**ỦY BAN NHÂN DÂN TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**TRẦN QUANG TÍN**

**TÌM HIỂU GAMEENGINE UNITY3D VÀ ỨNG DỤNG PHÁT TRIỂN MỘT GAME THỰC TẾ**

**TÓM TẮT KHÓA LUẬN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**TRÌNH ĐỘ ĐÀO TẠO: ĐẠI HỌC**

**TP. HỒ CHÍ MINH, THÁNG 12 NĂM 2017**

**MỞ ĐẦU**

1. **Lí do chọn đề tài**

Ở hầu hết những trường Đại học hiện nay sinh viên đã được bồi dưỡng đầy đủ những kiến thức nền tảng về web, phần mềm và thậm chí là những nền tảng di động. Tuy nhiên, có khá ít trường có những môn học hay những chuyên đề chính thức về game hay lập trình game. Đó vẫn là một mảng mà sinh viên ta vẫn còn thiếu những kiến thức nền tảng để chuẩn bị bước vào cuộc đời lập trình viên. Đứng trên phương diện một sinh viên chưa có kiến thức về game mong muốn trau dồi thêm kiến thức cho bản thân và qua đó chia sẻ với mọi người nên em đã chọn đề tài “Tìm hiểu GameEngine Unity3D và ứng dụng xây dựng một game thực tế”.

1. **Kết cấu bài báo cáo**

Ngoài phần mở đầu, kết luận và danh mục tài liệu tham khảo thì bố cục có bốn chương chính:

Chương 1: Tổng quan về GameEngine và Unity3D.

Chương 2: Chi tiết về Unity3D.

Chương 3: Ứng dụng Unity3D xây dựng một game thực tế.

Chương 4: Những kỹ thuật quan trọng.

**CHƯƠNG 1: TỔNG QUAN VỀ GAME ENGINE VÀ UNITY3D**

1. **Khái niệm Game Engine**

Một GameEngine là một phần mềm được viết để thiết kế và phát triển video game, hiểu đơn giản nó là loại phần mềm trung gian kết nối tương tác của nhiều thành phần đồ họa, logic… vào cùng 1 hệ thống với nhau. Có rất nhiều loại GameEngine dùng để thiết kế game cho các hệ máy như hệ consoles(Xbox, PlayStation) hay máy tính cá nhân(PC). Chức năng cốt lõi của GameEngine phần lớn nằm trong công cụ dựng hình(kết xuất đồ họa) cho các hình ảnh 2 chiều(2D) hay 3 chiều(3D), công cụ vật lý (hay công cụ tính toán và phát hiện va chạm), âm thanh, mã nguồn, hình ảnh động(animation), dựng ảnh... Quá trình phát triển game tiết kiệm được rất nhiều thời gian và kinh phí khi tái sử dụng và tái thích ứng những tài nguyên trong cùng một GameEngine để tạo nhiều game khác nhau.

Chúng ta cũng cần chú ý, GameEngine không chỉ bao gồm thành phần tạo hình ảnh mà bao gồm rất nhiều thành phần khác như thực thi mã chương trình, trí tuệ nhân tạo, phân luồng, quản lý bộ nhớ, và kết nối mạng… Nếu Engine chỉ có chức năng render hình ảnh thì thường được gọi là Graphic Engine.

1. **Unity3D**

Unity3D là một engine game đa nền tảng được phát triển bởi Unity Technologies bằng ngôn ngữ C/C++, hỗ trợ mã viết mã bằng C#, JavaScript hoặc Boo. Đây là công cụ phát triển game được cộng đồng làm game ưa thích sử dụng. Hiểu đơn giản Unity 3D Game là một công cụ trung gian tương tác giữa thiết kế và lập trình để phát triển game, gắn kết với nhiều thành phần với nhau trong cùng 1 hệ thống. Unity 3D hỗ trợ tốt cho game phát triển trên nhiều nền tảng khác nhau, ứng dụng được tạo ra có thể hoạt động trên nhiều nhóm thiết bị từ trình duyệt web trên PC cho tới điện thoại hay máy tính bảng.

1. **Tại sao chọn Unity3D?**

• Chi phí thấp: Theo chính sách của Unity Technology, những cá nhân và doanh nghiệp có doanh thu dưới 100.000 USD/năm được dùng miễn phí engine Unity3D, và chỉ thu phí 1.500 USD/năm cho bản Profestional, một con số rất khiêm tốn so với những gì engine này mang lại.

