanel gp):** Hàm tạo một đối khởi tạo đối tượng và hình ảnh.
    - **public void setDefaultValues() :** Cài các giá trị mặc định.
    - **public void getImages():** Lấy hình ảnh.
    - **public void update():** Cập nhật trạng thái thực thể.
    - **public void setAction():** Cập nhật hướng đi ngẫu nhiên.
    - **public void draw(Graphics2D g2):** Hiển thị thực thể.

1. **Lớp cow.**

* **Chức năng**: Thiết lập thực thể cow kế thừa từ lớp entity.
* **Thuộc tính:**
* **public int**: x.
* **public GamePanel:** gp.
* **Phương thức:**
  + - **public chicken(GamePanel gp):** Hàm tạo một đối khởi tạo đối tượng và hình ảnh.
    - **public void setDefaultValues() :** Cài các giá trị mặc định.
    - **public void getImages():** Lấy hình ảnh.
    - **public void update():** Cập nhật trạng thái thực thể.
    - **public void setAction():** Cập nhật hướng đi ngẫu nhiên.
    - **public void draw(Graphics2D g2):** Hiển thị thực thể.

1. **Lớp pig.**

* **Chức năng**: Thiết lập thực thể pig kế thừa từ lớp entity.
* **Thuộc tính:**
* **public int:** x.
* **public GamePanel**: gp.
* **Phương thức:**
  + - **public chicken(GamePanel gp):** Hàm tạo một đối khởi tạo đối tượng và hình ảnh.
    - **public void setDefaultValues():** Cài các giá trị mặc định.
    - **public void getImages():** Lấy hình ảnh.
    - **public void update():** Cập nhật trạng thái thực thể.
    - **public void setAction():** Cập nhật ngẫu nhiên hướng đi.
    - **public void draw(Graphics2D g2):** Vẽ thực thể.

1. **Lớp fruit\_carrot.**

* **Chức năng**: Thiết lập đối tượng cà rốt kế thừa từ lớp object.
* **Thuộc tính:**
* **Phương thức:**
* **public fruit\_carrot():** Hàm tạo cài đặt sẵn các giá trị, phương thức mặc định khi tạo ra đối tượng.
* **public void draw(Graphics2D g2, GamePanel panel):** Hàm vẽ đối tượng.

1. **lớp fruit\_corn.**

* **Chức năng**: Thiết lập đối tượng ngô kế thừa từ lớp object.
* **Thuộc tính:**
* **Phương thức:**
* **public fruit\_corn():** Hàm tạo cài đặt sẵn các giá trị, phương thức mặc định khi tạo ra đối tượng.
* **public void draw(Graphics2D g2, GamePanel panel):** hàm vẽ đối tượng.

1. **Class fruit\_tomato**

* **Chức năng**: Thiết lập đối tượng cà chua kế thừa từ lớp object.
* **Thuộc tính:**
* **Phương thức:**
* **public fruit\_tomato():** Hàm tạo cài đặt sẵn các giá trị, phương thức mặc định khi tạo ra đối tượng.
* **public void draw(Graphics2D g2, GamePanel panel):** hàm vẽ đối tượng.

1. **Class fruit\_potato.**

* **Chức năng**: Thiết lập đối tượng khoai tây kế thừa từ lớp object
* **Thuộc tính:**
* **Phương thức:**
* **public fruit\_carrot():** Hàm tạo cài đặt sẵn các giá trị, phương thức mặc định khi tạo ra đối tượng.
* **public void draw(Graphics2D g2, GamePanel panel):** Hàm vẽ đối tượng.

1. **Lớp gate.**

* **Chức năng**: Thiết lập vật thể cổng kế thừa từ lớp object.
* **Thuộc tính:**
* **public BufferedImage**: closegate, opengate.
* **Phương thức:**
* **public gate():** Hàm tạo cài đặt sẵn các giá trị, phương thức mặc định khi tạo ra đối tượng.
* **public void draw(Graphics2D g2, GamePanel panel):** Hàm vẽ đối tượng.

1. **Lớp house.**

* **Chức năng**: thiết lập vật thể house(ngôi nhà) kế thừa từ lớp object.
* **Thuộc tính:**

**Phương thức:**

* **public gate():** Hàm tạo cài đặt sẵn các giá trị, phương thức mặc định khi tạo ra đối tượng.
* **public void draw(Graphics2D g2, GamePanel panel):** Hàm vẽ đối tượng.

