**TRƯỜNG ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**LỚP CNTTC**

---------oOo---------



**BÁO CÁO ĐỒ ÁN**

**MÔN : PHÁT TRIỂN ỨNG DỤNG TRÒ CHƠI**

**CHỦ ĐỀ : BÁO CÁO TRÒ CHƠI HERO’S JOURNEY**

**GVHD : NGUYỄN ĐỖ THÁI NGUYÊN**

**Nhóm sinh viên:**

1. **Nguyễn Thanh Huy Hoàng 43.01.104.056**
2. **Trần Thiên Phúc 43.01.104.134**

MỤC LỤC

[I. TỔNG QUAN: 3](#_Toc30724)

[1) Tên đồ án:. 3](#_Toc26063)

[2) Giới thiệu: 3](#_Toc15533)

[3) Mục tiêu: 3](#_Toc24328)

[4) Các tính năng trong trò chơi: 3](#_Toc7666)

[II. NỘI DUNG: 3](#_Toc15085)

[1) Danh sách các Class sử dụng trong game: 3](#_Toc17126)

[2) Thiết kế các Class: 5](#_Toc27446)

[3) Phân tích các tính năng trong game : ...............................................................](#_Toc12924)16

[III. KẾT QUẢ: 34](#_Toc27130)

[IV. HƯỚNG PHÁT TRIỂN: 35](#_Toc21003)

[V. TÀI LIỆU THAM KHẢO: 35](#_Toc22959)

1. **TỔNG QUAN:**
2. **Tên đồ án:** Hero’s Journey.
3. **Giới thiệu:** Hero’s Journey là một game platform 2D đi cảnh trên PC được thiết kế quaứng dụng làm game Unity và ngôn ngữ lập trình C#. Game cho phép người chơi điều khiển nhân vật chạy nhảy, vượt chướng ngại vật, ăn tiền, đánh quái vật. Khi người chơi diệt được boss thì sẽ xuất hiện cổng để qua màn tiếp theo.
4. **Mục tiêu:**

* Tìm hiểu về ứng dụng làm game Unity, các phần mềm hỗ trợ để lập trình một game hoàn chỉnh.
* Tìm hiểu sâu hơn về ngôn ngữ C# trên nền tảng Unity engine trong công nghệ làm game.
* Thiết kế ra được sản phẩm game 2D đi cảnh đánh quái cho PC.

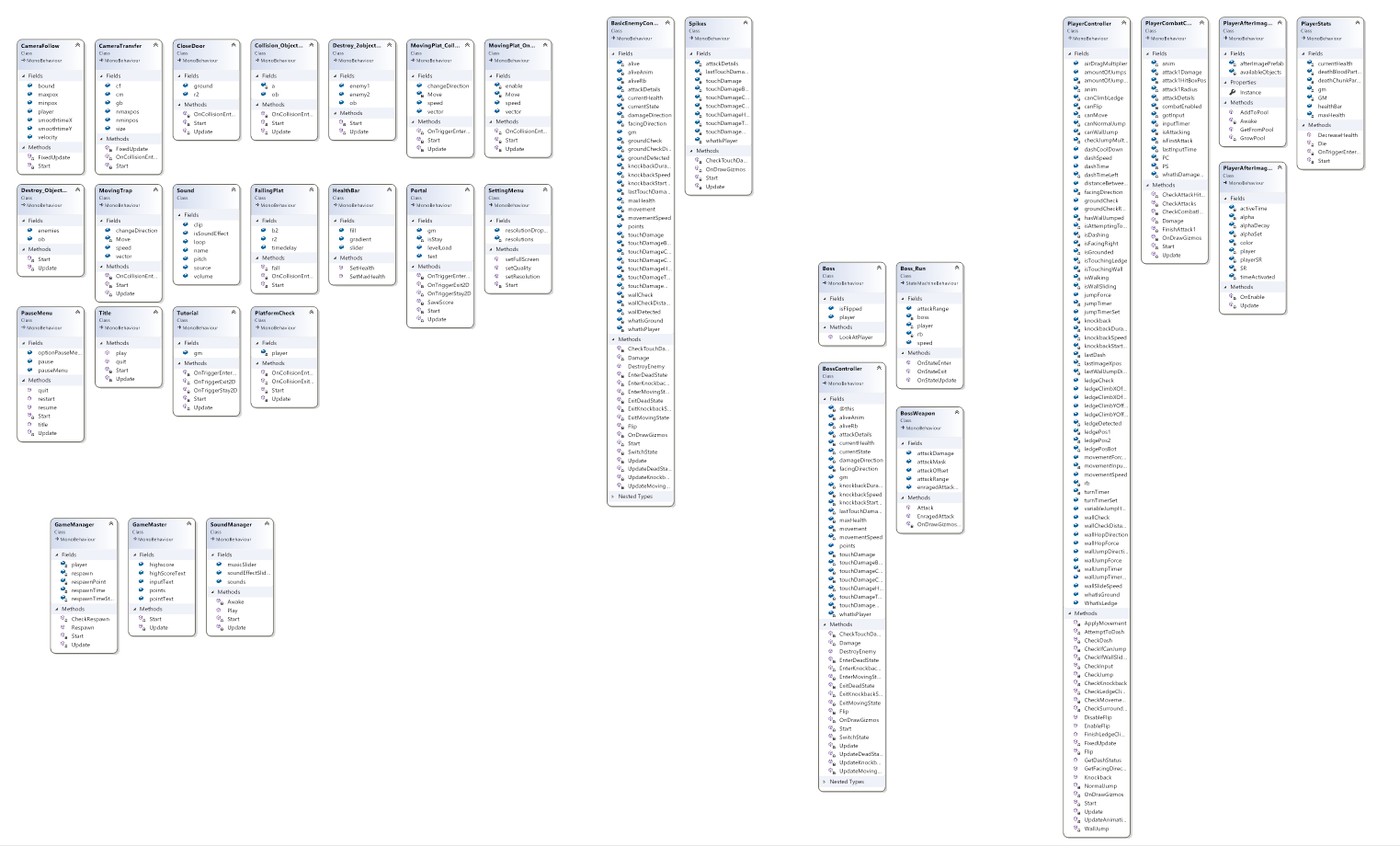
1. **Các tính năng trong trò chơi:**
   1. **Menu:**

* Start Menu
* Paused Menu
  1. **Gameplay:**
* Animation cho player và điều khiển chuyển đổi các trạng thái.
* Animation cho enemies.
* Điều khiển player.
* Player tương tác với enemies.
* Player tương tác với game object.
* Enemies tương tác với game object.
* Hiện Score, High Score, Health Bar cho Player.
* Môi trường, âm thanh, hiệu ứng trong trò chơi.
* Trùm.

1. **NỘI DUNG:**
2. **Danh sách các Class sử dụng trong game:**
   1. **Danh sách:**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên class** | **Chức năng** |
| 1 | BasicEnemyController | Chứa các hàm xử lý và hiển thị enemies |
| 2 | Spike | Chứa các hàm xử lý và hiển thị Spike |
| 3 | GameManager | Quản lý gameplay |
| 4 | GameMaster | Quản lý gameplay |
| 5 | SoundManager | Quản lý âm thanh |
| 6 | PlayerAfterImagePool | chứa các tàn ảnh từ PlayerAfterImageSprite và các xử lý liên quan |
| 7 | PlayerAfterImageSprite | tạo các sprite tàn ảnh của player |
| 8 | PlayerCombatController | Chứa các hàm xử lý đòn đánh của player |
| 9 | PlayerController | Chứa các hàm xử lý các hành động của player trừ tấn công. |
| 10 | PlayerStats | Chứa các hàm xử lý hp của player và khi player chết |
| 11 | CameraFollow | Chứa các hàm xử lý Camera đi theo player |
| 12 | CameraTransfer | Chuyển vị trí cho camera |
| 13 | CloseDoor | Chứa các hàm xử lý việc chuyển đổi RigidBody 2D từ Kinematic sang Dynamic thông qua va chạm |
| 14 | Collision\_ObjectAppear | Chứa các hàm xử lý việc va chạm làm set active animator cho enemies |
| 15 | Destroy\_2object\_Appear | Chứa các hàm xử lý việc phá hủy 2 đối tượng làm xuất hiện đồ vật |
| 16 | Destroy\_ObjectAppear | Chứa các hàm xử lý việc phá hủy 1 đối tượng làm xuất hiện đồ vật |
| 17 | FallingPlat | Chứa các hàm xử lý việc bậc thềm rơi tự do |
| 18 | HealthBar | Chứa các hàm xử lý cho thanh máu |
| 19 | MovingPlat\_Collider | Chứa các hàm xử lý cho việc bậc thềm chuyển động qua lại |
| 20 | MovingPlat\_OneTime\_Collider | Chứa các hàm xử lý cho việc bậc thềm chuyển động 1 lần duy nhất. |
| 21 | MovingTrap | Chứa các hàm xử lý cho việc bẫy có thể chuyển động |
| 22 | PauseMenu | Chứa các hàm xử lý cho việc quản lý và hiển thị pause menu |
| 23 | PlatformCheck | Chứa các hàm xử lý cho việc player chuyển động trên bậc thềm |
| 24 | Portal | Chứa các hàm xử lý và hiển thị portal qua màn |
| 25 | SettingMenu | Chứa các hàm xử lý và cho option menu |
| 26 | Sound | Chứa các biến về âm thanh |
| 27 | Title | Chứa các hàm xử lý và hiển thị Start Menu |
| 28 | Tutorial | Chứa các hàm xử lý về hiển thị hướng dẫn chơi game |
| 29 | Boss | Chứa hàm xử lý việc nhìn về phía người chơi |
| 30 | Boss\_Run | là State Behaviour gắn với anmation run của Boss chứa hàm xử lý việc đuổi theo và đánh người chơi |
| 31 | BossController | Chứa các hàm xử lý tác động Boss với player cũng như xử lý các thao tác giảm hp, chết của Boss |
| 32 | BossWeapon | Xử lý tấn công của Boss với người chơi |

