

TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT
VIỆN KỸ THUẬT CÔNG NGHỆ



BÁO CÁO TIỂU LUẬN CUỐI KỲ
UNITY ENGINE

Sinh viên thực hiện : Đình Thế Hoàng

Lớp : D20KTPM02

Khoá : 2023 - 2024

Ngành : Kỹ thuật phần mềm

Giảng viên hướng dẫn : ThS. Hồ Đắc Hưng

Bình Dương, tháng 4 / 2024

MỤC LỤC

PHẦN 1: UNITY 101	3
PHẦN 2: LỢI THẾ CẠNH TRANH CỦA UNITY Ở VIỆT NAM	8
PHẦN 3: PHÂN KHÚC KHÁCH HÀNG TẠI VIỆT NAM	10
PHẦN 4: ĐỐI THỦ CẠNH TRANH CỦA UNITY	12

PHẦN 1: UNITY 101

1. Khái niệm

Unity 101 là các bài học hoặc khóa học giới thiệu về cơ bản của Unity. Những nguồn tài nguyên này được thiết kế dành cho những người mới bắt đầu phát triển trò chơi hoặc mới làm quen với Unity. Chúng bao gồm các khái niệm cơ bản, bao gồm giao diện Unity, thiết lập dự án, đối tượng trò chơi, thành phần và viết script đơn giản.

Một khóa học Unity 101 điển hình có thể bao gồm:

- **Làm Quen với Giao Diện Unity:** Giới thiệu về bố cục, công cụ và cách điều hướng trong Unity.
- **Các Khái Niệm Cơ Bản:** Hiểu về đối tượng trò chơi, thành phần và cách chúng kết hợp với nhau để tạo ra chức năng trò chơi.
- **Cơ Bản về Scripting:** Giới thiệu về việc viết script bằng C# trong Unity để tạo ra cơ chế trò chơi cơ bản.
- **Hoạt Ảnh và Vật Lý:** Cơ bản về việc hoạt hóa nhân vật và đối tượng, cũng như áp dụng vật lý để tạo ra chuyển động và tương tác thực tế hơn.
- **Xây Dựng và Triển Khai:** Cách xây dựng một dự án trò chơi cho các nền tảng khác nhau và những điều cơ bản về triển khai trò chơi.

1.1. Game Object

Một GameObject trong Unity đơn giản là một đối tượng trong môi trường trò chơi của bạn. Nó có thể coi như một "container" hoặc "khung" mà bạn có thể thêm các tính năng và chức năng cụ thể thông qua các Components.

GameObjects có thể đại diện cho bất cứ thứ gì từ nhân vật, đối tượng môi trường như cây cỏ, đến các camera và nguồn sáng.

Một điểm đặc biệt quan trọng là GameObjects theo mặc định không làm gì cả; chúng trở nên "sống" và có ích khi được kết hợp với các Components.

1.2. Component

Components là các khối xây dựng cơ bản mà bạn gắn vào GameObjects để xác định hành vi, khả năng, và cách thức hoạt động của chúng.

Unity cung cấp một loạt các Components chuẩn như các hệ thống vật lý, hệ thống âm thanh, hệ thống đồ họa,...

Cách chúng hoạt động

GameObject và Components làm việc cùng nhau thông qua một mô hình "Composition" thay vì "Inheritance". Trong mô hình này, bạn có thể tạo ra các đối tượng phức tạp và linh hoạt bằng cách kết hợp nhiều Components khác nhau, thay vì phải kế thừa từ một lớp cơ sở phức tạp. Giúp cho phép bạn dễ dàng tùy chỉnh và tái sử dụng code, cũng như tạo ra các loại đối tượng trò chơi đa dạng mà không cần thay đổi cấu trúc cơ bản của chúng.

