1. **Cơ bản về Unity**

**1. Unity Version**

Bước đầu tiên trong quá trình tiếp cận Unity là cài đặt và làm quen với phiên bản Unity mà bạn sẽ sử dụng. Việc lựa chọn phiên bản phù hợp là rất quan trọng, bởi vì mỗi phiên bản Unity đều có những tính năng và cải tiến riêng. Phiên bản Unity thường được khuyến nghị là phiên bản LTS (Long Term Support) vì tính ổn định cao và hỗ trợ lâu dài.

**2. UnityHub**

UnityHub là công cụ quản lý các phiên bản Unity và các dự án của bạn. Sau khi cài đặt UnityHub, bạn có thể dễ dàng cài đặt và quản lý nhiều phiên bản Unity khác nhau, cũng như tạo và mở các dự án một cách thuận tiện. UnityHub giúp đơn giản hóa quá trình quản lý và nâng cấp phiên bản Unity, đồng thời cung cấp một giao diện thân thiện cho người dùng.

**3. Unity Layout**

Khi mở Unity, bạn sẽ thấy giao diện người dùng được chia thành nhiều phần: Scene, GameView, Hierarchy, Inspector, Project, và Console. Bố cục này có thể được tùy chỉnh để phù hợp với phong cách làm việc của bạn. Việc làm quen với bố cục này là bước quan trọng để bạn có thể làm việc hiệu quả hơn.

**4. Scene và GameView**

Scene là nơi bạn xây dựng thế giới trò chơi của mình, đặt các Game Object và thiết lập các thuộc tính của chúng. GameView là cửa sổ nơi bạn có thể xem trước và kiểm tra trò chơi của mình trong thời gian thực. Việc chuyển đổi giữa Scene và GameView giúp bạn dễ dàng kiểm tra và điều chỉnh các thành phần của trò chơi.

**5. Hierarchy & Inspector**

Cửa sổ Hierarchy hiển thị cấu trúc cây của tất cả các Game Object trong Scene của bạn. Tại đây, bạn có thể sắp xếp, tạo mới, hoặc xóa các Game Object. Cửa sổ Inspector hiển thị các thuộc tính và thành phần của Game Object được chọn, cho phép bạn chỉnh sửa chi tiết từng thành phần.

**6. Game Object và Component**

Game Object là các đối tượng trong Scene của bạn, có thể là bất kỳ thứ gì từ các hình dạng cơ bản đến các đối tượng phức tạp như nhân vật hoặc môi trường. Các Game Object được tạo thành từ các Component, là các mô-đun chức năng như RigidBody, Collider, và Renderer. Việc hiểu rõ cách hoạt động của Game Object và Component là nền tảng để xây dựng trò chơi trong Unity.

**7. Physics**

Unity cung cấp một hệ thống vật lý mạnh mẽ để mô phỏng chuyển động và tương tác giữa các Game Object. Các thành phần vật lý như RigidBody và Collider cho phép bạn tạo ra các đối tượng có thể di chuyển, va chạm và phản ứng với lực. Việc nắm vững các nguyên tắc vật lý trong Unity giúp bạn tạo ra các trò chơi chân thực và hấp dẫn hơn.

**8. Custom Component với MonoBehaviour Script**

MonoBehaviour là lớp cơ bản cho tất cả các script trong Unity. Bằng cách viết các script tùy chỉnh kế thừa từ MonoBehaviour, bạn có thể tạo ra các thành phần độc đáo cho Game Object của mình. Các script này cho phép bạn điều khiển hành vi của đối tượng, xử lý các sự kiện, và tương tác với các thành phần khác trong trò chơi.

**9. Unity 2D**

Unity không chỉ hỗ trợ phát triển trò chơi 3D mà còn rất mạnh mẽ trong phát triển trò chơi 2D. Với Unity 2D, bạn có thể dễ dàng tạo ra các sprite, thiết lập các collider 2D, và sử dụng các thành phần vật lý 2D để xây dựng các trò chơi 2D. Unity cung cấp các công cụ và tài nguyên phong phú để hỗ trợ bạn trong việc tạo ra các trò chơi 2D chất lượng cao

## Chương 2: Nội dung thực tập

### I. Quy trình phát triển trò chơi

#### 2.1.1. Khái niệm

Phát triển trò chơi là quá trình phát triển một trò chơi điện tử, từ ý tưởng ban đầu đến sản phẩm hoàn chỉnh.