* 1. **Sơ đồ class:**



1. **Thiết kế các Class:**
   1. **BasicEnemyController:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | State | enum | chứa các trạng thái của quái |
| 2 | currentState | State | ghi lại trạng thái hiện tại của quái |
| 3 | groundCheckDistance | float | chứa giá trị distance của Ground check khi dùng Raycast |
| 4 | wallCheckDistance | float | chứa giá trị distance của Wall check khi dùng Raycast |
| 5 | movementSpeed | float | tốc độ của quái |
| 6 | maxHealth | float | máu của quái |
| 7 | knockbackDuration | float | khoảng thời gian bị quái bị đẩy lùi |
| 8 | lastTouchDamageTime | float | lần cuối bị dính damage của player khi chạm quái |
| 9 | touchDamageCooldown | float | thời gian hồi khi gây damage va chạm |
| 10 | touchDamage | float | damage va chạm |
| 11 | touchDamageWidth | float | chiều rộng của hình vuông xác định damage va chạm |
| 12 | touchDamageHeight | float | chiều cao của hình vuông xác định damage va chạm |
| 13 | groundCheck | Transform | game object kiểm tra ground |
| 14 | wallCheck | Transform | game object kiểm tra wall |
| 15 | touchDamageCheck | Transform | game object kiểm tra damage va chạm |
| 16 | whatIsGround | LayerMask | xác định object nào là ground |
| 17 | whatIsPlayer | LayerMask | xác định object nào là player |
| 18 | knockbackSpeed | Vector2 | vector xác định đẩy lùi của quái |
| 19 | touchDamageBotLeft | Vector2 | vector dùng để xác định vùng gây damage đẩy lùi |
| 20 | touchDamageTopRight | Vector2 | vector dùng để xác định vùng gây damage đẩy lùi |
| 21 | currentHealth | float | máu hiện tại của quái |
| 22 | knockbackStartTime | float | thời gian bắt đầu bị đẩy lùi dùng để xác định thời điểm hết bị đẩy lùi |
| 23 | attackDetails | float[] | chứa thông tin của đòn va chạm |
| 24 | facingDirection | int | hướng quái đang hướng về dùng để đổi hướng quái khi di chuyển |
| 25 | damageDirection | int | hướng bị damage đẩy lùi |
| 26 | movement | Vector2 | vector xác định hướng bị đầy lùi |
| 27 | groundDetected | bool | xác định đang hay không chạm ground |
| 28 | wallDetected | bool | xác định đang hay không chạm wall |
| 29 | alive | GameObjcect | chứa reference đến game object của quái |
| 30 | aliveRb | Rigidbody2D | chứa reference đến Rigidbody2D của quái |
| 31 | aliveAnim | Animator | chứa reference đến Animator của quái |
| 32 | gm | GameMaster | chứa reference đến game object Game Master |
| 33 | point | int | chứa điểm của quái |
| 34 | Start() | void | thực hiện các set up khởi đầu cho quái |
| 35 | Update() | void | thực hiện các xử lý mỗi frame 1 lần |
| 36 | EnterMovingState() | void | Tiến vào trang thái di chuyển |
| 37 | UpdateMovingState() | void | Xử lý khi quái trong trạng thái di chuyển |
| 38 | ExitMovingState() | void | Kết thúc trạng thái di chuyển |
| 39 | EnterKnockbackState() | void | Tiến vào trang thái đẩy lùi |
| 40 | UpdateKnockbackState() | void | Xử lý khi quái trong trạng thái đẩy lùi |
| 41 | ExitKnockbackState() | void | Kết thúc trạng thái đẩy lùi |
| 42 | EnterDeadState() | void | Tiến vào trang thái chết |
| 43 | UpdateDeadState() | void | Xử lý khi quái trong trạng thái chết |
| 44 | ExitDeadState() | void | Kết thúc trạng thái chết |
| 45 | DestroyEnemy() | void | hủy game object của quái và cộng điểm |
| 46 | Damage(float[]) | void | xứ lý việc quái nhận damage |
| 47 | CheckTouchDamage() | void | xứ lý việc quái gây damge va chạm cho player |
| 48 | Flip() | void | đổi hướng mặt cúa quái |
| 49 | SwitchState(State) | void | Xử lý chuyển đổi giữa các trạng thái |
| 50 | OnDrawGizmos() | void | vẽ ra các object như wall check, ground check, touch damage check để test |

* 1. **Spike:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | lastTouchDamageTime | float | tương tự như xử lý touch damage cho quái |
| 2 | touchDamageCooldown | float | tương tự như xử lý touch damage cho quái |
| 3 | touchDamage | float | tương tự như xử lý touch damage cho quái |
| 4 | touchDamageWidth | float | tương tự như xử lý touch damage cho quái |
| 5 | touchDamageHeight | float | tương tự như xử lý touch damage cho quái |
| 6 | whatIsPlayer | LayerMask | tương tự như xử lý touch damage cho quái |
| 7 | touchDamageBotLeft | Vector2 | tương tự như xử lý touch damage cho quái |
| 8 | touchDamageTopRight | Vector2 | tương tự như xử lý touch damage cho quái |
| 9 | touchDameCheck | Transform | tương tự như xử lý touch damage cho quái |
| 10 | attackDetails | float[] | tương tự như xử lý touch damage cho quái |
| 11 | Start() | void | Không có gì |
| 12 | Update() | void | gọi hàm CheckTouchDamage() |
| 13 | CheckTouchDamage() | void | tương tự như xử lý touch damage cho quái |
| 14 | OnDrawGizmos() | void | tương tự như xử lý touch damage cho quái |

* 1. **GameManager:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | respawnPoint | Transform | Không sử dụng |
| 2 | player | GameObject | Không sử dụng |
| 3 | respawnTime | float | Không sử dụng |
| 4 | respawnTimeStart | float | Không sử dụng |
| 5 | respawn | bool | Không sử dụng |
| 6 | Start() | void | Không sử dụng |
| 7 | Update() | void | Không sử dụng |
| 8 | Respawn() | void | Không sử dụng |
| 9 | CheckRespawn() | void | Không sử dụng |

* 1. **GameMaster:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | point | int | điểm số mà player đạt được |
| 2 | highscore | int | high score mà player đạt được |
| 3 | pointText | Text | văn bản ghi điểm số của player trên Canvas |
| 4 | highScoreText | Text | văn bản ghi high score của player trên Canvas |
| 5 | inputText | Text | văn bản bạn muốn hiển thị trên Canvas |
| 6 | Start() | void | Khởi tạo các giá trị khi bắt đầu vào game |
| 7 | Update() | void | Cập nhật các thay đổi về pointText |

* 1. **SoundManager:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | sounds | Sound[] | Lưu danh sách dữ liệu đối tượng âm thanh |
| 2 | soundEffectSlider | Slider | Thanh slider điều chỉnh âm lượng hiệu ứng |
| 3 | musicSlider | Slider | Thanh slider điều chỉnh âm lượng nhạc nền |
| 4 | Awake() | void | Khởi tạo các Component Audio Source cho từng dữ liệu trong sounds |
| 5 | Start() | void | chơi nhạc nền và lấy giá trị âm lượng nhạc nền, hiệu ứng cho gán cho giá trị của 2 thanh slider soundEffect và music |
| 6 | Update() | void | Cập nhật các thay đổi về âm lượng của nhạc nền và hiệu ứng |
| 7 | Play(string) | void | Chơi nhạc |

* 1. **PlayerAfterImagePool:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | afterImagePrefab | GameObject | reference của object After Image |
| 2 | availableObjects | Queue<GameObject> | Queue chứa các After Image |
| 3 | Instance | PlayerAfterImagePool | instance của class này |
| 4 | Awake() | void | xử lý ban đầu |
| 5 | GrowPool() | void | thêm 10 after image vào pool |
| 6 | AddToPool(GameObject) | void | Hàm thêm object vào queue |
| 7 | GetFromPool() | GameObject | Lấy pool ra sử dụng |