1.3. Giao tiếp các Class trong Unity

Các class giao tiếp với nhau qua một số phương thức khác nhau. Giao tiếp này là cốt lõi của việc tạo ra các ứng dụng phức tạp, cho phép các phần của chương trình chia sẻ dữ liệu và hành vi với nhau, một số cách bao gồm

- Thông qua Các Phương Thức Public (Công Khai): Các class có thể gọi các phương thức của nhau nếu những phương thức đó được đánh dấu là public. Điều này cho phép một class "A" thực thi một hành động trong class "B" bằng cách gọi một phương thức của "B".
- Sử dụng biến public: Tương tự như các phương thức public, một class có thể truy cập trực tiếp các biến public của class khác. Tuy nhiên, phương pháp này thường được coi là không an toàn vì nó có thể dẫn đến việc dễ dàng thay đổi trạng thái nội bộ của đối tượng một cách không mong muốn.
- Sử dụng getter và setter: Để kiểm soát việc truy cập vào các biến nội bộ, một class có thể cung cấp các getters và setters - những phương thức cho phép đọc và ghi giá trị của một biến mà không trực tiếp tiếp xúc với biến đó. Điều này giúp bảo vệ dữ liệu và duy trì tính đóng gói.
- Sự Kiện và Delegates: Unity hỗ trợ sự kiện và delegates, cho phép các class gửi thông báo cho các class khác mà không cần biết chi tiết về chúng. Điều này tạo ra một mô hình lắng nghe và phản hồi linh hoạt, giúp giảm sự phụ thuộc lẫn nhau giữa các class.

- **Interfaces:** Một interface định nghĩa một tập hợp các phương thức mà một class phải triển khai. Điều này cho phép giao tiếp dựa trên "hợp đồng", mà không phụ thuộc vào triển khai cụ thể, tăng cường tính mô-đun và khả năng tái sử dụng mã.
- **Singleton Pattern:** Singleton là một pattern thiết kế mà nó đảm bảo một class chỉ có một thể hiện duy nhất và cung cấp một điểm truy cập toàn cục đến thể hiện đó. Điều này hữu ích cho việc quản lý tài nguyên hoặc thông tin chia sẻ trên toàn ứng dụng.
- **Dependency Injection:** Dependency Injection là một kỹ thuật trong đó một đối tượng cung cấp các phụ thuộc của một đối tượng khác. Điều này giúp giảm sự phụ thuộc giữa các class, làm cho code dễ quản lý và bảo trì hơn.

1.4. Static

Statics trong lập trình, đặc biệt là trong Unity và C#, đề cập đến các thành phần static (tĩnh) bao gồm các lớp, biến và hàm. Các thành phần này tồn tại độc lập với bất kỳ thể hiện nào của lớp mà chúng thuộc về, chúng được chia sẻ giữa tất cả các thể hiện của lớp hoặc thậm chí có thể được truy cập mà không cần một thể hiện

Các thành phần bao gồm:

- **Lớp tĩnh (Static Class):** Ít được sử dụng, nó là một lớp không thể tạo ra các đối tượng riêng lẻ.
- **Biến tĩnh (Static Variable):** Biến này được chia sẻ giữa tất cả các đối tượng của cùng một lớp. Thay đổi giá trị của biến tĩnh ở một chỗ sẽ ảnh hưởng đến giá trị của biến đó ở tất cả các đối tượng khác.
- **Hàm tĩnh (Static Function):** Hàm này có thể được gọi trực tiếp bằng tên lớp mà không cần tạo đối tượng trước. Nó không thể truy cập vào các biến không tĩnh của lớp.

1.5. Unity, Awake, Start và OnEnable

Trong Unity, Awake, Start và OnEnable là ba hàm đặc biệt được gọi trong các thời điểm khác nhau của vòng đời của một MonoBehaviour (thành phần chính để tạo hành vi cho GameObject).

Hàm	Thời điểm gọi	Thích hợp cho
Awake	Ngay sau khi khởi tạo	Khởi tạo mặc định, tham chiếu các thành phần cùng GameObject
Start	Sau khi tất cả Awake được gọi	Logic phụ thuộc các thành phần khác trong cảnh
OnEnable	Mỗi khi GameObject bật	Thiết lập logic khi GameObject được kích hoạt

1.6. Prefabs

Prefabs trong Unity là một tính năng mạnh mẽ giúp tối ưu hóa quy trình làm việc và quản lý cảnh trong quá trình phát triển trò chơi. Prefab (viết tắt của "prefabricated") là một đối tượng trò chơi hoặc một tập hợp các đối tượng trò chơi với các thành phần và cài đặt đã được định cấu hình sẵn, có thể được lưu trữ, tái sử dụng, và khởi tạo nhiều lần trong môi trường trò chơi mà không cần phải tái tạo từ đầu.