Tùy thuộc vào trò chơi và studio đảm nhận dự án, quá trình này có thể kéo dài từ vài tuần đến hơn một thập kỷ. Nó có thể liên quan đến hàng nghìn nhà thiết kế, nghệ sĩ, lập trình viên, nhà văn và người thử nghiệm hoặc được thực hiện bởi một nhà phát triển độc lập.

#### 2.1.2. Phát triển trò chơi và Thiết kế trò chơi

Phát triển trò chơi và thiết kế trò chơi là hai thuật ngữ khác nhau, mặc dù chúng thường được sử dụng thay thế cho nhau:

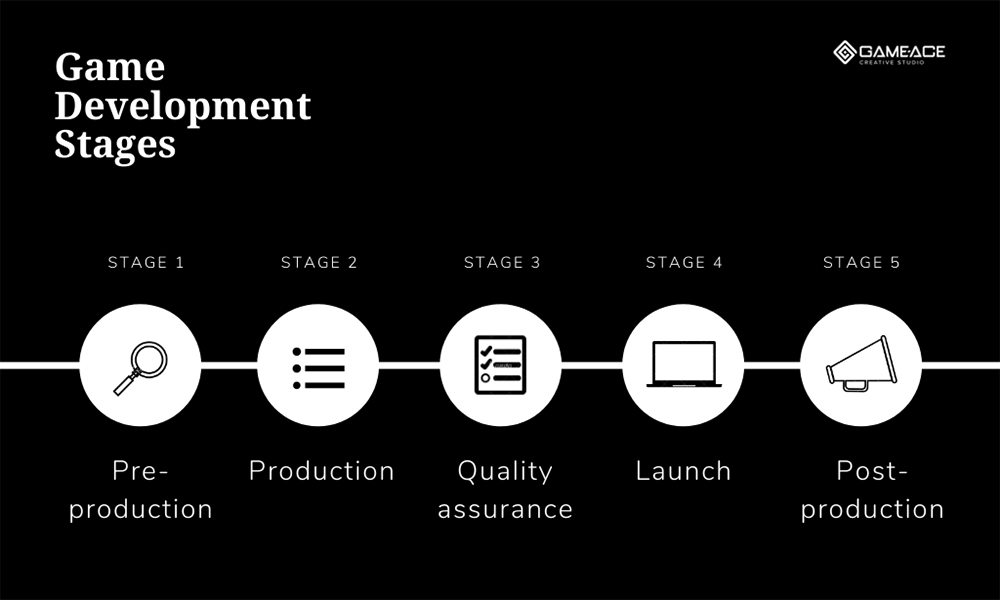
Thiết kế trò chơi đề cập đến khía cạnh khái niệm của mọi thứ: tầm nhìn ban đầu, cơ chế, câu chuyện, nhân vật, địa điểm, v.v.

Phát triển trò chơi là một thuật ngữ rộng hơn bao gồm thiết kế trò chơi và bao gồm cả việc triển khai kỹ thuật các khái niệm trò chơi.

Tại nhiều studio phát triển trò chơi nhỏ hơn, các thành viên cùng nhóm đội nhiều mũ và chịu trách nhiệm cho cả hai lĩnh vực. Tuy nhiên, tại các công ty lớn hơn, việc thiết kế và phát triển thường được xử lý riêng biệt.

#### 2.1.3. Các bước của quy trình

Một quy trình phát triển trò chơi thường bao gồm 5 giai đoạn chính: tiền sản xuất, sản xuất, đảm bảo chất lượng, khởi chạy và bảo trì hậu kỳ.



