

Nghiên cứu phát triển game 3D với công nghệ Unity Using the Unity Game Engine to Develop 3D Games: A Case Study

Chu Thị Mai Hương¹, Văn Thị Thiên Trang²

^{1,2} Khoa Công nghệ thông tin, Trường Đại học Kinh tế - Tài chính Thành phố Hồ Chí Minh – UEF
¹huongctm19@uef.edu.vn, ²trangvtt@uef.edu.vn

Tóm tắt: Bài báo này chủ yếu tập trung vào các kiến thức liên quan đến Unity Engine, và quá trình phát triển game nhập vai hành động có tên là Sunny Kingdom. Trong game, người chơi sẽ hoá thân thành một cô gái tên Jane, tham gia vào các cuộc phiêu lưu, chiến đấu với kẻ thù, thu thập trang bị và nâng cấp nhân vật.

Từ khóa: Unity, Game Engine, Lập trình game, Ngôn ngữ lập trình C#, Công nghệ thông tin

Abstract: This paper primarily focuses on knowledge related to Unity Engine and the development process of an action role play game called Sunny Kingdom. In the game, players will embody a character named Jane, participating in adventures, battling enemies, collecting equipment, and upgrading their character.

Keywords: Unity, Game Engine, Game Development, C# Programming Language, Information Technology

1. Giới thiệu

Phát triển game luôn là một chủ đề gây tranh cãi trong lĩnh vực học thuật. Trên toàn thế giới, hiện vẫn chưa có nhiều trường học tập trung vào đào tạo về phát triển game. Dù vậy, ngành công nghiệp game vẫn phát triển một cách đáng kể, và đạt doanh thu hàng tỷ đô la mỗi năm [1]. Ngành công nghiệp game cũng là một trong những động lực chính đằng sau sự tiến bộ của phần cứng hiện đại như bộ xử lý đa nhân (multi-core), bộ xử lý đa luồng (hyper-threading), đồ họa (GPU) hiệu năng cao, thiết bị xử lý âm thanh tiên tiến, và các thiết bị giao diện giữa con người và máy tính (human-computer interface devices) như kính thực tế ảo, v.v...

Đi cùng với sự phát triển vượt bậc của các thiết bị phần cứng kể trên, ngành công nghiệp game còn đóng góp vào việc sản xuất các bộ phim. Một số bộ phim nổi tiếng đã được xây dựng dựa trên các tựa game, ví dụ như Assassin's Creed [2]. Các bộ phim dựa trên game này thường tạo ra sự kết hợp giữa các yếu tố của game và điện ảnh, mang đến những trải nghiệm mới cho khán giả. Ngoài ra, chúng kiến sự xuất hiện của những game với đồ họa siêu thực như God of War, Ghost of Tsushima, The Last of Us, v.v..., khiến

người chơi khó phân biệt được sự khác biệt giữa thế giới thực và thế giới ảo. Điều này đặt ra câu hỏi về công cụ nào có thể hỗ trợ việc tạo ra những games tương tự như vậy?

Trong lĩnh vực này, Unity Engine đã chứng tỏ mình là một công cụ phát triển mạnh mẽ và linh hoạt. Unity được phát triển bởi Unity Technologies, một công ty được thành lập vào 2004 tại Đan Mạch, được sử dụng chủ yếu để tạo nội dung tương tác như game, mô phỏng kiến trúc, điện ảnh, v.v...

Unity cung cấp bốn gói dịch vụ. Gói miễn phí được gọi là Unity Personal và được sử dụng cho mục đích cá nhân và các dự án phi thương mại. Ba gói còn lại (Unity Plus, Unity Pro, Unity Enterprise) đều là các gói trả phí tùy thuộc vào quy mô của đội ngũ phát triển [3].