Tính năng và lợi ích

- **Tái Sử Dụng:** Prefabs cho phép bạn dễ dàng tái sử dụng các đối tượng trò chơi cùng với tất cả các thành phần và cài đặt của chúng, giảm thiểu công sức cần thiết để tái tạo các đối tượng giống nhau hoặc tương tự nhiều lần.
- **Quản Lý Trung Tâm:** Khi bạn thay đổi một Prefab, mọi thể hiện của Prefab đó trong cảnh hoặc dự án của bạn sẽ tự động cập nhật để phản ánh những thay đổi đó. Điều này làm cho việc quản lý và cập nhật trò chơi trở nên dễ dàng hơn nhiều.
- **Hiệu Suất:** Sử dụng Prefabs có thể giúp tối ưu hóa hiệu suất trò chơi bằng cách giảm bớt số lượng công việc cần thực hiện khi khởi tạo các đối tượng trong thời gian chạy.

- Tùy Biến: Mặc dù các Prefabs được định cấu hình sẵn, nhưng bạn vẫn có thể tùy chỉnh các thể hiện riêng lẻ của chúng trong cảnh mà không ảnh hưởng đến Prefab gốc.

PHẦN 2: LỢI THẾ CẠNH TRANH CỦA UNITY Ở VIỆT NAM

Một số lợi thế cạnh tranh chính của Unity tại Việt Nam.

Khả năng Tích hợp và Linh hoạt

Unity cung cấp một môi trường phát triển linh hoạt và tích hợp cao, cho phép các nhà phát triển tại Việt Nam tạo ra các sản phẩm từ đơn giản đến phức tạp, từ trò chơi di động đến các ứng dụng VR/AR. Hệ thống này hỗ trợ nhiều nền tảng khác nhau, giúp việc phát triển sản phẩm trở nên dễ dàng và tiết kiệm thời gian hơn.

Cộng đồng và Hỗ trợ

Unity có một cộng đồng lớn và đa dạng, bao gồm cả các nhà phát triển Việt Nam. Cộng đồng này cung cấp sự hỗ trợ, chia sẻ kiến thức và tài nguyên, giúp giải quyết các thách thức trong quá trình phát triển. Ngoài ra, Unity còn tổ chức các sự kiện, hội thảo, và cuộc thi để kết nối và tạo điều kiện cho các nhà phát triển học hỏi và phát triển.

Công nghệ tiên tiến

Unity liên tục cập nhật các công nghệ mới và tính năng tiên tiến, bao gồm trí tuệ nhân tạo, máy học, và đồ họa 3D chất lượng cao. Điều này giúp các sản phẩm phát triển tại Việt Nam có thể cạnh tranh được với các sản phẩm quốc tế về mặt công nghệ và đồ họa.

Hỗ trợ Giáo dục và Đào tạo

Unity có những chương trình đào tạo và giáo dục cụ thể, bao gồm cả các khóa học trực tuyến và tại chỗ, để hỗ trợ các nhà phát triển mới và các sinh viên tại Việt Nam. Việc tiếp cận với nguồn học liệu phong phú và chất lượng giúp nâng cao kỹ năng và kiến thức về phát triển game và ứng dụng.

Môi trường Kinh doanh Thuận lợi

Việt Nam đang trở thành một trung tâm công nghệ mới ở khu vực Đông Nam Á, với một môi trường kinh doanh thuận lợi và nhiều chính sách hỗ trợ từ chính phủ cho ngành công nghệ thông tin. Unity đã và đang tận dụng lợi thế này để mở rộng và củng cố vị thế của mình tại thị trường Việt Nam.

Trong đầu năm gần đây, không ít doanh nghiệp đã đưa thông báo tuyển dụng IT nhưng lại gặp khó khăn. Nguyên nhân là do Chính phủ đưa ra chính sách thúc đẩy chuyển giao công nghệ và lấy công nghệ là tiền đề cho sự phát triển của nhiều ngành nghề khác. Chính vì vậy, cơ khát nhân lực IT nói chung và Unity Developer nói riêng diễn ra mạnh mẽ, gây nên sự cạnh tranh khốc liệt giữa các doanh nghiệp.

Từ những phân tích trên, chúng ta đã thấy rõ sự thiếu hụt cung – cầu nhân lực IT nói chung và Unity Developer nói riêng nên cơ hội cho người theo đuổi là rất lớn. Sở hữu năng lực vững vàng, ứng viên hoàn toàn tự tin ứng tuyển Unity Developer và nhận được mức lương hấp dẫn, dao động từ 20 - 30 triệu đồng/tháng.