Hình 2. 1: Các giai đoạn của quy trình phát triển Game

##### 2.1.3.1. Giai đoạn 1: Tiền sản xuất trong quá trình phát triển trò chơi

Mỗi giai đoạn của quá trình tạo trò chơi là không thể tách rời, nhưng giai đoạn lập kế hoạch đầu tiên ảnh hưởng trực tiếp đến tất cả các chu kỳ tiếp theo. Điều quan trọng là bắt đầu quá trình phát triển trò chơi máy tính bằng cách thu thập thông tin về một sản phẩm trong tương lai, chẳng hạn như các yêu cầu kỹ thuật. Các nhà quản lý và chủ sở hữu sản phẩm làm việc để xác định mục tiêu và hoạt động kinh doanh của dự án và vạch ra khái niệm cốt lõi đằng sau nó.

##### 2.1.3.2. Giai đoạn 2: Sản xuất

Giai đoạn sản xuất là giai đoạn dài nhất và tốn nhiều công sức nhất của quy trình sản xuất trò chơi điện tử, được chia thành nhiều giai đoạn nội bộ.

**Prototyping**

Giai đoạn sản xuất đầu tiên là tạo ra một nguyên mẫu - một thử nghiệm của các cơ chế chính sẽ được sử dụng trong trò chơi. Tầm quan trọng của việc tạo mẫu này có thể được minh họa bằng ví dụ về tàu chiến Thụy Điển Vasa, được đóng vào thế kỷ 17.

**Visual content creation**

Khái niệm nội dung trực quan bao gồm mọi thứ mà một trò chơi chứa đầy - nhân vật, đạo cụ, môi trường, nội dung trò chơi. Tùy thuộc vào định dạng của trò chơi (2D / 3D), điều này sẽ ảnh hưởng lớn đến thời gian dành cho trò chơi.

**Game level design development**

Ở giai đoạn thiết kế cấp độ, các chuyên gia phát triển logic của các cấp độ trò chơi, khác nhau cho từng thể loại. Ví dụ: điều này có thể tăng mức độ khó sau khi hoàn thành nhiệm vụ tuyến tính hoặc khám phá thế giới trò chơi mở.

**Audio design and voice acting**

Âm thanh là một phần không thể thiếu trong thiết kế game nhập vai. Các chuyên gia âm thanh tạo nhạc phim và hiệu ứng âm thanh cho các tình huống khác nhau. Ngoài ra, hiệu ứng âm thanh cung cấp cho người chơi những tín hiệu cụ thể về nguy hiểm, chiến thắng, thất bại, v.v.

**Coding**

Khi tất cả các yếu tố của trò chơi đã sẵn sàng, các nhà phát triển viết hàng nghìn dòng mã để kết hợp tất cả nội dung lại với nhau và thường sử dụng công cụ trò chơi để làm điều đó. Khó khăn của nhiệm vụ này nằm ở chỗ một trò chơi phải phát triển giống như một trò chơi xếp hình, không bao gồm các cơ chế loại trừ lẫn nhau, và cũng hoạt động không có lỗi và hỏng hóc. Các nhà phát triển trò chơi cố gắng tính đến tất cả các yếu tố kỹ thuật có thể có và tạo ra trải nghiệm người dùng tích cực.

##### 2.1.3.3. Giai đoạn 3: Đảm bảo chất lượng

Một trò chơi có bất kỳ độ phức tạp nào cũng cần được kiểm tra để đảm bảo rằng trò chơi đó không có lỗi và hoạt động không có lỗi. Điều này là do một lỗi duy nhất có thể ảnh hưởng nghiêm trọng đến trải nghiệm người dùng và trải nghiệm tổng thể của trò chơi. Do đó, thử nghiệm chức năng, thử nghiệm phi chức năng, cũng như thử nghiệm Alpha và Beta thường được thực hiện.