• Cộng đồng rất lớn mạnh: Là engine phổ biến nhất trên thế giới nên Unity có cộng đồng sử dụng rất lớn mạnh. Mọi thắc mắc của bạn về Unity đều sẽ được thảo luận và trả lời trên website cộng đồng http://answers.unity3d.com. Các tân binh sẽ dễ dàng kế thừa, phát triển các scripts và kho đồ họa có sẵn để giúp cho sản phẩm phong phú hơn.

• Dễ dàng tiếp cận: theo đánh giá của các lập trình viên thì editor của unity là một điểm tích cực, giao diện cực kì trực quan và dễ sử dụng. Bên cạnh đó là cách kết hợp các component và script giúp lập trình viên dễ dàng hiểu được cách vận hành của hệ thống.

• Dễ dàng học hỏi: tài liệu để học trên trang web của unity được viết một cách rất bài bản từ mô tả cho đến các thuộc tính, đi kèm với đó là các ví dụ minh họa về sử dụng scripts hay các hướng dẫn thực hiện theo từng bước có cả ảnh chụp màn hình để minh họa. Ngoài ra còn có các chuỗi bài học với nhiều cấp độ từ những người mới bắt đầu cho đến những nhà phát triển đã có kinh nghiệm.

• Cập nhật thường xuyên: Unity 2017.2 là phiên bản mới nhất của được Unity giới thiệu vào tháng 10 năm 2017. Những tính năng ấn tượng được giới thiệu trong phiên bản mới này tạo thêm niềm tin tưởng vững chắc và sức mạnh khó vượt qua được từ Unity.

**CHƯƠNG 2: CHI TIẾT VỀ UNITY3D**

1. **Các khái niệm cơ bản**

* **Asset(tài sản):** Assets là thư mục nơi chứa tất cả các file asset trong Game như: meshes, textures, videos, animations, sounds, music......
* **Scene:** trong Unity, scene là một cảnh chơi (hoặc một phân đoạn), những màn chơi riêng biệt, một khu vực trong game hoặc thành phần có trong nội dung của trò chơi (các menu).
* **Game Object:** Có thể hiểu đơn giản Game Object là một thành phần để chứa các component tạo nên những thành phần trong game. Mọi cảnh chơi, mọi kịch bản đều được xây dựng lên từ các gameobject và tương tác giữa chúng.
* **Component:** là các thành phần được đính vào trong gameobject, bổ sung tính năng cho các gameobject. Mỗi component có chức năng riêng biệt. Đa phần các component phụ thuộc vào một transform, vì nó lưu trữ các thông số cơ bản của gameobject.
* **Script:** script được Unity xem như một component, đây là thành phần giúp lập trình viên xây dựng nên khung, luồng của game. Unity cung cấp cho lập trình viên khả năng viết script bằng ba ngôn ngữ: JavaScript, C#, Boo(ở những phiên bản cũ hơn).
* **Prefab:** Prefab thực chất là một gameobject được lưu trữ lại để tái sử dụng. Các gameobject được nhân bản từ một prefab sẽ giống nhau hoàn toàn, ngoại trừ thành phần Transform và các tham chiếu ra bên ngoài prefab để phân biệt và quản lý được tốt hơn.

1. **Những thành phần cơ bản**
   1. **Physic:** Hệ vật lý tích hợp của Unity cung cấp một cách đầy đủ các thành phần xử lý mô phỏng vật lý. Có hai hệ vật lý riêng biệt trong Unity: một cho vật lý 3D, và một cho vật lý 2D. Các khái niệm chính giống hệt nhau giữa hai hệ vật lí nhưng chúng được thực hiện bằng các component khác nhau.
      1. **Rigidbody:** Một Rigidbody là thành phần chính xử lí hành vi vật lý của một đối tượng. Với một Rigidbody đính kèm, đối tượng sẽ ngay lập tức đáp ứng những tính chất của trọng lực. Nếu một hoặc nhiều thành phần collider cũng được thêm vào, đối tượng sẽ phát hiện được va chạm và bị tác động từ lực của các va chạm. Thành phần rigidbody có một thuộc tính được gọi là IsKinematic giúp loại bỏ nó khỏi sự kiểm soát của các tác động vật lý và cho phép nó được điều khiển chuyển động từ script.
      2. **Collider:** collider là thành phần đề phát hiện va chạm trong hệ vật lí của Unity. Colliders có thể được thêm vào một đối tượng mà không cần thành phần Rigidbody để tạo ra các thành phần bất động của game như tường, sàn...
   2. **User Interface**: UI của unity cung cấp đầy đủ các thành phần cần thiết để xây dựng giao diện người dùng. Bên cạnh đó là khả năng tùy biến vô giới hạn các thành phần, ví dụ như có thể biến một hình ảnh thành một nút bấm bằng cách thêm vào cho nó một event trigger component để tương tác với các sự kiện.
      * **Animation:** Một animation trong Unity bao gồm hai thành phần chính: Clip: một chuyển động của nhân vật ví dụ như: chạy, nhảy, đứng yên …vv
      * **Animator:** chứa các clip của nhân vật và quyết định khi nào thì thể hiện các clip này. Mỗi gameobject chỉ có thể có một Animator và chứ nhiều clip trong Animator đó. Mọi thao tác điều khiển animation từ trong script đều thông qua Animator này mà không gọi trực tiếp tới clip.