Nếu có kỹ năng tốt và kinh nghiệm làm việc dày dặn trong ngành, Unity developer hoàn toàn có thể kiếm được mức lương cao hơn. Hay nếu làm việc tại công ty phát triển game lớn thì mức thu nhập sẽ cao hơn mặt bằng chung.

Bên cạnh đó là rất nhiều phúc lợi khác như được trang bị công nghệ hiện đại khi làm việc, tham gia khóa đào tạo nâng cao, cơ hội thăng tiến lên vị trí cao hơn như kỹ sư IT (IT Engineer), Giám đốc công nghệ thông tin (CIO),...

PHẦN 3: PHÂN KHÚC KHÁCH HÀNG TẠI VIỆT NAM

Trong ngành công nghiệp game ở Việt Nam, các studio game có thể hướng đến các phân khúc khách hàng dưới đây dựa trên sở thích, nhu cầu và khả năng chi trả:

1. Người Chơi Game Di Động

- **Đặc điểm:** Phân khúc này bao gồm người chơi mọi lứa tuổi, tập trung chủ yếu ở đô thị, thường xuyên sử dụng điện thoại thông minh. Họ tìm kiếm các trò chơi giải trí ngắn hạn, có thể chơi mọi lúc, mọi nơi.
- **Quy mô:** Đây là phân khúc lớn nhất tại Việt Nam do sự phổ biến của điện thoại thông minh.
- **Phù hợp với:** Các studio game Unity có khả năng phát triển game mobile với thời gian chơi ngắn, gameplay đơn giản nhưng hấp dẫn, có thể bao gồm các yếu tố xã hội để tăng tính tương tác.

2. Game Thủ PC và Console

- **Đặc điểm:** Đây thường là những người chơi có kinh nghiệm, dành nhiều thời gian và nguồn lực cho trò chơi, yêu cầu chất lượng cao về đồ họa và gameplay phức tạp.
- **Quy mô:** Phân khúc nhỏ hơn so với game di động, nhưng người chơi sẵn lòng chi tiêu nhiều hơn cho trải nghiệm chơi game chất lượng.
- **Phù hợp với:** Các studio có khả năng và nguồn lực để phát triển game có độ sâu nội dung và chất lượng cao, đặc biệt là các thể loại MMORPG, FPS, hoặc các game phiêu lưu, hành động.

3. Người Chơi Game Giáo Dục

- **Đặc điểm:** Phụ huynh và trẻ em tìm kiếm các trò chơi với mục đích giáo dục, giúp phát triển kỹ năng sống, học tập qua trò chơi.
- **Quy mô:** Phân khúc có tiềm năng tăng trưởng do sự quan tâm ngày càng cao của xã hội đối với giáo dục sớm và học qua trò chơi.
- **Phù hợp với:** Studio có khả năng phát triển các trò chơi giáo dục với nội dung phong phú, tương tác cao và thiết kế thân thiện với trẻ em.

4. Người Chơi Game Cạnh Tranh

- **Đặc điểm:** Phân khúc này bao gồm các game thủ tham gia các giải đấu và e-sports, tìm kiếm các trò chơi có tính cạnh tranh cao.
- **Quy mô:** Dù không lớn bằng phân khúc game di động, nhưng rất đam mê và sẵn lòng đầu tư thời gian, tiền bạc vào game và thiết bị chơi game.
- **Phù hợp với:** Studio có khả năng phát triển các tựa game cạnh tranh, hỗ trợ e-sports và có cơ chế thưởng phạt rõ ràng.

Phân khúc khách hàng phù hợp với các studio game Unity tại Việt Nam còn có thể được xác định dựa trên nhiều tiêu chí khác nhau, bao gồm độ tuổi, sở thích, mức thu nhập, và cả các yếu tố văn hóa cụ thể. ví dụ