|  |  |
| --- | --- |
| **Các loại thử nghiệm chức năng** | **Các loại thử nghiệm phi chức năng** |
| * Thử nghiệm trò chơi khả năng tương tác * Kiểm tra trò chơi hồi quy * Thử nghiệm trò chơi khói * Thử nghiệm trò chơi bản địa hóa * Thử nghiệm trò chơi kiểm soát truy cập bảo mật * Thử nghiệm trò chơi chấp nhận người dùng | * Kiểm tra hiệu suất trò chơi * Tải thử nghiệm trò chơi * Thử nghiệm trò chơi căng thẳng * Kiểm tra độ ổn định của trò chơi * Thử nghiệm trò chơi khối lượng * Thử nghiệm trò chơi khả năng sử dụng * Thử nghiệm trò chơi tuân thủ * Thử nghiệm cài đặt trò chơi * Kiểm tra cấu hình trò chơi * Thử nghiệm trò chơi khôi phục thảm họa |

Khi chức năng, hiệu suất, khả năng tương thích và các tính năng khác được kiểm tra và tất cả các lỗi đã được khắc phục, đó là lúc dành cho các giai đoạn phát triển trò chơi Alpha và Beta. Thử nghiệm Alpha là thử nghiệm cuối cùng của một trò chơi đã hoàn thành, trong đó các kỹ sư QA kiểm tra và sửa các lỗi cuối cùng. Đôi khi nó được theo sau bởi thử nghiệm Beta, không được thực hiện bởi các chuyên gia mà bởi những người bắt chước người dùng cuối.

##### 2.1.3.4. Giai đoạn 4: Khởi chạy

Sản phẩm ra mắt là giai đoạn phát triển trò chơi cuối cùng, được mọi người háo hức chờ đợi. Nhưng ra mắt không phải là kết thúc của câu chuyện. Thông thường, có đủ lỗi và sai sót ngay cả khi một trò chơi đã sẵn sàng, vì vậy song song với việc ra mắt, nhóm phát triển trò chơi tiếp tục mang đến nhiều chi tiết và cải tiến cho trò chơi. Đồng thời, người thử nghiệm thu thập những phản hồi đầu tiên từ người dùng thực để nhà phát triển thực hiện những thay đổi quan trọng.

##### 2.1.3.5. Giai đoạn 5: Hậu kỳ

Sau khi trò chơi được tung ra thị trường, các bản sửa lỗi và cải tiến yêu cầu phải theo dõi liên tục để kiểm tra lại mức độ ổn định và hiệu suất. Tốt nhất, các studio sẽ thường xuyên phát hành các bản cập nhật để đáp ứng các yêu cầu kỹ thuật của các nền tảng, các bản cập nhật này cũng được cập nhật.

Ngoài ra, nhiều công ty cập nhật nội dung trong trò chơi và đôi khi thêm các cơ chế mới thú vị để thu hút người dùng tham gia vào trò chơi. Tất cả điều này đòi hỏi phải duy trì chất lượng thông qua bảo trì liên tục và thử nghiệm thường xuyên các phiên bản mới của trò chơi.

### II. Dự án Brick Breaker

#### 2.2.1. Tổng quan dự án

Đây là một dự án Game 2D Casualcho Mobile mà anh Cường đã giao cho em thực hiện với sự giúp đỡ của anh Quang – Mentor của em trong kỳ thực tập.

##### 2.2.1.1. Phân công

Vì là dự án nhỏ nên sẽ chỉ có 4 người tham gia trực tiếp vào dự án:

* Anh Cường – Leader: Chịu trách nhiệm theo dõi và đánh giá tiến độ.
* Anh Phúc và Anh Danh – Designer: Chịu trách nhiệm thiết kế đồ hoạ.
* Em sẽ là Developer chính của dự án: Chịu trách nhiệm Code và Build Game.

##### 2.2.1.2. Gameplay

Bắn bóng lên để phá gạch và người chơi sẽ phải đỡ quả bóng khi rơi xuống bằng cách di chuyển Paddle (thanh ngang), khi bóng va chạm với Paddle thì sẽ phản xạ lên lại để tiếp tục phá những viên gạch cho đến khi hết gạch. Trong quá trình chơi sẽ có tỉ lệ rơi ra những vật phẩm trợ giúp để chơi dễ dàng hơn.