* 1. **PlayerAfterImageSprite:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | activeTime | float | thời gian giữa các lần thêm vào Pool |
| 2 | timeActivated | float | thời gian khới động |
| 3 | alpha | float | chỉ số alpha của màu |
| 4 | alphaSet | float | chỉ số alpha cài sẵn |
| 5 | alphaDecay | float | chỉ số alpha giảm theo thời gian |
| 6 | player | Transform | reference của player |
| 7 | SR | SpriteRenderer | chứa reference của SpriteRenderer của object này |
| 8 | playerSR | SpriteRenderer | chứa reference của SpriteRenderer của player |
| 9 | color | Color | màu có chỉ số alpha giảm theo thời gian |
| 10 | OnEnable() | void | Hàm lưu SpriteRenderer của player vào object này |
| 11 | Update() | void | Hàm giảm chỉ số alpha của SpriteRenderer của object này rồi lưu vào Pool |

* 1. **PlayerCombatController:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | combatEnabled | bool | Kiểm tra có vào combat không |
| 2 | inputTimer | float | dùng để kiểm tra khoảng thời gian giữa những đòn đánh |
| 3 | attack1Radius | float | bán kính cong của vỏng tròn xác định đòn đánh |
| 4 | Attack1Damage | float | Damage của đòn đánh |
| 5 | Attack1HitBoxPos | Transform | object vòng tròn xác định đòn tấn công |
| 6 | whatIsDamageable | LayerMask | xác định object có thể gây damage |
| 7 | gotInput | bool | kiểm tra xem đã có input tấn công |
| 8 | isAttacking | bool | kiểm tra xem có đang tấn công |
| 9 | isFirstAttack | bool | kiểm tra xem có là đòn đầu tiên |
| 10 | lastInputTime | float | thời gian nhấn tấn công cuối |
| 11 | attackDetails | float[] | chi tiết đòn đánh bao gồm damage hướng đánh |
| 12 | anim | Animator | chứa reference của Animator của player |
| 13 | PC | PlayerController | chứa reference của class PlayerController |
| 14 | PS | PlayerStats | chứa reference của class PlayerStats |
| 15 | Start() | void | set up các giá trị ban đầu |
| 16 | Update() | void | gọi các hàm theo từng frame |
| 17 | CheckCombatInput() | void | Kiểm tra có nhấn nút Attack |
| 18 | CheckAttacks() | void | Xử lý đòn tấn công |
| 19 | CheckAttackHitBox() | void | Xứ lý gây damage đối tượng trong vòng tấn công |
| 20 | FinishAttack1() | void | Xử lý kết thúc đòn tấn công |
| 21 | Damage(float[]) | void | Xử lý player bị gây damage |
| 22 | OnDrawGizmos() | void | Vẽ ra vòng kiểm tra đòn đánh để test |

* 1. **PlayerController:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | movementInputDirection | float | hướng nút di chuyển |
| 2 | jumpTimer | float | thời gian nhảy |
| 3 | turnTimer | float | thời gian quay người |
| 4 | wallJumpTimer | float | thời gian nhảy khi đang bám tường |
| 5 | dashTimeLeft | float | thời gian dash còn lại |
| 6 | lastImageXpos | float | vị trí x của tàn ảnh gần nhất |
| 7 | lastDash | float | thời gian dash gần nhất |
| 8 | knockbackStartTime | float | thời điểm bắt đầu đẩy lùi |
| 9 | knockbackDuration | float | thời gian đẩy lùi |
| 10 | amountOfJumpsLeft | int | số lần nhảy còn lại |
| 11 | facingDirection | int | hướng đang đối mặt |
| 12 | lastWallJumpDirection | int | Hướng thực hiện nhảy nhảy khi đang bám tường gần nhất |
| 13 | isFacingRight | bool | kiểm tra xem có đang hướng về bên phải |
| 14 | isWalking | bool | kiểm tra xem có đang chạy |
| 15 | isGrounded | bool | kiểm tra xem có đang chạm ground |
| 16 | isTouchingWall | bool | kiểm tra xem có đang chạm wall |
| 17 | isWallSliding | bool | kiểm tra xem có đang trượt wall |
| 18 | canNormalJump | bool | kiểm tra xem có thể nhảy thường |
| 19 | canWallJump | bool | kiểm tra xem có thể nhảy khi đang bám tường |
| 20 | isAttemptingToJump | bool | kiểm tra xem có thể nhảy |
| 21 | checkJumpMultiplier | bool | kiểm tra xem có thể nhảy nhiều lần |
| 22 | canMove | bool | kiểm tra xem có thể di chuyển |
| 23 | canFlip | bool | kiểm tra xem có thể đổi hướng |
| 24 | hasWallJumped | bool | kiểm tra xem có nhảy khi đang bám tường |
| 25 | isTouchingLedge | bool | kiểm tra xem có chạm góc tường |
| 26 | canClimbLedge | bool | kiểm tra xem có trèo |
| 27 | ledgeDetected | bool | Kiềm tra có phải góc tường |
| 28 | isDashing | bool | kiểm tra xem có đang dash |
| 29 | knockback | bool | kiểm tra xem có đang bị đẩy lùi |
| 30 | knockbackSpeed | Vector2 | đẩy lùi player theo hướng x,y |
| 31 | ledgePosBot | Vector2 | vị trí của object wallcheck |
| 32 | ledgePos1 | Vector2 | xác định vị trí khi bắt đầu trèo tường |
| 33 | ledgePos2 | Vector2 | xác định vị trí sau khi trèo tường |
| 34 | rb | Rigidbody2D | chứa reference của Rigidbody2D của player |
| 35 | anim | Animator | chứa reference của Animator của player |
| 36 | amountOfJumps | int | chứa số lần nhảy |
| 37 | movementSpeed | float | tốc độ của player |
| 38 | jumpForce | float | lực nhảy của player |
| 39 | groundCheckRadius | float | Bán kính của vòng tròn kiểm tra ground |
| 40 | wallCheckDistance | float | Khoảng cách kiểm tra wall |
| 41 | wallSlideSpeed | float | tốc độ trượt tường |
| 42 | movementForceInAir | float | Không sử dụng |
| 43 | airDragMultiplier | float | giá trị giám tốc khi bám tường |
| 44 | variableJumpHeightMultiplier | float | giá trị độ cao khác nhau tùy lực bấm nhảy |
| 45 | wallHopForce | float | không sử dụng |
| 46 | wallJumpForce | float | Lực nhảy khi đang bám tường |
| 47 | jumpTimerSet | float | cái đặt ban đầu thời gian nhảy |
| 48 | turnTimerSet | float | cái đặt ban đầu thời gian quay |
| 49 | wallJumpTimerSet | float | cái đặt ban đầu thời gian nhảy khi đang bám tường |
| 50 | ledgeClimpXOffset1 | float | offset x để xác định vi trí khi bắt đầu trèo tường |
| 51 | ledgeClimpYOffset1 | float | offset y để xác định vi trí khi bắt đầu trèo tường |
| 52 | LedgeClimpXOffset2 | float | offset x để xác định vi trí sau khi trèo tường xong |
| 53 | LedgeClimpYOffset2 | float | offset y để xác định vi trí sau khi trèo tường xong |
| 54 | dashTime | float | thời gian dash |
| 55 | dashSpeed | float | tốc đô dash |
| 56 | distanceBetweenImages | float | khoảng cách giữa các tàn ảnh |
| 57 | dashCoolDown | float | thời gian hồi dash |
| 58 | wallHopDirection | Vector2 | Không sử dụng |
| 59 | wallJumpDirection | Vector2 | hướng nhảy nhảy khi đang bám tường |
| 60 | groundCheck | Transform | object để xác định ground |
| 61 | wallCheck | Transform | object để xác định wall |
| 62 | ledgeCheck | Transform | object để xác định ledge |
| 63 | whatIsGround | LayerMask | xác định các Layer Gound |
| 64 | whatIsLedge | LayerMask | xác định các Layer Ledge |
| 65 | Start() | void | set up các giá trị |
| 66 | Update() | void | gọi các hàm xử lý mỗi frame |
| 67 | FixedUpdate() | void | gọi các hàm xử lý vật lý cho player |
| 68 | CheckIfWallSliding() | void | kiểm tra xem player có đang bám tường |
| 69 | GetDashStatus() | bool | lấy trạng thái dash của player |
| 70 | Knockback(int) | void | xử lý đẩy lùi cho player |
| 71 | CheckKnockback() | void | Xử lý đẩy lùi player |
| 72 | CheckLedgeClimb() | void | Xử lý player nhảy trèo tường |
| 73 | FinishLedgeClimb() | void | Xử lý player sau khi trèo tường xong |
| 74 | CheckSurrounding() | void | Kiếm tra playert có chạm ground, wall hay ledge |
| 75 | CheckIfCanJump() | void | Kiễm tra player có thể nhảy nữa không |
| 76 | CheckMovementDirection() | void | Xử lý hướng di chuyển của player |
| 77 | UpdateAnimation() | void | update animation của player theo frame |
| 78 | CheckInput() | void | Kiểm tra input được nhập và các xử lý tương ứng |
| 79 | AttemptToDash() | void | chuẩn bị các giá trị trước khi dash |
| 80 | GetFacingDirection() | int | lấy hướng của player |
| 81 | CheckDash() | void | Xử lý dash của player |
| 82 | CheckJump() | void | Xử lý các kiểu nhảy như nhảy thường nhảy khi đang bám tường |
| 83 | NormalJump() | void | Xử lý nhảy thường |
| 84 | WallJump() | void | Xử lý nhảy khi đang bám tường |
| 85 | ApplyMovement() | void | Xử lý di chuyển của player |
| 86 | DisableFlip() | void | vô hiệu hóa xoay hướng |
| 87 | EnableFlip() | void | cho phép xoay hướng |
| 88 | Flip() | void | Xử lý xoay hướng nhân vật |
| 89 | OnDrawGizmos() | void | vẽ ra các object check để test |