Với Unity, thao tác tạo và sử dụng animation cho các nhân vật trở nên dễ dàng và trực quan nhờ khả năng kéo thả, giúp lập trình viên không tốn quá nhiều thời gian để điều chỉnh và thiết lập các thông số.

* 1. **Networking:** Networking là một phần không thể thiếu cho một game engine. Ý thức được vấn đề này nên Unity đã xây dựng nên một bộ networking rất đầy đủ và dễ sử dụng để hỗ trợ những nhà phát triển trong việc mang game của mình lên mạng network hay xa hơn là thông qua internet.
  2. **Scripting:** Unity hỗ trợ hai loại ngôn ngữ để biết script bao gồm C# và Javascript. Trước đây Unity cũng có hỗ trợ ngôn ngữ Boo nhưng thị phần nhà phát triển quá ít nên đã được cắt giảm. Tuy hỗ trợ song song hai ngôn ngữ tuy nhiên những chức năng mới hay đặc biệt thì đa phần được hỗ trợ trên C# trước. Script trong Unity thường được đính kèm vào trong một gameobject và có thể thao tác với bất kì gameobject hay component nào nó được tham chiếu đến. Để có khả năng đính kèm vào một gameobject như một component và có các tính chất của game thì script phải được kế thừa từ một lớp MonoBehaviour.

Những phương thức cơ bản của một Monobehaviour:

* **Awake**: gọi đầu tiên và duy nhất một lần trong vòng đời của gameobject, thường thì chúng ta sẽ gọi và truyền những cài đặt cho màn chơi trong hàm này. Hàm này chạy khi chưa bắt đầu render đồ họa nên không truy xuất được đến các gameobject, chỉ có thể sử dụng các biến.
* **Start:** hàm này chạy ngay sau Awake và lặp lại mỗi khi gameobject được tắt và bật trở lại. Ta có thể bắt đầu tác động đến những gameobject ở giai đoạn này.
* **Update**: được gọi lặp đi lặp lại mỗi một khung hình. Thời gian chạy của hàm này tùy thuộc vào độ phức tạp của nó. Nếu mất quá nhiều thời gian để thực hiện hàm này thì trò chơi sẽ bị giật nên cần hạn chế tối đa việc lạm dụng hàm này.
* **OnDestroy**: hàm này được gọi khi gameobject được xóa và giải phóng biến. Thường thì ta sẽ gọi những hàm giải phóng những tài nguyên không sử dụng nữa trong này.
  1. **Virtual Reality và Augmented Reality:**
* **VR:** VR vẫn là một công nghệ mới đối với đa số những nhà phát triển trên toàn cầu, tuy nhiên Unity đã hỗ trợ tốt để cho các nhà phát triển tiếp cận công nghệ một cách nhanh chóng và dễ dàng nhất. Với ba thiết bị chiếm gần trọn thị phần kính VR trên thị trường thì Unity đã hỗ trợ trực tiếp hai thiết bị là Oculus Rift và Samsung Gear. Còn đối với HTC Vive thay vì kết nối thiết bị và sử dụng ngay như hai kính đã đề cập ở trên thì HTC Vive phải qua một bước cài plugin nữa, plugin này cũng có sẵn trên AssetStore và cũng chỉ cần vài phút để cài đặt. Đó là những minh chứng cho thấy Unity đang làm chủ mảng phát triển các ứng dụng VR.
* **AR:** AR là một công nghệ ra đời và phổ biến sau VR, tuy nhiên hiện nay thị hiếu của thị trường cho thấy AR đang được ưa chuộng và có những bước tiến lớn trong khi VR đã có dấu hiệu chững lại. Với sự ra đời của ARKit và ARCore và ngay lập tức được Unity hỗ trợ, cộng với việc là nền tảng chính để phát triển của Vuforia(một nền tảng AR của bên thứ ba phát triển trên Unity và AndroidStudio) thì Unity là lựa chọn số một nếu như không sử dụng những nền tảng của Apple hoặc Google.