##### 2.2.1.3. Đồ hoạ và âm thanh

Vì Game thuộc thể loại Casualnên đồ hoạ sẽ theo phong cách đơn giản và bắt mắt để phù hợp với mọi đối tượng người chơi. Hiệu ứng âm thanh sẽ khá cơ bản và thông dụng.

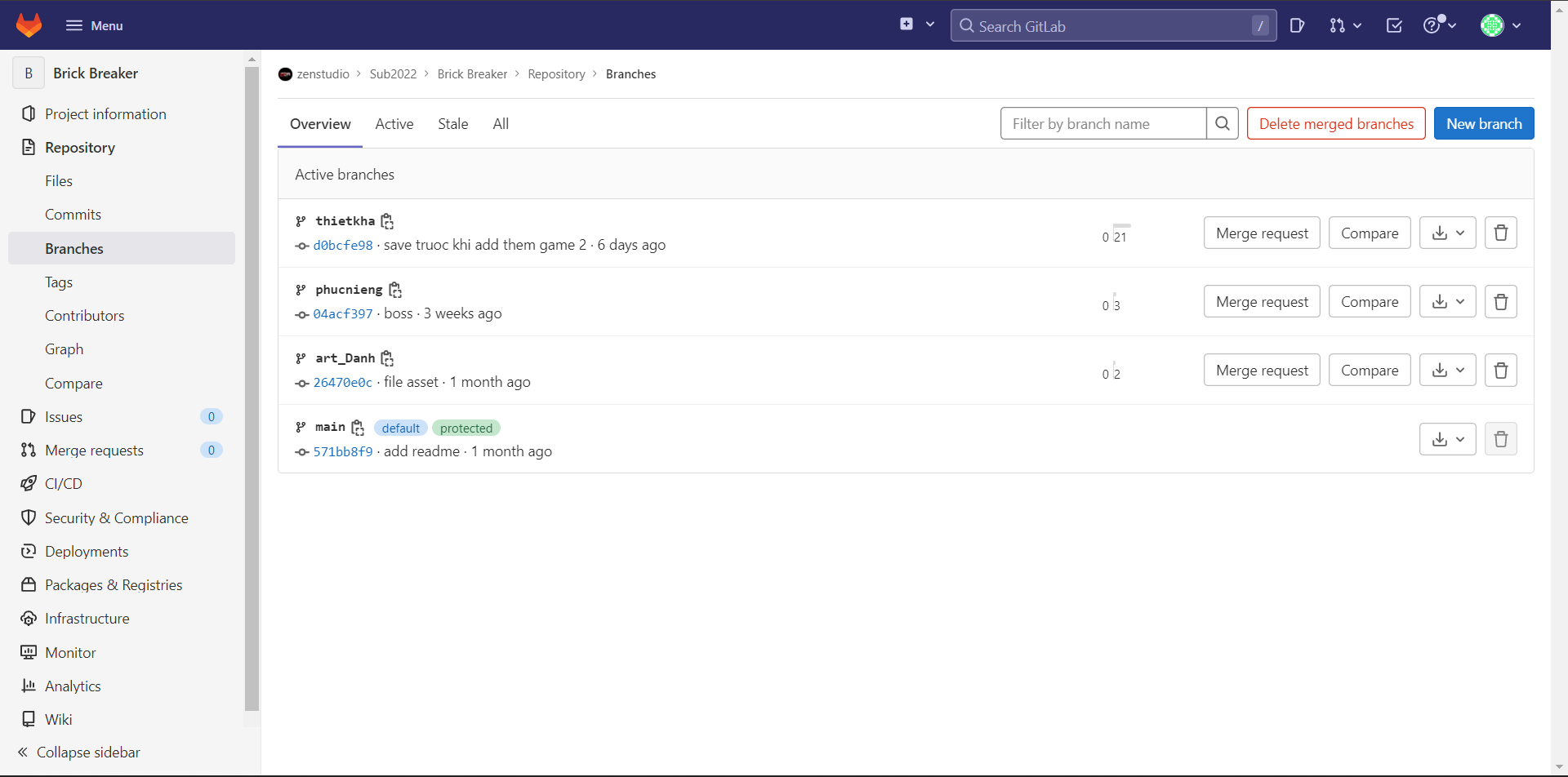
##### 2.2.1.4. Độ khó và mode

Game sẽ bao gồm khoảng 500 levels và một mode infinity (Chế độ chơi vô hạn) – trong mode này Gameplay sẽ có sự thay đổi sẽ không cần sử dụng Paddle nữa mà thay vào đó chỉ cần bắn bóng lên để phá gạch và điều đặc biệt ở đây là số lượng banh sẽ nhiều hơn, khi bắn chúng sẽ đi theo