* 1. **PlayerStats:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | maxHealth | float | máu tối đa của player |
| 2 | healthBar | HealthBar | chứa reference  của class HealthBar |
| 3 | deathChunkParticle | GameObject | Không sử dụng |
| 4 | deathBloodParticle | GameObject | Không sử dụng |
| 5 | currentHealth | float | máu hiện tại của nhân vật |
| 6 | GM | GameManager | chứa reference  của class GameManager |
| 7 | gm | gameMaster | chứa reference  của class gameMaster |
| 8 | Start() | void | cài đặt các giá trị ban đầu |
| 9 | DecreaseHealth(float) | void | Xử lý giảm máu nhân vật |
| 10 | Die() | void | Xử lý khi nhân vật chết |
| 11 | OnTriggerEnter2D(Collider2D) | void | Xử lý việc va chạm với đồng tiền |

* 1. **CameraFollow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | smoothtimeX | float | Thời gian trì hoãn việc camera di chuyển theo player theo phương x |
| 2 | smoothtimeY | float | Thời gian trì hoãn việc camera di chuyển theo player theo phương y |
| 3 | velocity | Vector2 | Vị trí thay đổi của camera dưới dạng velocity |
| 4 | player | GameObject | Đối tượng player |
| 5 | minpox | Vector2 | Vị trí tối thiểu mà camera cho phép đến theo phương x và y |
| 6 | maxpox | Vector2 | Vị trí tối đa mà camera cho phép đến theo phương x và y |
| 7 | bound | bool | Nếu là true thì camera chịu sự tác động của minpox và maxpox, còn false thì ngược lại |
| 8 | Start() | void | khởi tạo giá trị cho player |
| 9 | FixedUpdate() | void | Cập nhật sự biến đổi vị trí của camera khi di chuyển theo player, tùy theo resolution mà có các minpox và maxpos khác nhau |

* 1. **CameraTransfer:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | size | float | Size của camera |
| 2 | nminpos | Vector2 | Vị trí tối thiểu mới mà camera cho phép đến theo phương x và y |
| 3 | nmaxpos | Vector2 | Vị trí tối đa mới mà camera cho phép đến theo phương x và y |
| 4 | gb | GameObject | Đối tượng Game Object camera cần thay đổi |
| 5 | cf | CameraFollow | Chứa reference đến CameraFollow |
| 6 | cm | Camera | Chứa reference đến Camera |
| 7 | Start() | void | Khởi tạo giá trị cho cf và cm |
| 8 | FixedUpdate() | void | Hàm ảo |
| 9 | OnCollisionEnter2D(Collision2D) | void | Thực hiện việc gán minpox và maxpox với nminpos và nmaxpos đồng thời đổi size cho camera khi object va chạm với player |

* 1. **CloseDoor:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | ground | GameObject | Đối tượng cần thay đổi |
| 2 | r2 | Rigidbody2D | Chứa reference đến Rigidbody2D |
| 3 | Start() | void | Khởi tạo giá trị cho r2. |
| 4 | Update() | void | Hàm ảo |
| 5 | OnCollisionEnter2D(Collision2D) | void | Chuyển đổi rigid bodyType của đối tượng sang Dynamic khi object va chạm với player |

* 1. **Collision\_ObjectAppear:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | ob | GameObject | Đối tượng cần thay đổi. |
| 2 | a | Animator | Chứa reference đến Animator của |
| 3 | Start() | void | Khởi tạo giá trị cho a và vô hiệu hóa  a. |
| 4 | Update() | void | Hàm ảo |
| 5 | OnCollisionEnter2D(Collision2D) | void | Khi object va chạm với player và biến ob != null thì kích hoạt lại a |

* 1. **Destroy\_2object\_Appear:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | ob | GameObject[] | Danh sách đối tượng cần xuất hiện |
| 2 | enemy1 | GameObject | Đối tượng 1 cần tiêu diệt |
| 3 | enemy2 | GameObject | Đối tượng 2 cần tiêu diệt |
| 4 | Start() | void | Hàm ảo |
| 5 | Update() | void | Khi tiêu diệt enemy1 và enemy2 thì các đối tượng trong mảng ob xuất hiện |

* 1. **Destroy\_ObjectAppear:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | ob | GameObject[] | Danh sách đối tượng cần xuất hiện |
| 2 | enemies | Gameobject | Đối tượng cần tiêu diệt |
| 3 | Start() | void | Hàm ảo |
| 4 | Update() | void | Khi tiêu diệt enemies thì các đối tượng trong mảng ob xuất hiện |

* 1. **FallingPlat:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | r2 | Rigidbody2D | Rigidbody2D của Object |
| 2 | b2 | BoxCollider2D | BoxCollider2D của Object |
| 3 | timeDelay | float | thời gian trì hoãn việc rơi tự do của bậc thềm |
| 4 | Start | void | Lấy các giá trị cho r2 và b2 |
| 5 | OnCollisionEnter2D(Collision2D) | void | Nếu Object va chạm với player thì bậc thềm rơi |
| 6 | fall() | IEnumerator | Xử lý việc rơi tự do của bậc thềm sau khi trì hoãn |

* 1. **HealthBar:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | slider | Slider | chứa reference của Sider |
| 2 | gradient | Gradient | chứa reference của Gradient |
| 3 | fill | Image | Image thể hiện thanh máu |
| 4 | SetMaxHealth(float) | void | Cài đặt thanh máu với full máu |
| 5 | SetHealth(float) | void | xử lý thanh máu theo giá trị đưa vào |

* 1. **MovingPlat\_Collider:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | speed | float | Tốc độ chuyển động của bậc thềm |
| 2 | changeDirection | float | Chiều chuyển động của bậc thềm |
| 3 | vector | char | Phương chuyển động của bậc thềm |
| 4 | Move | Vector3 | hàm Vector3 |
| 5 | Start() | void | Gán Move với vị trí của Object |
| 6 | Update() | void | Cập nhật vị trí chuyển động của bậc thềm theo phương được chọn trước. Nếu pause thì ngừng chuyển động, còn không pause thì ngược lại |
| 7 | OnTriggerEnter2D(Collider2D) | void | Khi chịu va chạm với các Object nhận định là “Ground” hoặc “Spike” có set isTrigger thì sẽ đổi chiều |
| 8 | OnCollisionEnter2D(Collision2D) | void | Khi chịu va chạm với các Object nhận định là “Ground” hoặc “Spike” thì sẽ đổi chiều. |

* 1. **MovingPlat\_OneTime\_Collider:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | speed | float | Tốc độ chuyển động của bậc thềm |
| 2 | vector | char | Phương chuyển động của bậc thềm |
| 3 | enable | bool | Nếu true thì bậc thềm được quyền chuyển động, false thì speed = 0 |
| 4 | Move | Vector3 | hàm Vector3 |
| 5 | Start() | void | Gán Move với vị trí của Object |
| 6 | Update() | void | Cập nhật vị trí chuyển động của bậc thềm theo phương được chọn trước. Nếu pause thì ngừng chuyển động, còn không pause thì ngược lại |
| 7 | OnCollisionEnter2D(Collision2D) | void | Khi chịu va chạm với các Object là “Ground” hoặc “Spike” thì chuyển enable sang false, nếu chịu va chạm với player speed = 0.015f |

* 1. **MovingTrap:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | speed | float | Tốc độ chuyển động của bẫy |
| 2 | changeDirection | float | Chiều chuyển động của bẫy |
| 3 | vector | char | Phương chuyển động của bẫy |
| 4 | Move | Vector3 | hàm Vector3 |
| 5 | Start() | void | Gán Move với vị trí của Object |
| 6 | Update() | void | Cập nhật vị trí chuyển động của bẫy theo phương được chọn trước. Nếu pause thì ngừng chuyển động, còn không pause thì ngược lại |
| 7 | OnCollisionEnter2D(Collision2D) | void | Khi chịu va chạm với các Object nhận định là “Ground” hoặc “Spike” thì đổi chiều đồng thời tăng tốc độ chuyển động nếu speed > 0 và giảm nếu speed < 0. |

* 1. **PauseMenu:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | pause | bool | nếu true thì pause, nếu false thì ngược lại |
| 2 | pauseMenu | GameObject | Object pauseMenu |
| 3 | optionPauseMenu | GameObject | Object optionMenu |
| 4 | Start() | void | Bỏ kích hoạt Pause Menu |
| 5 | Update() | void | Cập nhật sự xuất hiện của Pause Menu |
| 6 | resume() | void | Bỏ trạng thái Pause |
| 7 | restart() | void | Reset lại màn chơi hiện tại |
| 8 | title() | void | Quay lại Start Menu |
| 9 | quit() | void | Thoát trò chơi |