**CHƯƠNG 3: ỨNG DỤNG UNITY3D XÂY DỰNG MỘT GAME ĐƠN GIẢN**

1. **Mô tả game**

Game có hai chế độ chơi: chơi với máy và chơi với bạn bè thông qua internet hoặc mạng LAN.

Quá trình chơi chung của cả 2 hình thức chơi: mỗi người chơi sẽ có một xe tăng của riêng mình. Nhiệm vụ của người chơi là di chuyển xe tăng và tiêu diệt những xe tăng khác. Tất cả các xe tăng chơi chung đều là đối thủ cần phải tiêu diệt. Game sẽ quy định thời gian tối thiểu giữa 2 lần bắn của người chơi, sau mỗi lần bắn người chơi phải đợi hết khoảng thời gian này mới bắn được phát tiếp theo. Trong quá trình chơi, game sẽ tự động tạo ra các vật phẩm y tế tại các vị trí ngẫu nhiên, khi người chơi ăn được các vật phẩm y tế này thì sẽ được khôi phục năng lượng. Năng lượng và sức mạnh của đạn sẽ phụ thuộc vào level của người chơi và chế độ chơi.

Đối với chế độ chơi offline với máy thì người chơi sẽ có nhiều map tương ứng với nhiều level khác nhau, người chơi phải hoàn thành các điều kiện của level này để có thể chơi ở level tiếp theo. Các điều kiện có thể là:

* + Tiêu diệt các xe tăng khác trong một khoảng thời gian nhất định
  + Tiêu diệt tất cả các xe tăng với số lần chết ít hơn hoặc bằng một số cho trước

Độ khó của game sẽ tăng theo từng level. Những tiêu chí đánh giá độ khó của game:

o Khoảng cách địch có thể phát hiện ra người chơi(tỷ lệ thuận với độ khó)

o Thời gian tối thiểu giữa 2 lần bắn của người chơi(tỷ lệ thuận với độ khó)

o Năng lượng của xe tăng địch(tỷ lệ thuận với độ khó)

o Tốc độ xe tăng của người chơi(tỷ lệ nghịch với độ khó)

o Tốc độ xe tăng địch(tỷ lệ thuận với độ khó)

o Thời gian xuất hiện các vật phẩm y tế

Đối với chế độ chơi ở mạng LAN, mọi người chơi trong cùng một phòng sẽ có cùng level khi bắt đầu để đảm bảo tính công bằng. Khi người chơi đầu tiên bắt đầu sẽ thiết lập các tùy chọn của game. Mỗi lượt bắn trúng hay mỗi lượt giết được đối phương sẽ được cộng điểm. Người chơi đạt được số mạng quy định nhanh nhất sẽ là người giành chiến thắng. Sau khi kết thúc khoảng thời gian chơi mà không ai đạt được số mạng cần thiết thì ai giành nhiều điểm nhất sẽ thắng. Số lượng mạng không giới hạn nhưng người chơi phải chờ một khoảng thời gian để được hồi sinh.