nhau như một đường thẳng và những viên gạch sẽ có “máu” chỉ khi hết “máu” thì viên gạch mới bị phá huỷ. Để có thể phá những viên gạch cực nhiều máu thì số lượng banh sẽ không chỉ có một nên trong quá trình chơi thì số lượng banh có thể tăng lên thông qua vật phẩm.

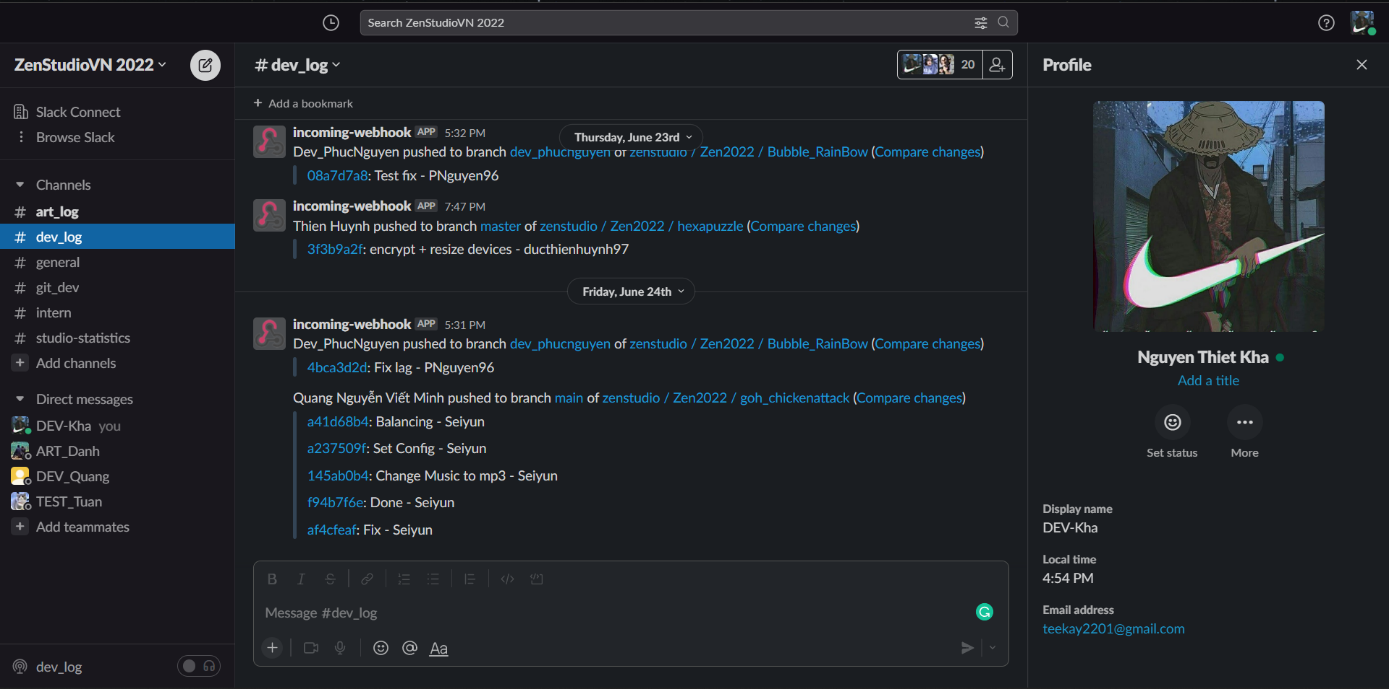
#### 2.2.2. Chi tiết dự án

##### 2.2.2.1. Quản lý

Tất cả file code, design sẽ đều được quản lý,chỉnh sửa và cập nhập trên GitLab. Báo cáo tiến độ dự án hàng ngày thông qua Slack. Tất cả những hoạt động sẽ được anh Cường theo dõi và kiểm soát.