* 1. **PlatformCheck:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | player | PlayerController | Chứa reference đến PlayerController |
| 2 | Start() | void | Khởi tạo giá trị |
| 3 | Update() | void | Hàm ảo |
| 4 | OnCollisionEnter2D(Collision2D) | void | Nếu player va chạm với object bị nhận định là “Moving Plat” thì chuyển động của player không bị cô lập với chuyển động của bậc thềm |
| 5 | OnCollisionExit2D(Collision2D) | void | Nếu player thoát va chạm với object bị nhận định là “Moving Plat” thì chuyển động của player độc lập với chuyển động của bậc thềm |

* 1. **Portal:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | levelLoad | int | Màn chơi cần load |
| 2 | gm | GameMaster | Chứa reference đến GameMaster |
| 3 | text | string | Văn bản cần xuất hiện |
| 4 | isStay | bool | Nếu true thì được quyền qua màn mới, false thì ngược lại |
| 5 | Start() | void | Gán false cho isStay và khởi tạo giá trị cho gm |
| 6 | Update() | void | Nếu trường hợp isStay = true và nhấn nút chỉ định thì sẽ qua màn mới |
| 7 | OnTriggerEnter2D(Collider2D) | void | Nếu player va chạm với Portal, hiện đoạn text hướng dẫn qua màn. Trong trường hợp là màn cuối thì sẽ hiện đoạn text cảm ơn. |
| 8 | OnTriggerStay2D(Collider2D) | void | Nếu player vẫn còn va chạm với player thì gán true cho isStay |
| 9 | OnTriggerExit2D(Collider2D) | void | Nếu player thoát va chạm với Portal thì gán false cho isStay |
| 10 | SaveScore() | void | Lưu điểm để sử dụng ở màn mới. Nếu màn hiện tại là màn cuối thì lưu High Score |

* 1. **SettingMenu:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | resolutionDropdown | Dropdown | Dropdown chứa độ phân giải màn hình |
| 2 | resolutions | Resolution[] | Danh sách độ phân giải màn hình |
| 3 | fullScreen | Toggle | Object Toggle bật tắt toàn màn hình |
| 4 | Start() | void | Khởi tạo giá trị cho resolutionDropdown |
| 5 | setResolution(int) | void | Chỉnh độ phân giải màn hình |
| 6 | setQuality(int) | void | Chỉnh Quality Graphic |
| 7 | setFullScreen(bool) | void | Chỉnh toàn màn hình hoặc dạng cửa sổ |

* 1. **Sound:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | name | string | Tên của audio |
| 2 | clip | AudioClip | Source của audio |
| 3 | volume | float | Âm lượng của audio |
| 4 | pitch | float | Cao độ của audio |
| 5 | loop | bool | Nếu muốn audio lặp lại thì true, không muốn thì false |
| 6 | isSoundEffect | bool | True nếu là hiệu ứng âm thanh, false là nhạc nền |
| 7 | source | AudioSource | Chứa reference đến AudioSource |

* 1. **Title:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | Start() | void | Hàm ảo |
| 2 | Update() | void | Hàm ảo |
| 3 | play() | void | Vào màn chơi đầu tiên |
| 4 | quit() | void | Thoát khỏi trò chơi |

* 1. **Tutorial:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | gm | GameMaster | Chứa reference đến Object GameMaster |
| 2 | Start() | void | Khởi tạo cho gm |
| 3 | Update() | void | Hàm ảo |
| 4 | OnTriggerEnter2D(Collider2D) | void | Nếu object va chạm với player thì đoạn text hướng dẫn sẽ xuất hiện |
| 5 | OnTriggerStay2D(Collider2D) | void | Nếu object vẫn đang va chạm với player thì đoạn text hướng dẫn sẽ xuất hiện |
| 6 | OnTriggerExit2D(Collider2D) | void | Nếu object thoát va chạm với player thì đoạn text hướng dẫn sẽ biến mất |

* 1. **Boss**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | player | Transform | xác định player |
| 2 | isFlipped | bool | Xác định xem có xoay hướng |
| 3 | LookAtPlayer() | void | Hàm xử lý hướng về phía player |

* 1. **Boss\_Run:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | speed | float | tốc độ của boss |
| 2 | attackRange | float | tầm đánh của boss |
| 3 | player | Transform | xác định player |
| 4 | rb | Rigidbody2D | lấy Rigidbody2D của boss |
| 5 | boss | Boss | chứa reference đến class boss |
| 6 | OnStateEnter(Animator, AnimatorStateInfo, int) | void | Xử lý lấy các giá trị ban đầu khi vào animation Boss\_Run |
| 7 | OnStateUpdate(Animator, AnimatorStateInfo, int) | void | Xử lý boss thực hiện đuổi theo player và tấn công khi trong tầm đánh |
| 8 | OnStateExit(Animator, AnimatorStateInfo, int) | void | reset lại trigger tấn công cho lần đánh kế |

* 1. **BossController:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | State | enum | tương tự class BasicEnemyController |
| 2 | currentState | State | tương tự class BasicEnemyController |
| 3 | movementSpeed | float | tương tự class BasicEnemyController |
| 4 | maxHealth | float | tương tự class BasicEnemyController |
| 5 | knockbackDuration | float | tương tự class BasicEnemyController |
| 6 | lastTouchDamageTime | float | tương tự class BasicEnemyController |
| 7 | touchDamageCooldown | float | tương tự class BasicEnemyController |
| 8 | touchDamage | float | tương tự class BasicEnemyController |
| 9 | touchDamageWidth | float | tương tự class BasicEnemyController |
| 10 | touchDamageHeight | float | tương tự class BasicEnemyController |
| 11 | touchDamageCheck | Transform | tương tự class BasicEnemyController |
| 12 | whatIsPlayer | LayerMask | tương tự class BasicEnemyController |
| 13 | knockbackSpeed | Vector2 | tương tự class BasicEnemyController |
| 14 | touchDamageBotLeft | Vector2 | tương tự class BasicEnemyController |
| 15 | touchDamageTopRight | Vector2 | tương tự class BasicEnemyController |
| 16 | currentHealth | float | tương tự class BasicEnemyController |
| 17 | knockbackStartTime | float | tương tự class BasicEnemyController |
| 18 | attackDetails | float[] | tương tự class BasicEnemyController |
| 19 | facingDirection | int | tương tự class BasicEnemyController |
| 20 | damageDirection | int | tương tự class BasicEnemyController |
| 21 | movement | Vector2 | tương tự class BasicEnemyController |
| 22 | this | GameObject | tương tự class BasicEnemyController |
| 23 | aliveRb |  | tương tự class BasicEnemyController |
| 24 | aliveAnim | GameObject | tương tự class BasicEnemyController |
| 25 | gm | GameObject | tương tự class BasicEnemyController |
| 26 | point | int | tương tự class BasicEnemyController |
| 27 | Start() | void | tương tự class BasicEnemyController |
| 28 | Update() | void | tương tự class BasicEnemyController |
| 29 | EnterMovingState() | void | tương tự class BasicEnemyController |
| 30 | UpdateMovingState() | void | tương tự class BasicEnemyController |
| 31 | ExitMovingState() | void | tương tự class BasicEnemyController |
| 32 | EnterKnockbackState() | void | tương tự class BasicEnemyController |
| 33 | UpdateKnockbackState() | void | tương tự class BasicEnemyController |
| 34 | ExitKnockbackState() | void | tương tự class BasicEnemyController |
| 35 | EnterDeadState() | void | tương tự class BasicEnemyController |
| 36 | UpdateDeadState() | void | tương tự class BasicEnemyController |
| 37 | ExitDeadState() | void | tương tự class BasicEnemyController |
| 38 | DestroyEnemy() | void | tương tự class BasicEnemyController |
| 39 | Damage(float[]) | void | tương tự class BasicEnemyController |
| 40 | CheckTouchDamage() | void | tương tự class BasicEnemyController |
| 41 | Flip() | void | tương tự class BasicEnemyController |
| 42 | SwitchState(State) | void | tương tự class BasicEnemyController |
| 43 | OnDrawGizmos() | void | tương tự class BasicEnemyController |

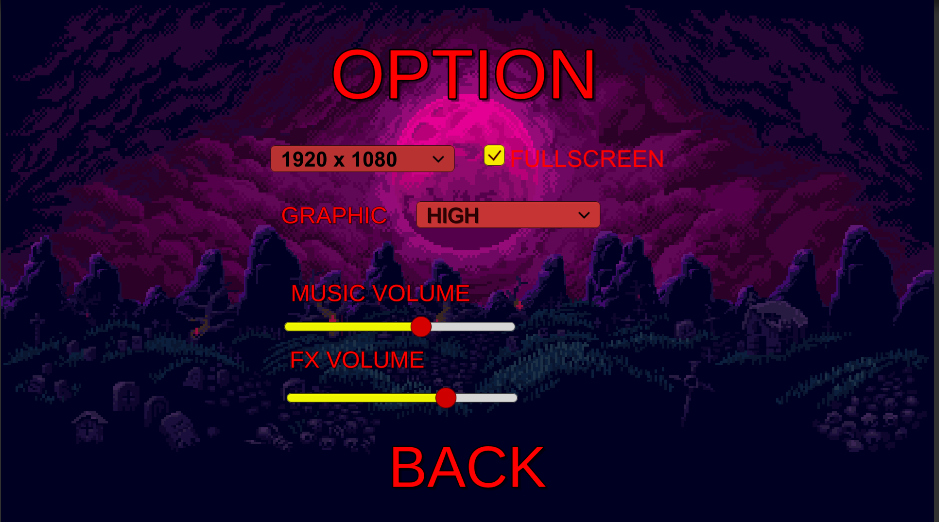
* 1. **BossWeopon:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên hàm/Tên biến** | **Kiểu dữ liệu / Kiểu trả về** | **Chức năng** |
| 1 | attackDamage | int | damage đòn tấn công của boss |
| 2 | enragedAttackDamage | int | không sử dụng |
| 3 | attackOffset | Vector3 | offset của đòn tấn công của boss |
| 4 | attackRange | float | Tầm đánh |
| 5 | attackMask | LayerMask | Xác định object có thể tấn công (Ở đây là player) |
| 6 | Attack() | void | Xử lý boss tấn tông |
| 7 | EnragedAttack() | void | không sử dụng |
| 8 | OnDrawGizmosSelected() | void | Vẽ ra các object để test |