1. **Lớp land.**

* **Chức năng**: Thiết lập vật thể đất trồng kế thừa từ lớp object.
* **Thuộc tính:**
* **Public BufferedImage:** tomato[] , potato[], corn[], carrot[], watering1, watering2, image1, image2.
* **Phương thức:**
* **public house():** Hàm tạo cài đặt sẵn các giá trị, phương thức mặc định khi tạo ra đối tượng.
* **public void draw(Graphics2D g2, GamePanel panel):** Hàm vẽ đối tượng.

1. **Lớp landtag.**

* **Chức năng**: Thiết lập vật thể kế thừa từ lớp object.
* **Thuộc tính:**
* **Phương thức:**
* **public landtag ():** Hàm tạo cài đặt sẵn các giá trị, phương thức mặc định khi tạo ra đối tượng.
* **public void draw(Graphics2D g2, GamePanel panel):** Hàm vẽ đối tượng.

1. **Lớp player.**

* **Chức năng**: Thiết lập thực thể player kế thừa từ lớp entity.
* **Thuộc tính:**
* **public int**: spriteCount, spriteNum, screenX, screenY.
* **public GamePanel**: gp.
* **public keyHandler:** keyh.
* **public boolean:** checkgate.
* **Phương thức:**
* **public player ():** Hàm tạo cài đặt sẵn các giá trị, phương thức mặc định khi tạo ra đối tượng.
* **public void getPlayerImages():** Hàm lấy hình ảnh đối tượng.
* **public void update():** Hàm cập nhật trạng thái đối tượng.
* **public void setTree(int i):** Hàm tương tác trồng cây với các đối tượng cây quả.
* **public void getTree(int i):** Hàm thu hoạch cây trồng khi người chơi tương tác.
* **public void Menu(int i):** hàm hiện menu khi nhân vật tương tác với các vật thể có bảng menu.
* **public void draw(Graphics2D g2, GamePanel panel):** Hàm vẽ đối tượng.

1. **Lớp shop.**

* **Chức năng**: Thiết lập vật thể shop (cửa hàng) kế thừa từ lớp object.
* **Thuộc tính:**
* **Phương thức:**
* **public shop():** Hàm tạo cài đặt sẵn các giá trị,phương thức mặc định khi tạo ra đối tượng.
* **public void draw(Graphics2D g2, GamePanel panel):** Hàm vẽ đối tượng.

1. **Lớp tray.**

* **Chức năng**: Thiết lập vật thể tray (khay thức ăn) kế thừa từ lớp object.
* **Thuộc tính:**
* **Phương thức:**
* **public tray():** Hàm tạo cài đặt sẵn các giá trị, phương thức mặc định khi tạo ra đối tượng.
* **public void draw(Graphics2D g2, GamePanel panel):** Hàm vẽ đối tượng.
  1. **Gói map.**

1. **Lớp tileManager.**

* **Chức năng**: Thiết lập, cài đặt các ô gạch trong game.
* **Thuộc tính:**
* **public GamePanel:** gpTile.
* **public tiles[]:** tiles.
* **public int:** map[][].
* **public String**: line.
* **Phương thức:**
* **public tileManager(GamePanel gpTile):** Hàm tạo cài đặt sẵn các giá trị đối tượng khi được tạo ra.
* **public void loadMap():** Hàm lấy dữ liệu của map.
* **public void getTileImage():** Hàm lấy hình ảnh.
* **public void setup(int index, String imgpath, boolean collision):** Hàm cài đặt các ô gạch **public void draw(Graphics2D g2):** Hàm vẽ bản đồ gồm nhiều ô gạch.

1. **Lớp tiles.**

* **Chức năng**: Thiết lập, cài đặt các ô hình trong game, là lớp cha của các ô vuông.
* **Thuộc tính:**
* **public BufferedImage** image.
* **public boolean** collision.
  1. **Gói Key\_Handler.**
     + - 1. **Lớp keyHandler.**
* **Chức năng**: Thiết lập, cài đặt các thao tác tương tác từ bàn phím thay đổi các giá trị của thuộc tính.
* **Thuộc tính:**
* **public boolean:** up, down, left, right.
* **public GamePanel:** gp.
* **Phương thức:**
  + - **public keyHandler(GamePanel Panel) :** Hàm tạo thiết lập sẵn các giá trị khi tạo đối tượng.
    - **public void keyPressed(KeyEvent e):** Hàm thực hiện thao tác khi có sự kiện nút bấm được bấm.
    - **public void keyReleased(KeyEvent e):** Hàm thực hiện thao tác khi có sự kiện nút bấm được thả ra.
  1. **Gói database.**
     + - 1. **Lớp JDBCUtil.**
* **Chức năng**: Thiết lập kết nối dữ liệu với SQL sever.
* **Thuộc tính:**
* **public GamePanel:** gp.
* **public Connection:** con.
* **Phương thức:**
  + - **public JDBCUtil(GamePanel gp):** Hàm tạo thiết lập sẵn các giá trị khi tạo đối tượng.
    - **public void update():** Hàm cập nhật dữ liệu vào SQL sever.
    - **public void delete():** Hàm xóa dữ liệu ở SQL sever.
    - **public void getdata():** Hàm lấy dữ liệ ở SQL server.