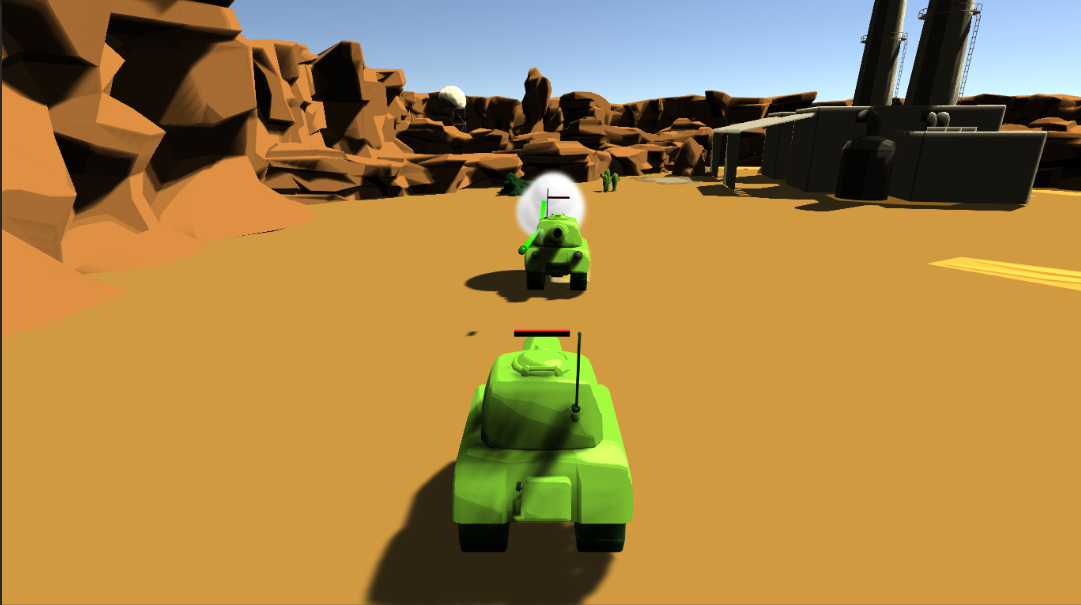
1. **Cách chơi và những màn chơi chính**

Khi khởi động game, người chơi sẽ được đưa đến màn hình bắt đầu. Ở đây cung cấp những thông tin về level hiện tại, những phòng hiện đang tồn tại. Ngoài ra người chơi cũng có thể tạo mới phòng và xem trước xe tăng của mình ở màn hình này. Thao tác tạo phòng chỉ đơn giản là nhập tên phòng và nhấn nút Create để tạo. Sau khi tạo phòng thì mặc định người tạo phải ở trong phòng đó và chờ những người chơi tiếp theo. Nếu không muốn tạo phòng mà tham gia vào những phòng có sẵn thì có thể nhấn Refresh tìm phòng rồi sau đó nhấn vào tên phòng để bắt đầu vào game.

****  
Hình 3.1: Màn hình khi vừa bắt đầu game

**Thao tác chơi**: người chơi sẽ dùng các nút mũi tên trên bàn phím để di chuyển và nhấn space để bắn. Sau khi đạt yêu cầu mà level quy định thì người chơi sẽ chiến thắng.

* **Chơi offline**: Khi bắt đầu một màn chơi người chơi sẽ có thời gian để đọc những thông tin về màn chơi như chế độ chơi, những thông tin về năng lượng, máu, thời gian bắn… Sau khi nắm thông tin người chơi sẽ bấm nút bắt đầu để tiếp tục.

  
Hình 3.3: Màn hình Game

* **Chơi online**: Vì đặc thù khi chơi trong mạng là chỉ có một chế độ chơi nên khi vào chơi sẽ không có màn hình thông tin mà vào thẳng game. Người vô sau vẫn sẽ có chung một điều kiện như những người vào trước đó. Quá trình và cách thức chơi tương tự như khi chơi offline.

**KẾT LUẬN**

Trên đây là những kiến thức cơ bản về bức tranh lập trình game hiện đại. Ta có thể thấy nếu đã có kiến thức nền tảng về ngôn ngữ C# thì phần còn lại không quá khó để có thể nắm bắt. Điều đó cho thấy sự yếu kém về mặt chuyên môn của sinh viên khi ra trường phần lớn là do sự thụ động, khả năng tự học không hiệu quả của sinh viên Với thời lượng của một bài khóa luận thì không thể nào truyền tải được hết kiến thức của cả một nền tảng game, thế nhưng lại là một phản ánh chân thực từ một góc nhìn của người mới bắt đầu.

Về phía nhà trường, để nâng cao hiểu biết cũng như kích thích sinh viên tìm tòi, học hỏi những cái mới thì nên tổ chức nhiều hơn nữa những buổi nói chuyện chuyên đề, giới thiệu công nghệ mới. Những buổi nói chuyện này tuy không truyền tải được nhiều kiến thức nhưng mang lại niềm tin và cái nhìn thực tế cho sinh viên từ những người đi trước.

Về phía sinh viên, trong thời buổi mà ngành Công nghệ thông tin liên tục phát triển thì việc sinh viên phải tìm hiểu và học hỏi những cái mới là không thể thiếu. Việc trang bị kiến thức chuyên môn rộng sẽ giúp ta dễ dàng hơn trong bước đầu tìm kiếm việc làm. Vậy nên sinh viên cần siêng năng hơn nữa, năng động hơn nữa để không bị bỏ lại trong thời đại chạy đua công nghệ này.