1. **Phân tích các tính năng trong game:**
   1. **Menu:**
      1. **Start Menu:**



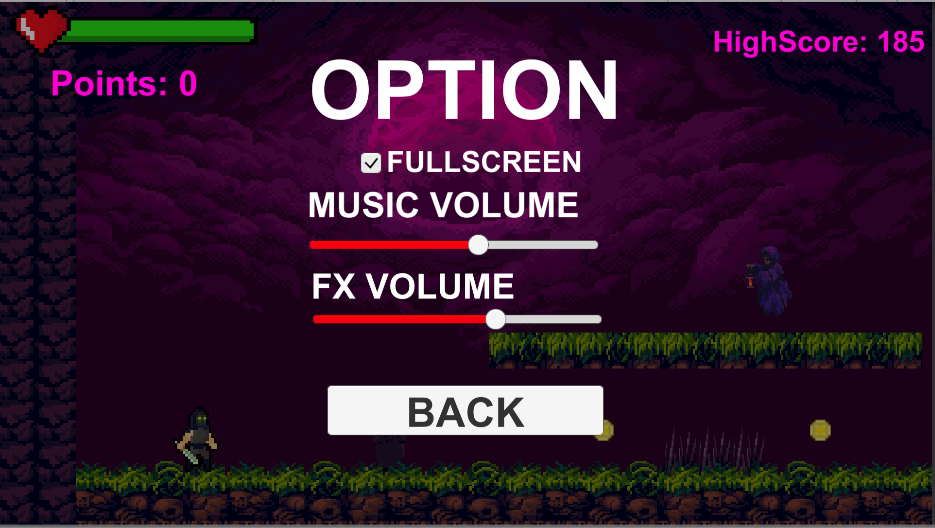
* Đây là Scene đầu tiên khi bạn vào trò chơi, trong scene gồm 3 nút là **PLAY**, **OPTION** và **QUIT**. Với nút **PLAY** bạn sẽ trực tiếp vào màn 1 của game và với nút **QUIT** thì bạn sẽ thoát khỏi trò chơi. Còn với nút **OPTION** thì bạn sẽ làm chuyển qua một Menu con là **Option Menu**.



* Trong phần Option Menu, gồm có:
* Một Dropbox chứa độ phân giải màn hình: bằng cách chọn một độ phân giải tương ứng thì trò chơi sẽ được set theo resolution đó. Trong drop box chứa các resolutions từ 320 x 200 đến 1920 x 1080.
* Một Dropbox chứa Graphic Quality: bằng cách chọn một Quality tương ứng thì trò chơi sẽ được set theo quality đó. Trong drop box chứa các Quality gồm: **VERY LOW**, **LOW**, **MEDIUM**, **HIGH**, **VERY HIGH**, **ULTRA**.
* Một check box với chức năng bạn có muốn trò chơi theo dạng toàn màn hình hay dạng cửa sổ.
* Hai thanh slider điều khiển âm thanh trong trò chơi. Một thanh điều khiển âm thanh cho nhạc nền, một thanh điều khiển âm thanh cho hiệu ứng.
* Và cuối cùng là nút **BACK** dùng để trở về **START MENU**.
  + 1. **Paused Menu:**



* Trong màn chơi, nếu bạn nhấn nút ESC thì trò chơi sẽ vào trạng thái Paused và **PAUSED MENU** sẽ xuất hiện với 5 nút sau: **RESUME**, **RESTART**, **OPTION**, **MAIN MENU**, **QUIT**. Nút **RESUME** thì bạn sẽ giải trừ trạng thái Paused, nhấn nút **RESTART** thì bạn sẽ chơi lại màn hiện tại đang chơi, nút **MAIN MENU** sẽ chuyển bạn về **START MENU**, nút **QUIT** sẽ giúp bạn thoát khỏi trò chơi. Và nút **OPTION** sẽ chuyển qua Menu con là **Option Menu** (của **Paused Menu**).



* Giống với Option Menu của START MENU nhưng chỉ có 1 check box kiểm tra toàn màn hình và 2 thanh slider điều khiển âm thanh cho nhạc nền và hiệu ứng.
  1. **Gameplay:**
     1. **Animation cho player và điều khiển chuyển đổi các trạng thái:**
* Animation:
* Đứng yên:

**Idle**

* Chạy:

****

* Tấn công:

****

* Trèo tường:



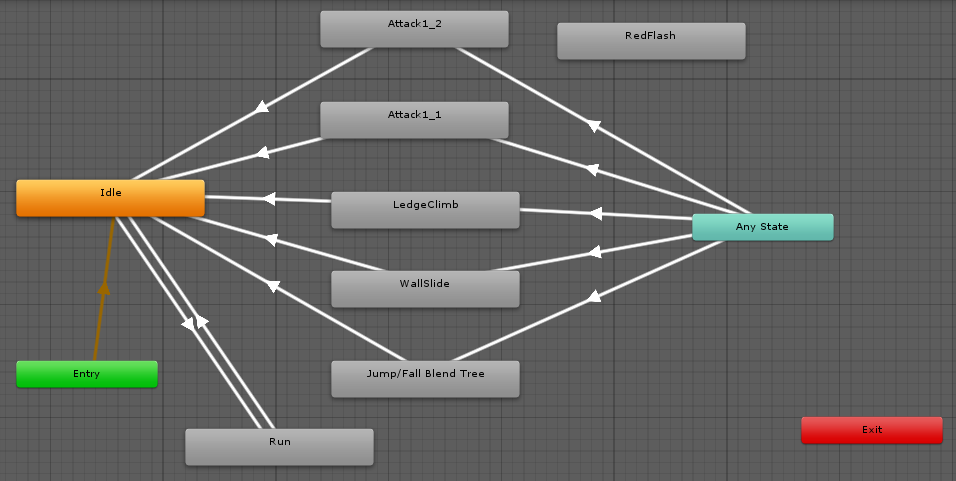
* Trượt tường:

Wall Slide

* Nhảy/Rơi:

Jump

* Bị thương: không có sprites, khi bị thương sẽ chớp đỏ.
* Animator: Bắt đầu với trạng thái Idle, sau đó tùy theo người điều khiển mà có các trạng thái khác nhau như: Run, WallSlide, Attack, LedgeClimb, RedFlash, Jump/Fall.



* + 1. **Animation cho enemies:**
* Animation
* Chạy:

ghost-3ghost-2ghost-1ghost-4

ghost-halo-3ghost-halo-2ghost-halo-1ghost-halo-4

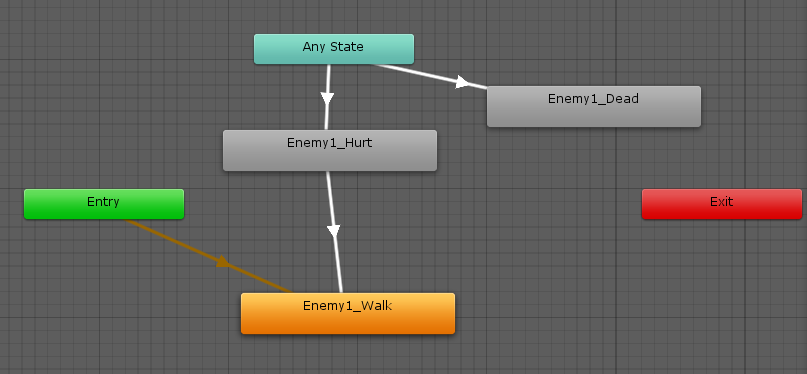
hell-gato-3hell-gato-2hell-gato-1hell-gato-4

skeleton-7skeleton-6skeleton-5skeleton-4skeleton-3skeleton-2skeleton-1skeleton-8

* Bị thương: Không có sprites, khi bị thương sẽ chớp màu khác.
* Chết:

enemy-death-4enemy-death-3enemy-death-2enemy-death-1enemy-death-5

* Animator chung: lúc bắt đầu là sẽ ở trạng thái Walk, sau đó tùy vào khi bị player tấn công sẽ xuất hiện 1 trong 2 trạng thái sau là: Hurt và Dead.