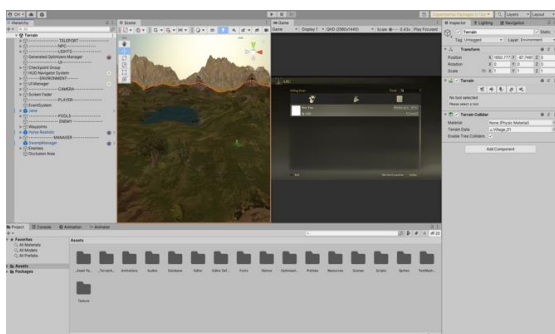
Ngoài ra, Unity còn hỗ trợ nhiều môi trường phát triển tích hợp (IDE) phổ biến như Visual Studio, Visual Studio Code, JetBrains Rider. Trước đây, bằng việc sử dụng Unity, có thể viết game bằng nhiều ngôn ngữ lập trình như C#, JavaScript và Boo. Tuy nhiên, hiện tại Unity chỉ tập trung chủ yếu vào hỗ trợ C# cho việc lập trình game. C# được coi là một ngôn ngữ mạnh

mẽ và hiệu quả, được phát triển bởi Microsoft cho framework .NET.

2. Unity Game Engine

Phần này cung cấp những đặc điểm chính của Unity mà chúng tôi coi là những yếu tố quan trọng đã đóng góp vào sự thành công của engine này, khiến nó trở thành một công cụ phát triển game tuyệt vời, và được ưa chuộng trong ngành công nghiệp game.

Dễ học và sử dụng. Unity được thiết kế để dễ học và sử dụng, đặc biệt đối với những người mới bắt đầu trong lĩnh vực phát triển game. Nó cung cấp một giao diện trực quan và có nhiều tài liệu hướng dẫn, video hướng dẫn và nguồn tài nguyên trực tuyến để hỗ trợ quá trình học.



Hình 1. Giao diện Unity Editor.

Cộng đồng phát triển mạnh mẽ. Unity có một cộng đồng phát triển đông đảo và tích cực (<https://forum.unity.com/>). Những người tham gia vào các dự án game có thể chia sẻ kinh nghiệm, tài liệu, cũng như nhận được sự hỗ trợ từ cộng đồng để giải quyết các vấn đề trong quá trình phát triển.

Đa nền tảng. Unity hỗ trợ phát triển game trên nhiều nền tảng, bao gồm PC, điện thoại di động, máy chơi game cầm tay, web và thực tế ảo/ảo hóa. Điều này cho phép các nhà phát triển game tiếp cận đối tượng người chơi đa dạng trên các thiết bị khác nhau, mang đến tiềm năng kinh doanh cao, giúp game đạt được sự phổ biến và tiếp cận người chơi trên một phạm vi rộng lớn.

Đồ họa và âm thanh. Với khả năng làm việc với đồ họa 2D và 3D, Unity cho phép tạo ra hiệu ứng hình ảnh và âm thanh đa dạng. Các tính năng này giúp tạo ra trải nghiệm hấp dẫn cho người chơi, từ hình ảnh sắc nét và đẹp mắt đến âm thanh sống động và phong phú. Bên cạnh đó, Unity cũng hỗ

trợ các hiệu ứng đặc biệt như ánh sáng, bóng đổ và các hiệu ứng đặc biệt khác để làm tăng tính chân thực và sáng tạo của game.

Công cụ phân tích và tối ưu hóa. Unity cung cấp các công cụ phân tích và tối ưu hóa để giúp các lập trình viên hiểu và cải thiện hiệu năng của game. Các công cụ này cho phép xem thống kê về tài nguyên sử dụng, thời gian xử lý, và khám phá các vấn đề tiềm ẩn. Điều này giúp tối ưu hóa game để đảm bảo chạy mượt mà và tiết kiệm tài nguyên.

