

HỌC VIỆN CÔNG NGHỆ THÔNG TIN & THIẾT KẾ VTC ACADEMY



2D GAME ENGINE

ASHEN'S BLADE

Ngành : CÔNG NGHỆ THÔNG TIN

Chuyên ngành : LẬP TRÌNH GAME

Giảng viên hướng dẫn : Thầy Nguyễn Trung Hiếu

Học viên thực hiện : Tạ Đức Thành

MSHV: 124010124035 Lớp: K24GD-02

TP. Hồ Chí Minh, 2025

HỌC VIỆN CÔNG NGHỆ THÔNG TIN & THIẾT KẾ VTC ACADEMY



2D GAME ENGINE

ASHEN'S BLADE

Ngành : **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành : **LẬP TRÌNH GAME**

Giảng viên hướng dẫn : Thầy Nguyễn Trung Hiếu

Học viên thực hiện : Tạ Đức Thành

MSHV: 124010124035 Lớp: K24GD-02

TP. Hồ Chí Minh, 2025

1. Giới thiệu về game



Tên game: Ashen Blade

Thể loại: Platformer hành động 2D

Mô tả:

Shadow Quest là game platformer 2D, nơi người chơi điều khiển nhân vật là một samurai nữ tên “Ashen” vượt qua các màn chơi đầy thử thách. Người chơi phải tránh chướng ngại vật, tiêu diệt enemy thường và đánh bại boss ở màn cuối để chiến thắng binh đoàn samurai tàn ác. Mỗi màn chơi tăng dần độ khó: enemy di chuyển nhanh hơn, xuất hiện nhiều hơn và boss sẽ chỉ xuất hiện ở màn chơi cuối.

Mục tiêu của người chơi:

- Di chuyển qua các màn chơi.
- Tấn công và tiêu diệt enemy và boss.
- Thay vì tính điểm người chơi sẽ phải bảo toàn lượng máu trong các màn chơi để sống sót tới chiến thắng
- Hoàn thành các màn chơi mà không bị chết hết máu.

2. Gameplay & cách tính điểm

Player:



- Có thể di chuyển sang trái/phải, nhảy, và tấn công bằng kiếm hoặc ném phi tiêu.
- Có thanh HP hiển thị trên HUD.
- Mỗi lần nhận sát thương từ enemy sẽ mất HP.

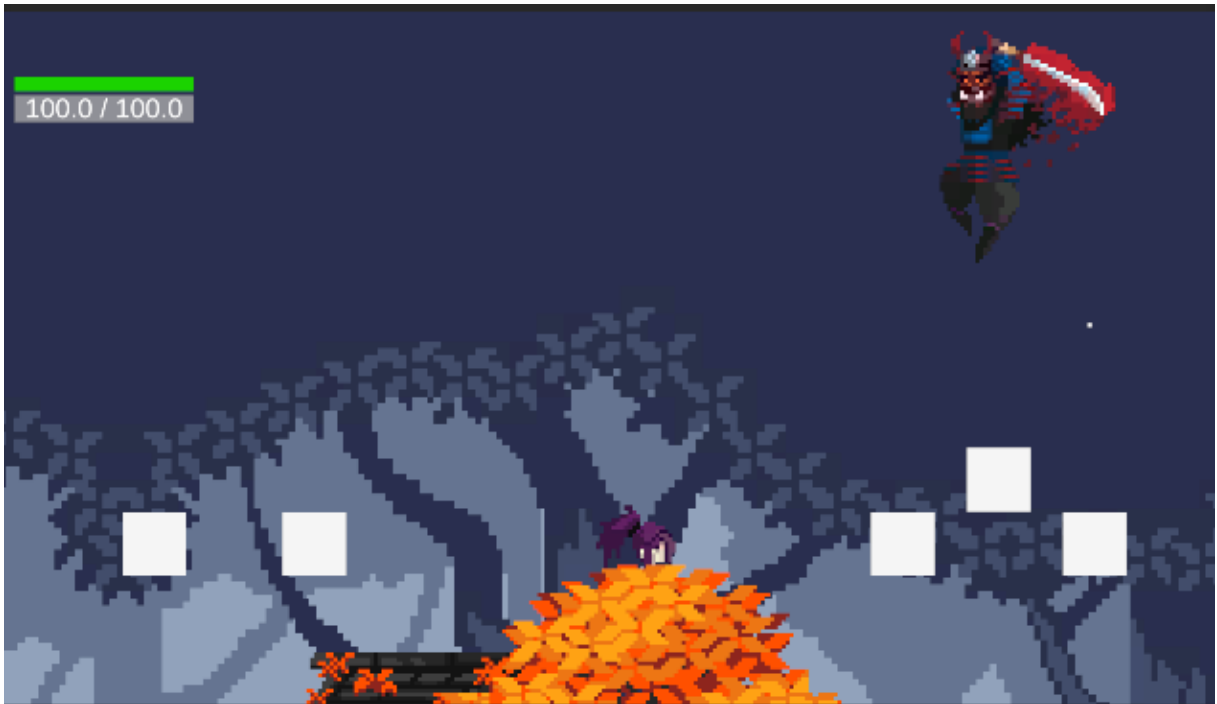
Enemy thường:



- Di chuyển tuần tra qua lại chỉ khi gặp tường hoặc không có đường đi mới đổi hướng sau khi nhận damage sẽ di chuyển theo player.
- Gây sát thương player bằng các đòn đánh.

- Enemy sẽ có chiêu thức dịch chuyển ra đằng sau player sau đó tấn công khi trong phạm vi và có thời gian hồi chiêu.

Boss:



- Có ít nhất 2 chiêu thức khác nhau (tấn công gần và tấn công khi nhảy).
- Nhiều HP hơn enemy thường.

Level progression:

- **Màn 1:** Enemy thường số lượng ít, các chỉ số trong enemy data sẽ thấp hơn, không có boss.
- **Màn 2:** Enemy thường, số lượng nhiều hơn, các chỉ số sẽ cao hơn và có boss mạnh hơn enemy thường có thể tấn công khi enemy đứng trên cao.

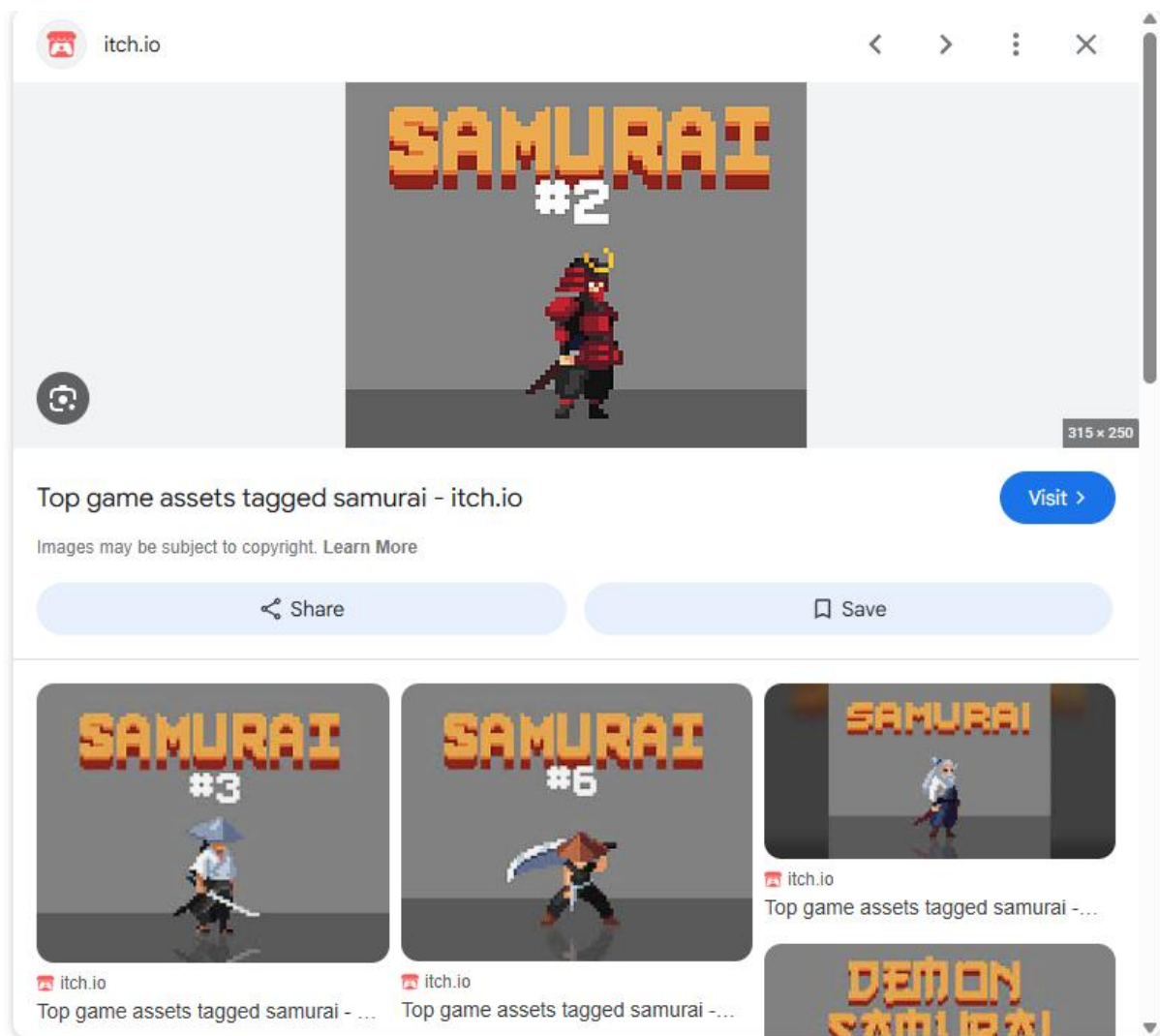
Vật phẩm trong game:



- Người chơi sẽ phải ăn những nguồn thức ăn được rải rác trong game để sống sót qua các màn chơi.

3. Assets sử dụng

Art:

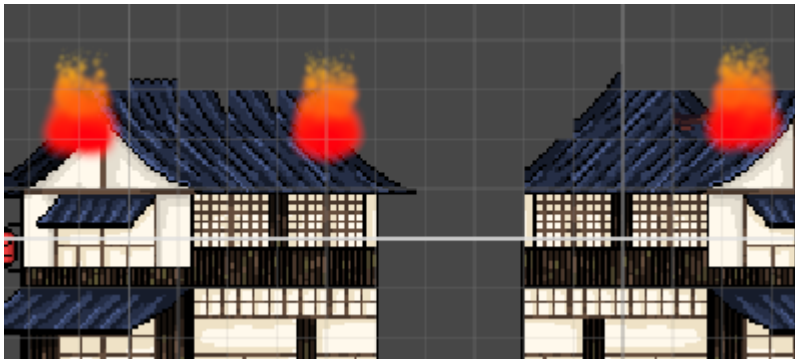


- Player sprite & animation: *Samurai #4 Asset* – itch.io
- Enemy sprite & animation: *Samurai #5 & Samurai #2 Asset* – itch.io
- Boss sprite & animation: *Demon Samurai* – itch.io free assets
- Background sprite: 3 background layers – itch.io free assets
- Tilemap: *Trees, Props, Grounds Tileset* – itch.io free assets

Audio:

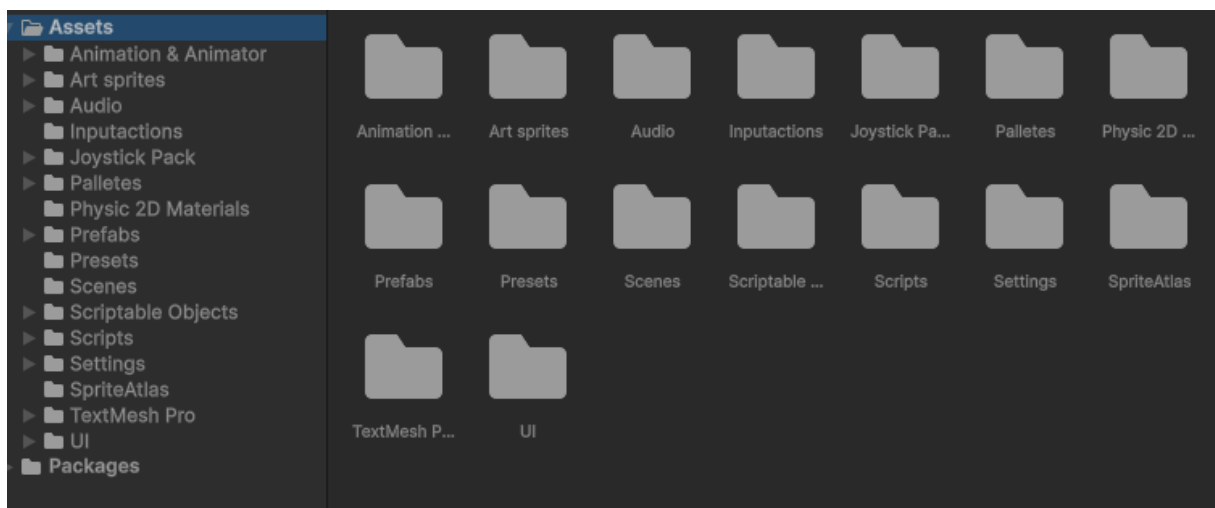
- Nhạc nền: *Ambient Samurai OST* – Youtube
- SFX tấn công player/enemy: *Sword Swing, Hit Sound* – SFX Online
- SFX boss idle talking: *Aatrox lines* – Youtube

VFX:



- Fire: Particle Effect Unity

4. Cấu trúc project Unity



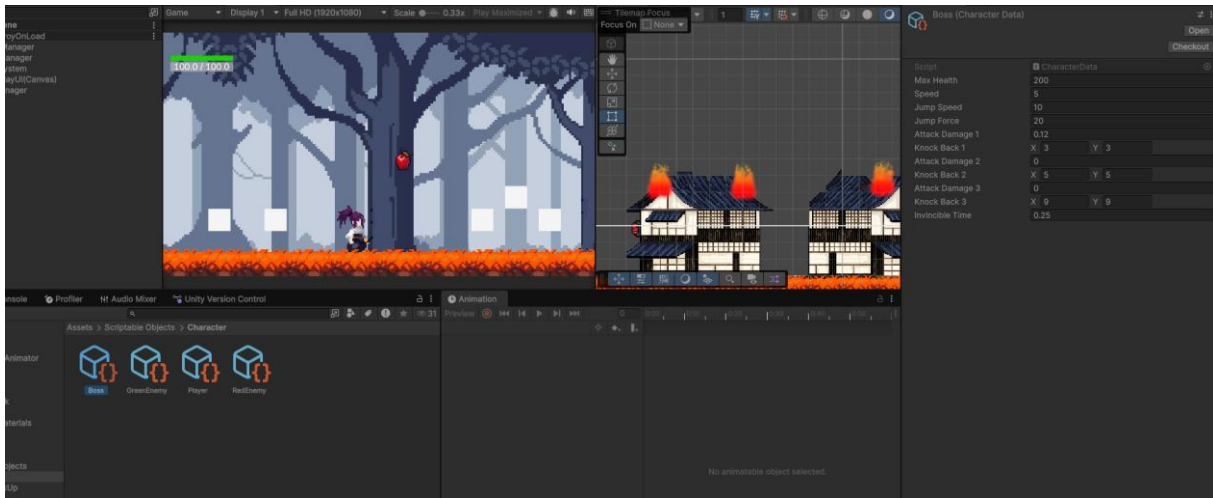
Assets/

- ├─ Art/ (sprite, animation)
- ├─ Audio/ (music, sfx)
- ├─ Prefabs/ (player, enemy, boss, items)
- ├─ Scenes/ (MainMenu.unity, Level1.unity, Level2.unity, Victory.unity)
- ├─ Scripts/ (PlayerController.cs, EnemyController.cs, GameManager.cs, UIManager.cs)
- └─ UI/ (HUD.prefab, Menu.prefab, PauseMenu.prefab, GameOver.prefab)

Quy tắc:

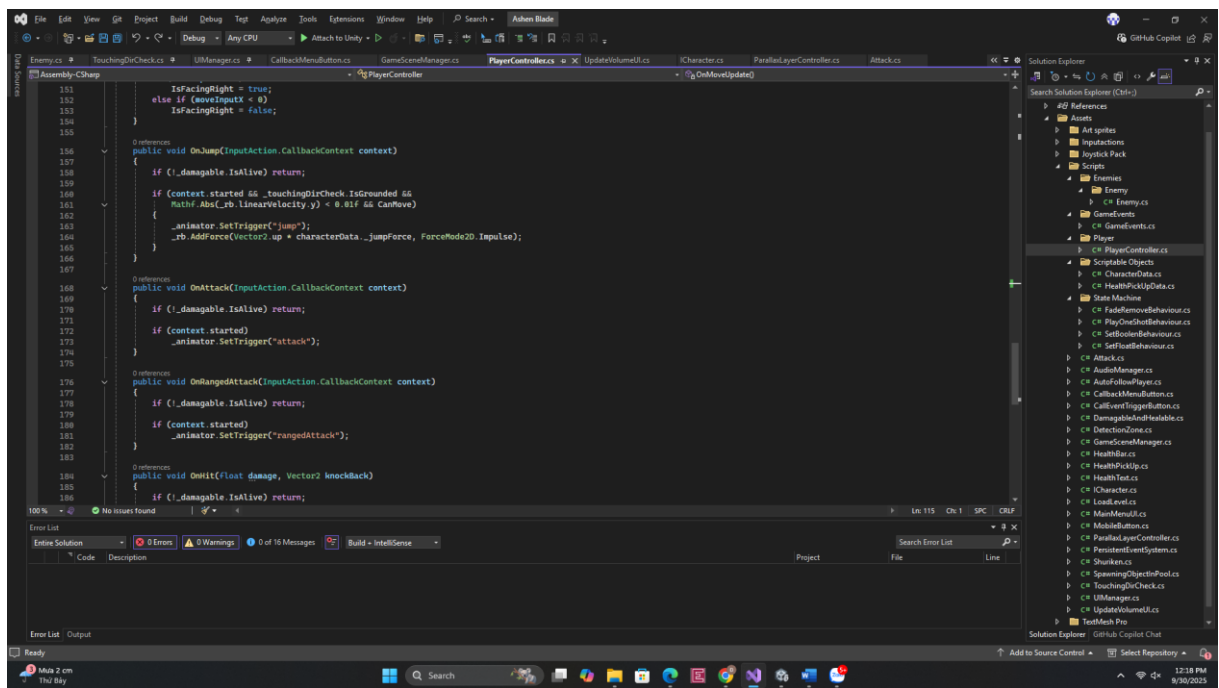
- Scripts tách riêng theo class.
- Prefab đặt trong Prefabs, không đặt trong Scenes.
- Asset sắp xếp rõ ràng theo loại (Art, Audio...).

5. Kỹ thuật Unity sử dụng



- **Rigidbody2D + Collider2D:** xử lý va chạm player-enemy, player-map, enemy-map.
- **Animator:**
 - Player: idle, move, attack, death
 - Enemy: idle, move, attack, death
 - Boss: idle, move, attack1, attack2, death
- **Âm thanh:** nhạc nền + SFX (attack, hit, skill)
- **Hiệu ứng hình ảnh:** Unity particle effects
- **Tilemap/Prefab:** xây dựng map và các đối tượng dễ dàng tái sử dụng.
- **Scriptable Objects:** để lưu chỉ số cho item và stats mặc định của nhân v
- **Prefab Pools:** để tái sử dụng lại projectile (Shuriken) phóng ra từ Player

6. Code sạch sẽ (Clean Code)



- Biến và hàm đặt tên rõ ràng, ví dụ `PlayerController.Move()`, `EnemyController.TakeDamage()`.
- Scripts chia thành nhiều file: `PlayerController`, `EnemyController`, `BossController`, `GameManager`, `UIManager`.
- Không viết logic toàn bộ trong 1 file duy nhất.
- Các hàm nhỏ, dễ bảo trì.