1. **Giới thiệu và mô tả chức năng của trò chơi.**
   1. **Giới thiệu:**

* Trong trò chơi, có 4 loại nông sản ( cà chua, ngô, cà rốt, khoai tây) tương ứng với 4 loại hạt giống.



* Có 3 con vật ( lợn, gà, bò ) tương ứng với 3 loại thức ăn.



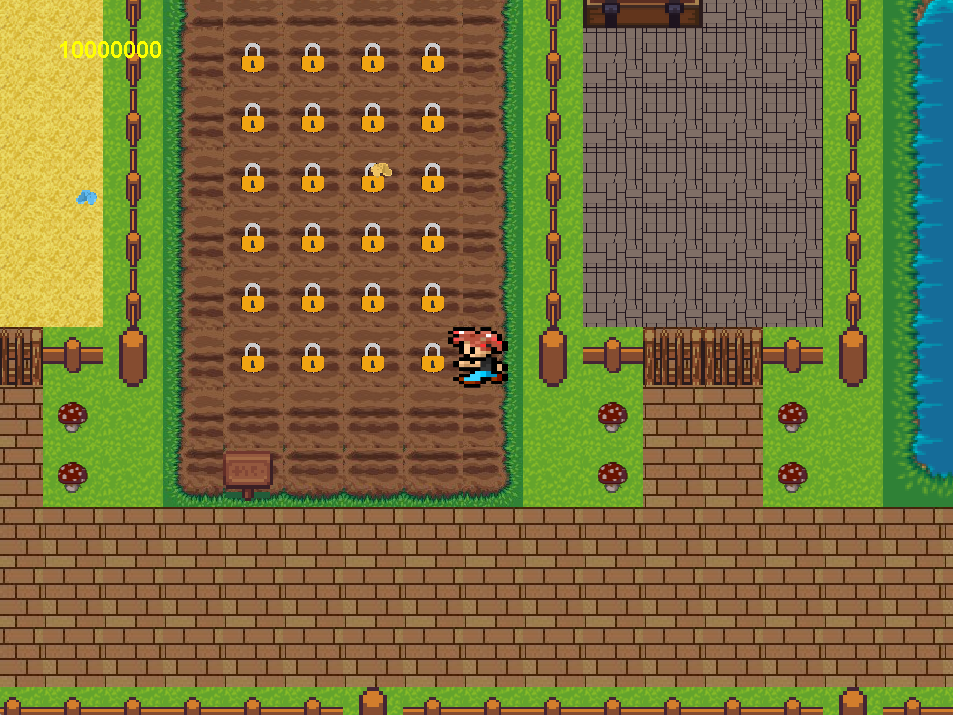
* Có một kho đồ để chứa nông sản, một cửa hàng để mua vật phẩm ( hạt giống, con vật, thức ăn ).



* 1. **Chức năng của trò chơi:**
* Để bắt đầu trò chơi mới, người chơi chọn vào biểu tượng New Game.



* Người chơi bắt đầu với 4 ô đất được mở khóa, có thể trồng các loại hạt giống khác nhau. Khi đủ tiền, có thể mở thêm ô đất trong cửa hàng, giá của ô đất sẽ tăng dần.



* Để trồng cây, người chơi chọn hạt giống trong đối tượng LandTag, nhấn Enter để chọn loại hạt, nhấn phím C để thoát LandTag, sau đó thực hiện va chạm với ô đất để trồng cây.



* Sau một khoảng thời gian, cây có thể được thu hoạch bằng cách va chạm với ô đất đó, đồng thời làm mới ô đất và nông sản trong kho cũng tăng lên.
* Thực hiện va chạm với đối tượng Tray để bỏ thức ăn vào chuồng, khi con vật va chạm với đối tượng Tray thì thức ăn sẽ được cộng vào cho con vật và thức ăn trong Tray sẽ bị trừ.