- **Phân khúc khách hàng trẻ em và gia đình:** Trò chơi giáo dục, trò chơi vui nhộn và an toàn cho trẻ em. Ví dụ VNG Game Studios với các tựa game như “Sky Garden” có thể thu hút đối tượng gia đình và trẻ em.
- **Phân khúc khách hàng thanh thiếu niên và người trẻ tuổi:** Trò chơi có yếu tố cạnh tranh cao, game phiêu lưu, hành động, và e-sports. Ví dụ Amanotes, với các trò chơi âm nhạc độc đáo như “Magic Tiles 3”, hấp dẫn đối tượng thanh thiếu niên và người trẻ tuổi yêu thích âm nhạc và trò chơi có tính cạnh tranh.
- **Phân khúc khách hàng người lớn và làm việc:** Game có nội dung sâu sắc, chiến lược, và trò chơi mô phỏng. Ví dụ Hiker Games với các tựa game như “Falcon Age” có thể thu hút đối tượng người lớn thích các trò chơi có yếu tố phiêu lưu và mô phỏng.
- **Phân khúc khách hàng yêu thích văn hóa và lịch sử Việt Nam:** Game lấy cảm hứng từ lịch sử, văn hóa, và truyền thống Việt Nam. Ví dụ Emobi Games với tựa game “7554” mang đến trải nghiệm về lịch sử Việt Nam trong chiến tranh chống Pháp, phù hợp với những ai quan tâm đến lịch sử và văn hóa Việt Nam.

PHẦN 4: ĐỐI THỦ CẠNH TRANH CỦA UNITY

Unreal Engine

Ưu điểm so với Unity:

- Đồ họa: Unreal Engine cung cấp chất lượng đồ họa ấn tượng và hiệu ứng ánh sáng thực tế, thường được ưu tiên cho các dự án AAA.
- Hệ thống Blueprints: Cho phép phát triển game mà không cần đến lập trình chi tiết, thuận tiện cho những nhà phát triển không chuyên.

Nhược điểm so với Unity:

- Độ khó: Có độ học hỏi cao hơn Unity, đặc biệt với người mới bắt đầu.
- Tài nguyên: Cần cấu hình máy tính mạnh mẽ hơn để chạy mượt mà.

Godot

Ưu điểm so với Unity:

- Mã nguồn mở và miễn phí: Godot là một lựa chọn phổ biến cho các nhà phát triển độc lập và nhỏ với ngân sách hạn chế.
- Hỗ trợ đa nền tảng: Tương tự như Unity, nhưng với lợi ích của việc hoàn toàn miễn phí.

Nhược điểm so với Unity:

- Cộng đồng nhỏ hơn: Có ít tài liệu hướng dẫn và hỗ trợ kỹ thuật so với Unity.
- Tính năng: Mặc dù phát triển nhanh, nhưng Godot vẫn còn thiếu một số tính năng cao cấp so với Unity.

Cocos2d-x

Ưu điểm so với Unity:

- Hiệu suất cao với game 2D: Được tối ưu hóa cho các trò chơi 2D, đặc biệt là trên các thiết bị di động.
- Nhẹ và linh hoạt: Tốt cho các dự án với yêu cầu thời gian tải và hiệu suất cao.

Nhược điểm so với Unity:

- Hạn chế với game 3D: Không mạnh mẽ như Unity hoặc Unreal Engine trong việc phát triển game 3D.
- Ngôn ngữ lập trình: Sử dụng C++ có thể là một rào cản cho một số nhà phát triển.

Lợi Thế Cạnh Tranh của Unity

Unity Engine là một công cụ phát triển game và ứng dụng 3D phổ biến tại Việt Nam với nhiều lợi thế cạnh tranh so với các đối thủ khác có thể kể đến bao gồm

- Dễ sử dụng: Unity có giao diện trực quan và dễ sử dụng, giúp cho việc bắt đầu phát triển game và ứng dụng 3D trở nên dễ dàng hơn, ngay cả đối với những người mới bắt đầu.
- Tính linh hoạt: Unity có thể được sử dụng để tạo ra các sản phẩm cho nhiều nền tảng khác nhau, bao gồm PC, mobile, console, VR/AR, và web. Điều này giúp cho các nhà phát triển có thể tiếp cận nhiều đối tượng người dùng hơn.
- Cộng đồng lớn: Unity có một cộng đồng lớn và tích cực gồm các nhà phát triển, nghệ sĩ và người đam mê, những người luôn sẵn sàng giúp đỡ và chia sẻ kiến thức. Cộng đồng này cũng cung cấp nhiều tài nguyên học tập miễn phí và trả phí cho người dùng Unity.
- Giá cả hợp lý: Unity cung cấp nhiều gói giá cả khác nhau phù hợp với nhu cầu của mọi đối tượng người dùng, từ cá nhân đến doanh nghiệp.
- Nhiều tài nguyên học tập: Có rất nhiều tài nguyên học tập miễn phí và trả phí available online and offline cho Unity, bao gồm sách, video hướng dẫn, bài viết, và khóa học.