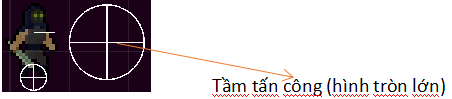


* + 1. **Điều khiển player:** Đây là một tính năng mà mọi trò chơi đi cảnh đánh quái đều có, đó là điều khiển nhân vật. Người dùng có thể điều khiển nhân vật thông qua các nút trên bàn phím hoặc con chuột.
* Tấn công: khiến player tấn công + thực hiện animation tấn công.
* Nhảy: khiến player nhảy + thực hiện animation nhảy.
* Chạy: khiến player chạy + thực hiện aniamtion chạy.
* Lướt: Lướt tới phía trước mặt một khoảng cách. Khi đang lướt sẽ để lại tàn ảnh. Lướt có thể được sử dụng khi đang chạy, đang tấn công và khi nhảy. Lướt có thời gian hồi 2.5s.



Khi player lướt

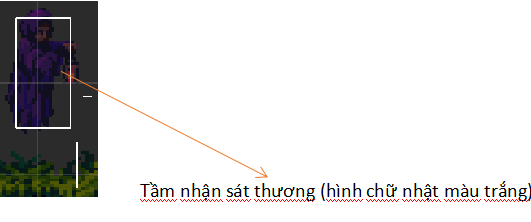
* + 1. **Player tương tác với enemies:** player tương tác với enemies qua việc tấn công, bị va chạm và khi lướt qua:
* Khi tấn công enemies: Nếu các vật thể bị player nhận định là enemies ngay trong tầm tấn công của player thì enemies sẽ bị nhận sát thương và tùy vào tình trạng thì enemies sẽ bị Hurt hoặc Dead.





Player tấn công enemies

* Khi enemies va chạm: Nếu các vật thể bị enemies nhận định là player vào tầm nhận sát thương của enemies thì player sẽ bị đẩy lùi và tùy vào tình trạng sẽ khiến player bị thương hoặc chết.





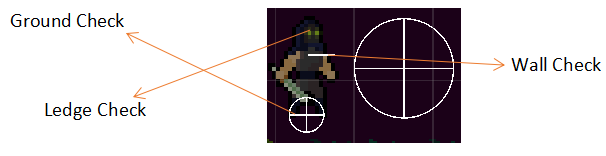
Player bị Enemies va chạm

* Khi lướt qua enemies: player sẽ không nhận sát thương từ enemies và enemies sẽ không nhận sát thương từ player.



Player lướt qua enemies

* + 1. **Player tương tác với game object:**
* Mặt đất (ground): có rất nhiều phương thúc khác nhau để player tương tác với ground:

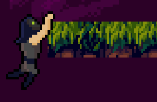


* Ground Check: là object con nằm trong Player. Nhiệm vụ của Ground Check giúp cho Player nhận biết được mình có đang đứng trên mặt đất hay không. Nếu các vật thể bị nhận định là Ground mà Player đang va chạm với nó trên bề mặt thì các trạng thái Idle, Run, Jump có thể được thực hiện.
* Wall Check: Khi player đang trong trạng thái Jump/Fall để tới mặt đất cao hơn hoặc thấp hơn mà player lại không thể chạm tới bề mặt chiều dài của Ground, lại chỉ va chạm với chiều cao bề mặt của Ground. Khi việc đó xảy ra, các vật thể bị nhận định là Ground mà Player đang va chạm với nó thì việc Wall Slide (trượt tường) sẽ được thực hiện.



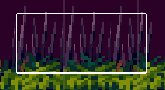
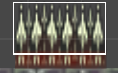
Player trượt tường

* Ledge Check: Khi player đang trong trạng thái Jump/Fall để tới mặt đất cao hơn hoặc thấp hơn mà player lại không thể chạm tới bề mặt chiều dài của ground, lại chỉ va chạm với góc cạnh của ground. Khi việc đó xảy ra, các vật thể bị nhận định là có thể leo được mà Player đang va chạm với nó thì việc Ledge Climb (leo tường) sẽ được thực hiện.



Player leo tường

* Bẫy (trap):

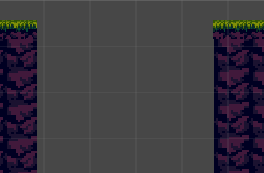


Tầm nhận sát thương (đường chữ nhật màu trắng)

* Gai (Spike): là một game object thường thấy xuyên suốt màn chơi. Khi các vật thể bị nhận định là player va chạm với tầm nhận sát thương của Spike thì tùy theo lượng máu mà player có thể bị chết hoặc nhận sát thương.



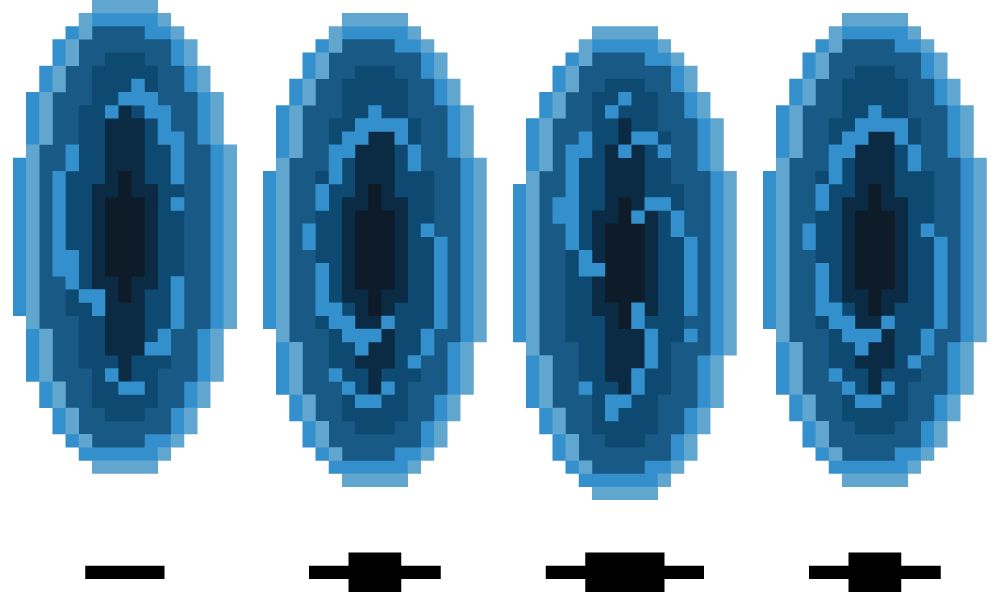
* Tường ép: là một bẫy chết người. Khi các vật thể bị nhận định là player va chạm với tầm nhận sát thương của tường ép thì player sẽ chết ngay lập tức.



* Vực (World End): Khi các vật thể bị nhận định là player rơi xuống vực thì player sẽ chết ngay lập tức.
* Đồng tiền (coin): là một game object thường thấy xuyên suốt màn chơi. Khi các vật thể bị nhân định là player va chạm với nó thì player sẽ “ăn” nó để tăng điểm.

Coin

* Cổng (portal): là một game object dùng để chuyển tiếp màn chơi. Các vật thể bị nhận định là player khi va chạm với portal sẽ hiện lên dòng chữ hướng dẫn chuyển tiếp màn chơi.





Player va chạm với Portal. Nhấn E để chuyển sang màn tiếp theo

* Đá hướng dẫn: các vật thể bị nhận định là player khi va chạm với tảng đá này thì sẽ hiện lên dòng hướng dẫn chơi game.





Player va chạm với đá hướng dẫn, từ đó dòng hướng dẫn chơi game hiện lên

* Bậc thềm platform:

* Moving platform: là bậc thềm có thể di chuyển qua lại, trên xuống. Khi các bậc thềm này đến điểm chỉ định hoặc va chạm với những thứ nhận định là vật cản thì chúng sẽ di chuyển theo chiều hướng ngược lại và cứ lặp lại như vậy. Player tương tác với Moving platform qua việc chuyển động của player không bị cô lập khi đứng trên bậc thềm.



Chuyển động của khi player không bị cô lập khi đứng trên Moving platform

* Falling platform: là bậc thềm ngừng giữa không trung. Khi có các vật thể bị nhận định là player va chạm với nó thì sau số thời gian delay, bậc thềm sẽ rơi tự do kèm theo player. Bậc thềm này thường được đặt ở chỗ có vực sâu hoặc có gai để tăng tính hiệu quả cho bẫy.



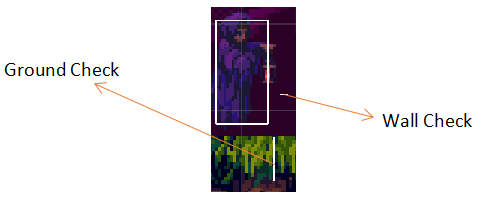
Player đứng trên bậc thềm và sau đó rơi cùng bậc thềm

* Up Platform: là bậc thềm có thể cho player nhảy lên qua bậc thềm đó.



Up platform

* + 1. **Enemies tương tác với game object:**



* Khi sắp va chạm tường (Wall Check): khi enemies di chuyển và tầm Wall Check va chạm với các vật thể bị nhận định là Ground thì enemies sẽ quay đầu và di chuyển theo chiều ngược lại.