Phần tiếp theo của bài báo tập trung vào việc giải thích cách Unity đã được áp dụng để triển khai các thành phần quan trọng trong game Sunny Kingdom. Các thành phần này bao gồm nhân vật người chơi (player), các nhân vật không phải người chơi (NPCs), vật phẩm, hiệu ứng, môi trường và giao diện người dùng. Bằng cách sử dụng công nghệ Unity, chúng tôi đã có thể tạo ra những thành phần này một cách linh hoạt và hiệu quả.

3. Phân tích thiết kế

Nội dung chủ yếu tập trung vào việc thực hiện các thiết kế cho game Sunny Kingdom, bao gồm biểu đồ luồng game, người chơi, nhân vật phi người chơi (NPCs), vật phẩm, hiệu ứng và môi trường cảnh quan.

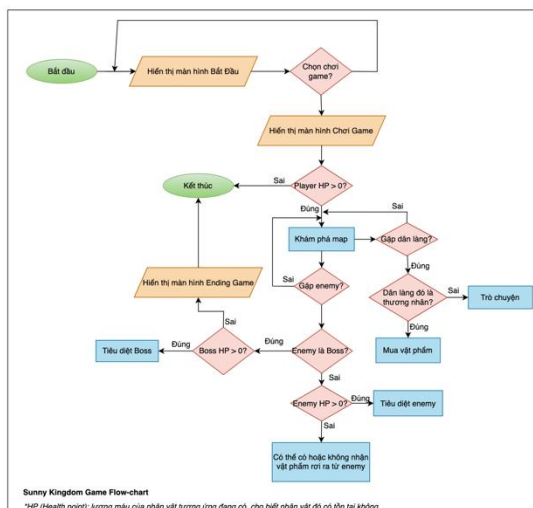
3.1. Game flow – chart

Trong game, người chơi sẽ bắt đầu tại một vị trí được chỉ định trên bản đồ. Họ có thể tự do di chuyển và khám phá các vùng đất chưa được tiết lộ. Trong quá trình khám phá, người chơi có thể tương tác với một số thành phần trong game.

Nếu người chơi gặp enemy, hệ thống sẽ xác định xem đó có phải là boss hay không. Nếu đó là boss, người chơi sẽ phải đánh bại boss đó để hoàn thành màn chơi và chuyển tới màn hình kết thúc. Nếu không phải boss, người chơi có thể tiêu diệt enemy đó để nhận được vật phẩm ngẫu nhiên.

Nếu người chơi gặp dân làng, hệ thống sẽ kiểm tra xem đó có phải là thương nhân hay không. Nếu đó là thương nhân, người chơi có thể mua các vật phẩm từ thương nhân đó. Trong trường hợp không phải thương nhân, người chơi có thể trò chuyện với dân làng, hoặc đơn thuần là chỉ quan sát, ngắm nhìn.

Trong suốt quá trình chơi, hệ thống sẽ theo dõi trạng thái HP của người chơi. Nếu HP giảm xuống 0, game sẽ kết thúc và hiển thị thông báo "You die", đồng nghĩa với việc thất bại trong game.



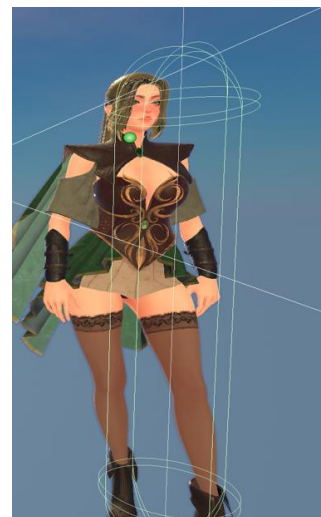
Hình 2. Sunny Kingdom game flow – chart.

3.2. Các nhân vật

3.2.1. Người chơi

Player là thuật ngữ được sử dụng để chỉ người chơi hoặc người điều khiển game. Player có thể là người thực, người chơi trực tiếp game bằng cách sử dụng bàn phím, chuột hoặc tay cầm. Player đóng vai trò trung tâm trong game, là người tạo ra sự tương tác và điều khiển các hoạt động trong game.

