

Họ tên	Chức năng	Vị trí chức năng
Tiến Anh	<ul style="list-style-type: none"> <li>• reset game</li> <li>• Thiết lập chế độ cho game</li> <li>• Bắt đầu(tạm dừng) game</li> </ul>	<ul style="list-style-type: none"> <li>• resetGame() trong Worm</li> <li>• modeGame() trong Worm</li> <li>• pauseGame() trong Worm</li> <li>• keyPressed() trong InGameScreen nhưng chỉ xem cmt "// <u>Khi người dùng ấn SPACE BAR thì game sẽ bắt đầu</u>"</li> </ul>
Thái Bảo	<ul style="list-style-type: none"> <li>• Thiết lập hướng di chuyển</li> <li>• Chuyển động của sâu</li> <li>• Tăng tốc độ cho sâu</li> </ul>	<ul style="list-style-type: none"> <li>• setVertot() trong Worm</li> <li>• moveHeadWorm() trong Worm</li> <li>• moveWorm() trong Worm</li> <li>• keyPressed() trong InGameScreen nhưng chỉ xem cmt "// <u>Xử lý sự kiện từ bàn phím</u>"</li> <li>• getCurrentSpeed() trong Worm</li> </ul>
Trung Thành	<ul style="list-style-type: none"> <li>• Nút chuyển đổi giao diện(vẽ backGroup cho game)</li> <li>• Hiển thị text thông báo để hướng dẫn người</li> <li>• Tạo con sâu và quả táo</li> </ul>	<p>Class <u>SwitchGameScreen</u> (xem hết mấy phương thức tạo frame) Class <u>InGameScreen()</u> + xem kỹ (trong class InGameScreen) <u>run();</u> <u>paintBg();</u> <u>paintFrame();</u> <u>paint();</u></p> <ul style="list-style-type: none"> <li>• modelnWorm() trong Worm</li> <li>• getModeApple() trong Worm</li> </ul>
Quang Trọng	<ul style="list-style-type: none"> <li>• Lưu trữ thông tin</li> <li>• Đọc dữ liệu từ file</li> <li>• Đưa dữ liệu vào frame thành tích</li> </ul>	<ul style="list-style-type: none"> <li>• Class DataUser</li> <li>• <u>insertPlayerDataOnLeaderBoard()</u></li> </ul>

**Chức năng:**

## 1) Thiết lập hướng di chuyển

- **Mô tả:** Thiết lập hướng di chuyển của con sâu dựa trên các sự kiện từ bàn phím.
- **Vị trí:** Phương thức `setVertot()` trong lớp `Worm`.

### Ý tưởng thực hiện:

- **Nhập hướng mới:** Nhận đầu vào từ người chơi để xác định hướng di chuyển mới.
- **Kiểm tra hướng hợp lệ:** Đảm bảo rằng hướng mới không ngược lại với hướng hiện tại (để tránh việc con sâu tự va vào mình).
- **Cập nhật hướng:** Nếu hợp lệ, cập nhật hướng di chuyển của con sâu.

### Pseudocode:

```
java
Sao chép mã
public void setVertot(Direction newDirection) {
    if (this.currentDirection != newDirection.getOpposite()) {
        this.currentDirection = newDirection;
    }
}
```

## 2. Chuyển động của sâu (moveHeadWorm)

### Chức năng:

- **Mô tả:** Di chuyển đầu của con sâu theo hướng hiện tại.
- **Vị trí:** Phương thức `moveHeadWorm()` trong lớp `Worm`.

### Ý tưởng thực hiện:

- **Tính toán vị trí mới:** Dựa trên hướng di chuyển hiện tại, tính toán vị trí mới cho đầu của con sâu.
- **Cập nhật vị trí:** Di chuyển đầu của con sâu đến vị trí mới.

### Pseudocode:

```
java
Sao chép mã
public void moveHeadWorm() {
    Position newHeadPosition = this.headPosition.move(this.currentDirection);
    this.positions.addFirst(newHeadPosition); // Thêm vị trí mới vào đầu danh sách
}
```

## 3. Di chuyển toàn bộ con sâu (moveWorm)

### Chức năng:

- **Mô tả:** Di chuyển toàn bộ con sâu, bao gồm cả việc cập nhật vị trí các phần thân của con sâu.
- **Vị trí:** Phương thức `moveWorm()` trong lớp `Worm`.

### Ý tưởng thực hiện:

- **Di chuyển đầu:** Gọi `moveHeadWorm()` để di chuyển đầu con sâu.
- **Di chuyển thân:** Di chuyển các phần thân theo, loại bỏ phần cuối cùng của con sâu để duy trì chiều dài.

### Pseudocode:

```
java
Sao chép mã
public void moveWorm() {
    this.moveHeadWorm();
    if (!this.growing) { // Nếu không đang lớn lên (ăn táo), xóa phần cuối
        this.positions.removeLast();
    } else {
        this.growing = false; // Đặt lại trạng thái lớn lên
    }
}
```

## 4. Xử lý sự kiện từ bàn phím (keyPressed)

### Chức năng:

- **Mô tả:** Xử lý các sự kiện từ bàn phím để điều khiển hướng di chuyển của con sâu.
- **Vị trí:** Phương thức `keyPressed()` trong lớp `InGameScreen`.

### Ý tưởng thực hiện:

- **Nhận sự kiện phím:** Lắng nghe và nhận sự kiện từ bàn phím.
- **Thiết lập hướng:** Gọi phương thức `setVertot()` của `Worm` để thiết lập hướng di chuyển mới dựa trên phím được nhấn.

### Pseudocode:

```
java
Sao chép mã
public void keyPressed(KeyEvent e) {
    switch (e.getKeyCode()) {
        case KeyEvent.VK_UP:
            worm.setVertot(Direction.UP);
            break;
        case KeyEvent.VK_DOWN:
```

```

        worm.setVertot(Direction.DOWN);
        break;
    case KeyEvent.VK_LEFT:
        worm.setVertot(Direction.LEFT);
        break;
    case KeyEvent.VK_RIGHT:
        worm.setVertot(Direction.RIGHT);
        break;
    }
}

```

## 5. Tăng tốc độ cho sâu (getCurrentSpeed)

### Chức năng:

- **Mô tả:** Xác định và trả về tốc độ hiện tại của con sâu, có thể thay đổi tùy thuộc vào độ dài của con sâu hoặc các yếu tố khác.
- **Vị trí:** Phương thức `getCurrentSpeed()` trong lớp `Worm`.

### Ý tưởng thực hiện:

- **Tính toán tốc độ:** Tính toán tốc độ hiện tại dựa trên độ dài của con sâu hoặc các yếu tố khác như cấp độ hiện tại.
- **Trả về tốc độ:** Trả về giá trị tốc độ hiện tại.

### Pseudocode:

```

java
Sao chép mã
public int getCurrentSpeed() {
    int baseSpeed = 100; // Tốc độ cơ bản
    int lengthFactor = this.positions.size() / 5; // Giảm tốc độ khi con sâu
    dài hơn
    return Math.max(baseSpeed - lengthFactor * 10, 20); // Tốc độ tối thiểu
    là 20
}

```

## Tổng kết

Các chức năng trên cung cấp cơ sở cho việc điều khiển con sâu trong game Apple Worm. Việc thiết lập hướng di chuyển, di chuyển đầu và toàn bộ con sâu, xử lý sự kiện từ bàn phím, và tính toán tốc độ hiện tại đều là các yếu tố quan trọng để đảm bảo trò chơi diễn ra mượt mà và thú vị.