* Khi con vật ăn đủ thức ăn thì có thể bán và mua con vật mới trong cửa hàng.



* 1. **Kết nối cơ sở dữ liệu**

Trong cơ sở dữ liệu, nhóm em dử dụng SQL server để tạo ra 1 cơ sử dữ liệu tên là “FarmGame2D” trong đó có các bảng ‘house’ để chứa dữ liệu các vật phhamr, nông sản thu hoạch hoặc đã mua, bảng “shop” để lưu trữ số ô đất và giá đất, bảng ‘animal’ để lưu số động vật, bảng land để lưu trạng thái các ô đất, bảng tray để lưu số thức ăn của các con vật, bảng ‘time tree’ để lưu giữ các giai đoạn cây trồng .

1. **Hướng dẫn chạy mã lệnh trên máy tính.**

**1.đối với hệ điều hành window**

* Bước 1: Tải mã nguồn về sang máy tính .
* Bước 2: Cài đặt JDK trên máy tính nếu chưa có ,có thể tải JDK từ trang web chính thức của Oracle và cài đặt theo hướng dẫn .
* Bước 3: Thiết lập biến môi trường PATH để máy tính có thể tìm thấy JDK. Đối với hệ điều hành Windows :
  + Bước 1: Mở thanh tim kiếm của máy tính và tìm từ khóa "system" để mở cửa sổ system properties.
  + Bước 2: Chọn " Advanced system settings" bên cạnh " Environment Variables".
  + Bước 3: Nhấp vào nút " Environment Variables".
  + Bước 4: Tìm biến PATH và nhấp vào nút " Edit".
  + Bước 5: Thêm đường dẫn đến thư mục bin của JDK vào cuối của giá trị biến PATH. Ví dụ: C:\Program Files\Java\jdk1.8.0\_291\bin
* Bước 4: Mở Command Prompt hoặc Terminal và di chuyển đến thư mục chứa mã nguồn của ứng dụng. Sử dụng lệnh "cd" để di chuyển đến thư mục chứa tệp .java.
* Bước 5: Biên dịch mã nguồn bằng lệnh "javac". Ví dụ: javac main.java Lệnh này sẽ tạo ra các tệp .class.
* Bước 6: Chạy ứng dụng bằng lệnh "java" ví dụ: java main. Lệnh này sẽ chạy chương trình và hiển thị kết quả trên màn hình.
  1. **Đối với hệ điều hành IOS/Linux**

Mở Command Prompt (hoặc Terminal trên macOS/Linux).

* Di chuyển đến thư mục chứa mã nguồn của dự án bằng cách sử dụng lệnh "cd". Ví dụ: nếu mã nguồn của dự án nằm trong thư mục "C:\my\_project", bạn có thể nhập lệnh sau: cd C:\my\_project
* Sau đó , sử dụng lệnh javac để biên dịch tệp mã nguồn của dự án
* Lệnh này sẽ biên dịch tệp mã nguồn MyClass.java thành tệp bytecode MyClass.class. Nếu mã nguồn của bạn bao gồm nhiều tệp, hãy nhập lệnh biên dịch cho mỗi tệp.
* Sau khi biên dịch, các tệp bytecode sẽ được tạo ra trong cùng thư mục với tệp mã nguồn của dự án. Bạn có thể chạy ứng dụng bằng cách sử dụng lệnh "java".ví dụ : java MyClass

1. **Phần kết.**

Thông qua việc tìm hiểu và tự tạo nên một trò chơi giúp nhóm em hiểu thêm về ngôn ngữ Java. Tuy trò chơi đã gần như hoàn thiện, nhưng vẫn còn một số vấn đề đặt ra: đồ họa trò chơi vẫn chưa đẹp mắt, phần âm thanh chưa sinh động, phần kết nối cơ sở dữ liệu , mở rộng bản đồ, thêm các chức năng cho game, tuy có một số khó khăn nhưng cơ bản đã hoàn thiện đầy đủ các chức năng: lấy dữ liệu, thêm dữ liệu và xóa dữ liệu. Trong quá trình làm bài tập lớn cũng như viết báo cáo này, không thể tránh khỏi những thiếu sót, rất mong nhận được sự góp ý của Thầy và các bạn. Cuối cùng, nhóm chúng em cảm ơn giảng viên bộ môn và các bạn trong lớp đã hỗ trợ, giúp đỡ nhóm trong quá trình hoàn thiện bài tập lớn này.