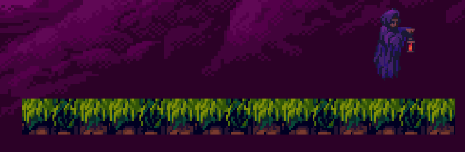


Khi enemies đụng phải vật cản,

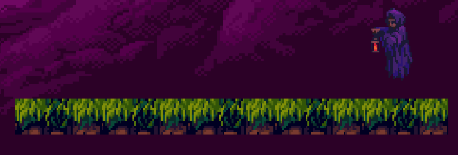


enemies quay đầu và di chuyển theo hướng ngược lại

* Khi sắp đi hết bề mặt đất (Ground Check): khi enemies di chuyển và tầm Ground Check sắp va chạm với khoảng không thì enemies sẽ quay đầu và di chuyển theo chiều ngược lại.

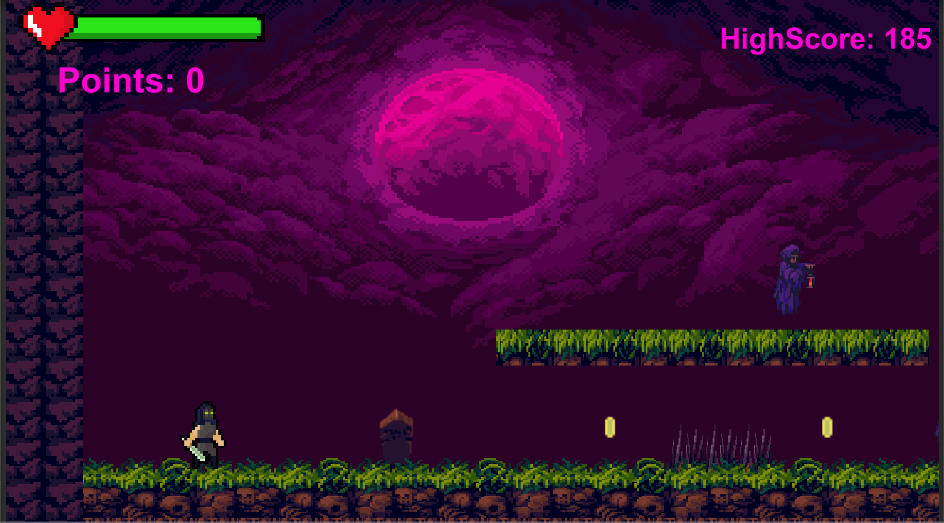


Khi enemies đi tới cuối đường,



enemies quay đầu và di chuyển theo hướng ngược lại

* + 1. **Hiện Score, High Score, Health Bar cho Player:**



* Score: Đây là điểm số của player. Khi player ăn được đồng tiền hay giết chết enemies thì điểm số sẽ tăng lên. Bằng việc sử dụng PlayerPrefs (một hàm lưu trữ thông tin trong game để sử dụng cho các màn chơi) thì điểm số của bạn sẽ được lưu trữ để sử dụng tiếp cho các màn tiếp theo.
* High Score: Đây là điểm cao nhất mà player đạt được. Khi player chết đi hoặc thành công qua màn cuối, điểm số của player sẽ biến thành High Score. Trong trường hợp điểm số của player thấp hơn High Score thì High Score vẫn giữ nguyên.
* Health Bar: Là thanh máu của player. Khi player nhận sát thương, thanh máu sẽ giảm xuống và khi giảm tới một mức nhất định thì thanh máu sẽ chuyển màu. Khi hết thanh máu, player sẽ chết và màn chơi sẽ được restart lại.



Thanh máu khi đầy (màu xanh)



Khi còn 70% máu (màu vàng)



Khi còn 30% máu (màu đỏ)

* + 1. **Môi trường, âm thanh, hiệu ứng trong trò chơi:**
* Môi trường: Dựa theo môi trường thời trung cổ, đầy vẻ ma quái và tăm tối, với kẻ thù thây ma, xương và chó địa ngục, hoàn cảnh nghĩa địa và lâu đài.
* Âm thanh: Nhạc nền hành động, âm thanh của tiếng kiếm xoẹt qua, tiếng đồng tiền “keng keng” khi player ăn đồng tiền.
* Hiệu ứng: độc đáo, cổ điển.
  + 1. **Boss:**
* Animation:
* Idle:

**idle3idle2idle1idle4**

* Attack:

**punch5punch4punch3punch2punch1punch6**

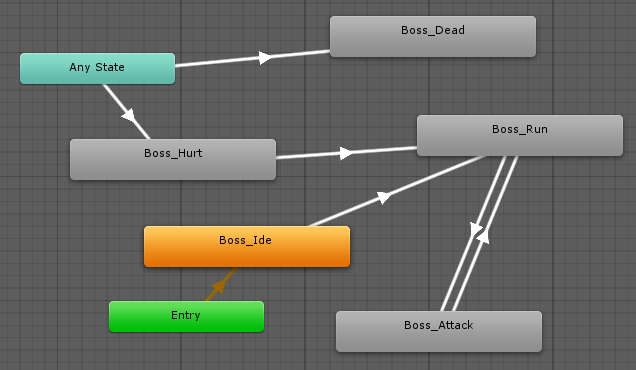
* Run:

**walk5walk4walk3walk2walk1walk6**

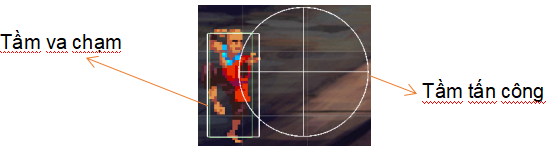
* Hurt: không có sprites.
* Death:



* Animator: Bắt đầu với trạng thái Idle, sau đó tùy theo tương tác với player mà có các trạng thái khác nhau như: Run, WallSlide, Attack, Hurt, Dead.



* Tương tác với player:



* Luôn đuổi theo player: Boss luôn luôn đuổi theo các vật thể bị nhận định là player.
* Khi tấn công player: Nếu các vật thể bị boss nhận định là player vào tầm tấn công của boss thì boss sẽ chuyển sang trạng thái tấn công và thực hiện animation Attack. Nếu player vẫn trong tầm tấn công khi boss đang thực hiện animation Attack thì tùy theo tình trạng mà dẫn đến 2 trường hợp là player bị nhận sát thương hoặc chết.
* Khi va chạm với player: Nếu các vật thể bị boss nhận định là player vào tầm va chạm của boss thì player sẽ bị đẩy lùi và tùy theo tình trạng mà dẫn đến 2 trường hợp là player bị nhận sát thương hoặc chết.
* Khi bị player tấn công: Các vật thể bị player nhận định là enemies (boss là enemies) khi bị player tấn công thì boss sẽ chuyển sang trạng thái Hurt hoặc Dead.

1. **KẾT QUẢ:**

* Hiểu thêm được các ứng dụng làm game Unity, các phần mềm hỗ trợ để lập trình một game hoàn chỉnh.
* Hiểu sâu hơn về ngôn ngữ C# trên nền tảng Unity engine trong công nghệ làm game.
* Làm được một game platform 2D đi cảnh đánh quái trên PC với những chức năng cơ bản.
* Tuy nhiên, vẫn còn tồn tại những hạn chế.

1. **HƯỚNG PHÁT TRIỂN:**

* Phát triển thêm về boss, tạo được bot mạnh hơn cho boss.
* Phát triển thêm về enemies, tạo được bot cho enemies cũng như nhiều loại enemies khác nhau như: enemies biết bắn, biết bay,…
* Phát triển thêm về môi trường, âm thanh, hiệu ứng khiến trò chơi đặc sắc hơn.
* Phát triển thêm về bẫy, tạo được nhiều loại bẫy đa dạng hơn.
* Nhiều cấp độ chơi hơn.
* Có thể cho phep chọn độ khó.

1. **TÀI LIỆU THAM KHẢO:**

* Unity và C#:

<https://edu.toidayhoc.com/lop-hoc/hoc-lap-trinh-game-huong-dan-lam-game-2d-tren-unity/>

<https://docs.unity3d.com/ScriptReference/>

* Assets:

<https://assetstore.unity.com/packages/2d/characters/gothicvania-town-101407?fbclid=IwAR12JMuNOxOYh_BFQnX1n0J1cKqWkqxh8n3AifolKOa-TUlRjQ3BeSgDlKg>

<https://assetstore.unity.com/packages/2d/characters/gothicvania-cemetery-120509?fbclid=IwAR18QrSdofSTOJAT7NRlXm59qR-xQpPZ0JgzuWHKJqJ-2FZSLPqmMRcBtAo>

<https://assetstore.unity.com/packages/2d/characters/gothicvania-swamp-152865?fbclid=IwAR2gj79ZjZ6RTOWoG8cC-fBnVpfpuFTiP2q8XUfMarT3lNSkiLVs5_pzFYg>

<https://assetstore.unity.com/packages/2d/characters/gothicvania-church-pack-147117?fbclid=IwAR0vCS0PSHqYkuKy-fFni8tE-CqN0h28-MDlAc_i32Ukn8u2qiNXNWorfq4>

**HẾT**