Trong game Sunny Kingdom, character controller được ứng dụng để quản lý tương tác giữa player và bàn phím. Nó nhận các tín hiệu đầu vào từ người chơi và sử dụng chúng để cập nhật vị trí và hướng di chuyển của player trong không gian game. Character controller cũng có trách nhiệm quản lý các thông số quan trọng như tốc độ di chuyển, độ cao nhảy, hướng nhìn và tương tác vật lý với môi trường xung quanh. Trong Hình 3, character controller được biểu thị bằng một hình trụ màu xanh nhạt bao quanh nhân vật player.



Hình 3. Character controller cho nhân vật player.

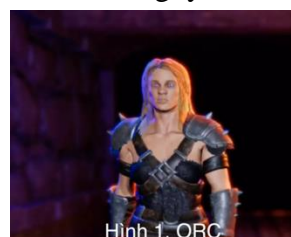
3.2.2. NPCs

• Enemy (kẻ địch)

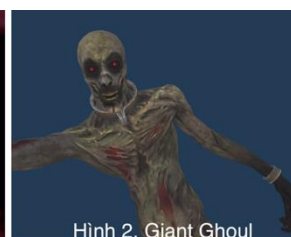
Enemy là một thành phần quan trọng không thể thiếu trong bất kỳ game nào. Có enemy thì mới có những thách thức và hấp dẫn đem đến cho người chơi trong việc trải nghiệm game.

Khi xuất hiện enemy, người chơi sẽ cần vận dụng các kỹ năng và chiến thuật để vượt qua và chiến thắng. Quá trình chiến đấu với enemy sẽ tạo ra những sự hưng phấn và kích thích khi đối đầu.

Chính vì lẽ đó, game được thiết kế với sự hiện diện của 4 loại enemies: ORC, Giant Ghoul, Gargoyle và Bernie.



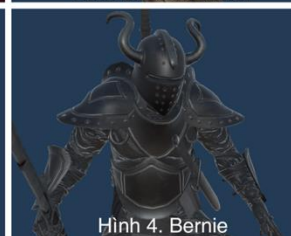
Hình 1. ORC



Hình 2. Giant Ghoul



Hình 3. Gargoyle



Hình 4. Bernie

Hình 4. Các enemies xuất hiện trong game.

NavMesh Agent là một thành phần mạnh mẽ của Unity, được sử dụng để điều khiển và quản lý di chuyển của đối tượng trong một môi trường NavMesh. NavMesh Agent hỗ

trợ điều hướng trong môi trường có nhiều trở ngại như vật cản, tường, đồ vật, v.v. [4]

Trong Sunny Kingdom, NavMesh được sử dụng cho tất cả các hành vi chuyển của enemies để xác định các vùng di chuyển khả thi và tìm đường đi an toàn và tối ưu để tránh va chạm với các trở ngại nhất.

NavMesh Agent tích hợp với hệ thống pathfinding, tương tác tốt với các hệ thống pathfinding khác trong Unity. Nó có thể được sử dụng cùng với hệ thống NavMesh để tạo ra một mạng lưới điều hướng chính xác và tùy chỉnh cho môi trường game. Điều này cho phép tạo ra các thuật toán pathfinding phức tạp như A* hoặc điều chỉnh độ ưu tiên của đường đi. Từ đó, NavMesh Agent sẽ tự động tìm đường đi tối ưu trên NavMesh và điều khiển đối tượng di chuyển đến đích.

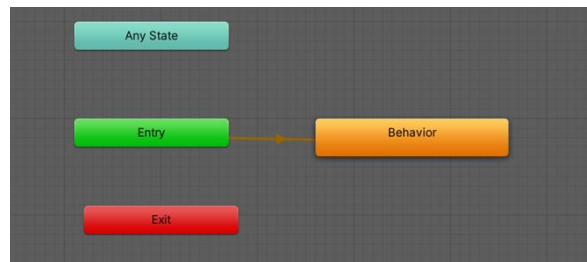
NavMesh Agent được tối ưu hóa để cung cấp hiệu năng cao trong việc điều khiển di chuyển của các đối tượng. Nó sử dụng NavMesh để xác định đường đi tối ưu và tận dụng các thuật toán pathfinding hiệu quả. Điều này giúp giảm thiểu tài nguyên tính toán và đảm bảo di chuyển mượt mà và chính xác.

- **Villager (dân làng)**



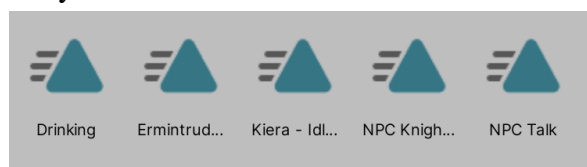
Hình 5. Một số villagers xuất hiện trong game.

Tương ứng với mỗi nhân vật, sẽ có Animator Controller riêng thể hiện trạng thái hành động của nhân vật đó.



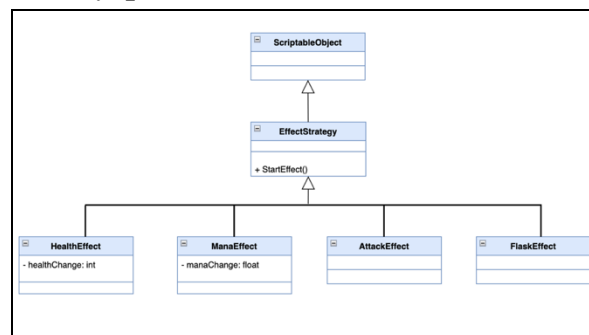
Hình 6. Animator Controller của nhân vật villagers.

Tương ứng với “Behavior”, sẽ có Animation Clip tương ứng thể hiện cho trạng thái này.



Hình 7. Một số animation clips sử dụng trong game.

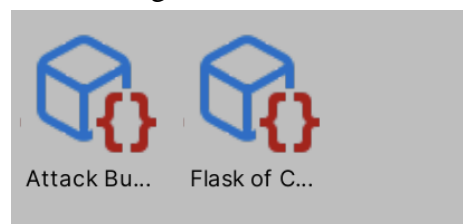
3.3. Vật phẩm



Hình 8. Class diagram minh họa các vật phẩm.

Lớp EffectStrategy kế thừa từ lớp ScriptableObject cho phép tạo ra các đối tượng dữ liệu có thể được sử dụng và quản lý trong game. Bằng cách tạo ra một thực thể của ScriptableObject, có thể tùy chỉnh các dữ liệu và quản lý chúng một cách linh hoạt.

Khi các lớp như HealthEffect, ManaEffect, AttackEffect, FlaskEffect kế thừa từ lớp EffectStrategy, cũng đồng thời thừa kế thừa từ lớp ScriptableObject. Khi đó, các lớp này có thể lưu trữ dưới dạng asset riêng biệt (như Hình 9 dưới), cho phép tối ưu hóa việc sử dụng bộ nhớ và tải dữ liệu.



Hình 9. Dạng asset của các classes kế thừa từ ScriptableObject.

Tương ứng với mỗi một lớp, chúng tôi sẽ tùy biến phương thức StartEffect() khác nhau tùy thuộc vào yêu cầu chức năng của loại vật phẩm tương ứng với lớp đó.

3.4. Hiệu ứng

Hiệu ứng luôn đóng một vai trò quan trọng và mang lại nhiều lợi ích trong việc trải nghiệm chơi game. Nhờ có hiệu ứng, game có thêm các trải nghiệm hấp dẫn, tăng cường sự tương tác và đem lại cảm giác thích thú cho người chơi. Có 2 loại hiệu ứng phổ biến nhất trong game là: SFX (Sound Effect) – hiệu ứng âm thanh và VFX (Visual Effect) – hiệu ứng hình ảnh.

3.4.1. SFX (Hiệu ứng âm thanh)

SFX được sử dụng để tái tạo âm thanh cho các sự kiện, hành động và tương tác trong game. Chẳng hạn, các tiếng nổ, tiếng bắn, tiếng nước chảy, tiếng cửa mở, tiếng tiếp xúc và tiếng nổ bom, v.v... SFX giúp tạo ra môi trường âm thanh sống động, tạo cảm giác thú vị cho người chơi.

Một trong các ứng dụng của SFX trong game Sunny Kingdom là phát ra một âm thanh (chẳng hạn, Item Receive Sound) nhằm cho biết người chơi mới vừa nhặt vật phẩm.

Để gọi SFX, đầu tiên cần tham chiếu đến thành phần AudioSource (_source) để phát ra âm thanh, sau đó gọi hàm PlayOneShot() với đối số là AudioClip (itemReceiveSound) được truyền vào để xác định âm thanh cần phát được minh họa trong Hình 10 dưới đây.

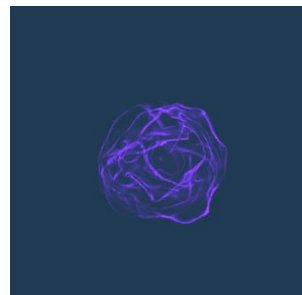
```
? usages ? overrides ? ext methods ? exposing APIs
public void PlayItemReceiveSound()
{
    _source.PlayOneShot(itemReceiveSound);
}
```

Hình 10. Đoạn mã minh họa việc xử lý SFX.

3.4.2. VFX (Hiệu ứng hình ảnh)

VFX là các hiệu ứng đồ họa được tạo ra để cải thiện trực quan và tạo ra những trải nghiệm hấp dẫn cho người chơi. Chẳng hạn, hiệu ứng ánh sáng, hiệu ứng hỏa tiễn, hiệu ứng nổ, hiệu ứng phóng tên lửa, v.v... VFX giúp tạo ra một thế giới game đẹp mắt, tăng tính thẩm mỹ, tạo cảm giác mãn nhãn cho người chơi.

Một trong các ứng dụng của VFX trong Sunny Kingdom là tạo ra hiệu ứng ma thuật khi người chơi chọn tấn công bằng phép thuật, được thể hiện trong Hình 11 dưới đây.



Hình 11. Hiệu ứng ma thuật.

Tuy nhiên, việc mỗi lần làm xuất hiện và biến mất 1 VFX, sẽ tương ứng với những lần Garbage Collector (Bộ dọn rác) dọn dẹp bộ nhớ cho những lần 1 VFX biến mất. Điều này diễn ra thường xuyên sẽ gây giảm hiệu năng đáng kể trong game vì Garbage Collector cần có thời gian và xử lý bộ nhớ Heap cho những lần dọn dẹp đó.

3.5. Môi trường cảnh quan

Toàn bộ môi trường cảnh quan được bao quanh bởi những dãy núi cao hùng vĩ, nhằm ngăn chặn người chơi hoặc bất kỳ đối tượng nào có thể di chuyển được thoát ra khỏi vùng giới hạn trong map.

Bắt đầu game, người chơi sẽ xuất hiện trong vùng màu xanh lá cây. Tuy nhiên, đích đến trong cuộc hành trình này lại là vùng màu đỏ. Tại vùng này, người chơi sẽ phải tìm vị trí của boss cuối trong vùng này.

Khi lang thang trên đường, người chơi sẽ gặp nhiều thắng cảnh (hang động, khu mộ, những ngôi làng, toà lâu đài, v.v...), dân làng và kẻ địch.



Hình 12. Toàn bộ môi trường cảnh quan trong game.

4. Kết quả thực nghiệm

Kết quả cho thấy ứng dụng của công nghệ Unity trong phát triển game đã mang lại một

số kết quả đáng chú ý. Hình 13c là màn hình khi vào game, cung cấp cho người chơi hướng dẫn để chuyển sang màn hình tiếp theo – màn hình menu (Hình 13b).

Tại màn hình menu, người chơi được cung cấp nhiều lựa chọn, bao gồm bắt đầu game, xem cốt truyện (Hình 13d) hoặc thoát game. Khi người chơi chọn bắt đầu, họ sẽ được chuyển đến màn hình Loading (Hình 13e). Màn hình này hiển thị tiến độ tải dữ liệu và tiến trình chuẩn bị game.

Sau khi tải hoàn tất, màn hình in – game (Hình 13f) sẽ hiển thị. Đây là phần giao diện chính, cung cấp tổng quan về các thông tin quan trọng cho người chơi như thanh máu, thanh mana, thanh điều hướng vị trí, số tiền đang có được, và vật phẩm đang trang bị trong quá trình chơi.

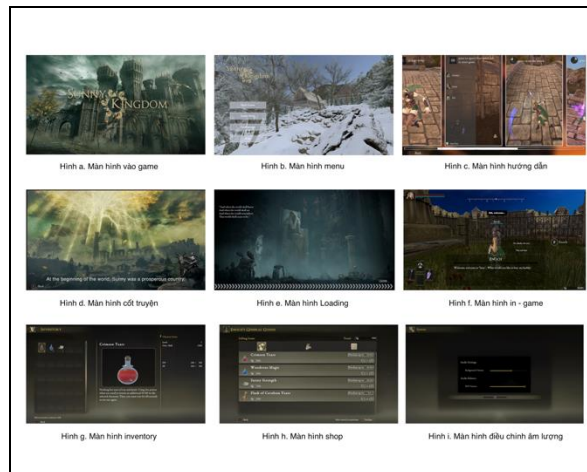
Ngoài ra, còn có các màn hình tương tác nhỏ khác như màn hình inventory (Hình 13g)

5. Kết luận

Dựa trên các kết quả nghiên cứu này, có thể thấy rằng sự ứng dụng của Unity trong phát triển game là rất đa dạng và linh hoạt. Unity không chỉ cung cấp nhiều tính năng và công cụ hỗ trợ, giúp tạo ra những game sống động và hấp dẫn, mà còn mở ra nhiều cơ hội nghề nghiệp và tiềm năng kinh doanh trong ngành công nghiệp game.

Đặc biệt, Unity đã trở thành một công cụ hàng đầu và được ưa chuộng trong lĩnh vực phát triển game không chỉ ở Việt Nam mà trên toàn thế giới. Sự phổ biến và ưu thế của Unity đã góp phần thúc đẩy sự phát triển và tiến bộ của ngành công nghiệp game, tạo ra những trải nghiệm mới mẻ và đáng nhớ cho người chơi.

giúp người chơi quản lý được các vật phẩm đã thu thập được, màn hình shop (Hình 13h) - mua các vật phẩm có sẵn trong game, và màn hình điều chỉnh âm lượng của âm thanh (Hình 13i).



Hình 13. Các màn hình giao diện trong game.

Tài liệu tham khảo

- [1] “Game Industry Usage and Revenue Statistics 2023”. Available: <https://helplama.com/game-industry-usage-revenue-statistics/#:~:text=The%20video%20game%20industry%20worth,for%20%2445.8%20billion%20in%202022>
- [2] “How the Video Game Industry Is Changing”. Available: <https://www.investopedia.com/articles/investing/053115/how-video-game-industry-changing.asp>
- [3] “Unity Pricing and Plans”. Available: <https://unity.com/pricing#plans-individualsand-teams>
- [4] “Bài toán tìm đường đi trong Unity”. Available: <https://viblo.asia/p/bai-toan-tim-duong-di-trong-unity-Qpmlenx75rd>
- [5] Dijkstra, E. W. (2022), “A note on two problems in connexion with graphs”, In Edsger Wybe Dijkstra: His Life, Work, and Legacy, pp. 287-290.