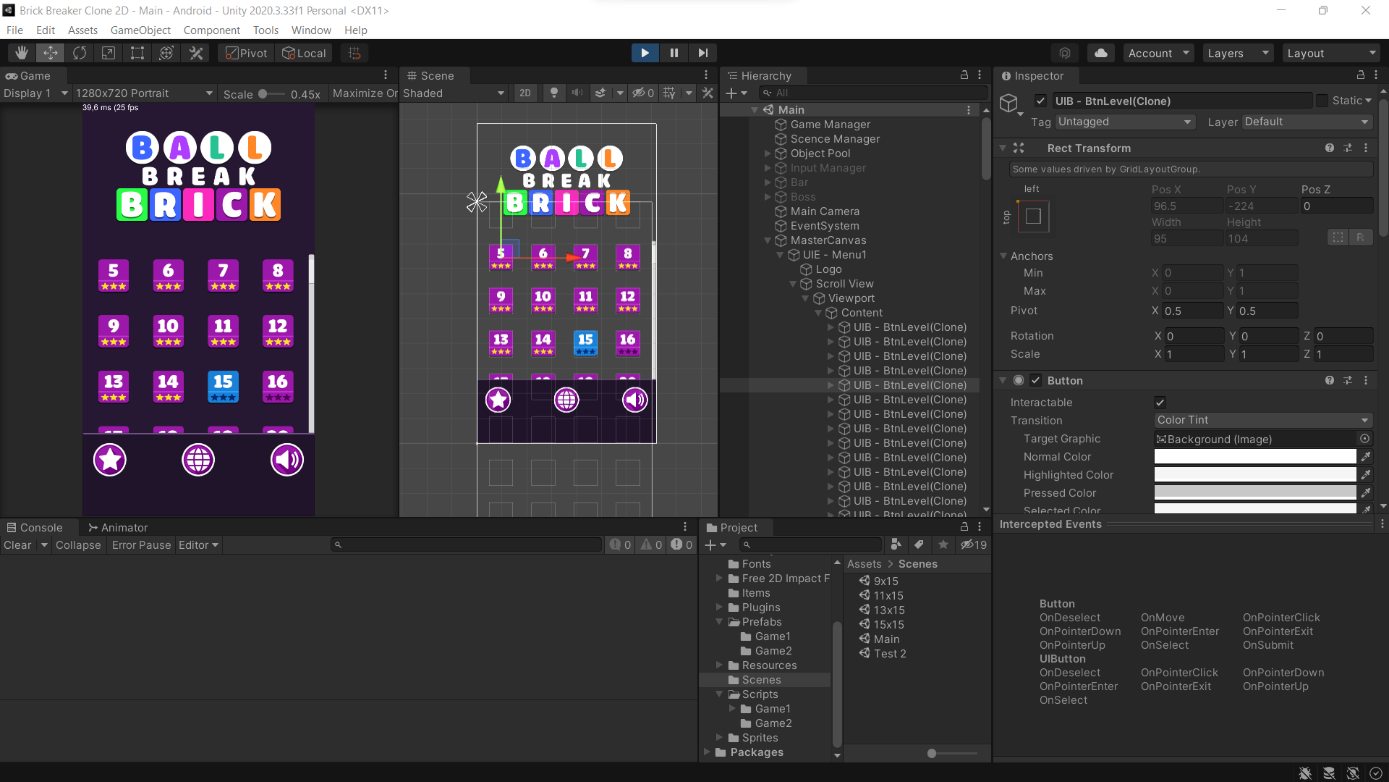
Hình 2. 2: GitLab



Hình 2. 3: Slack

##### 2.2.2.2. Game Engine

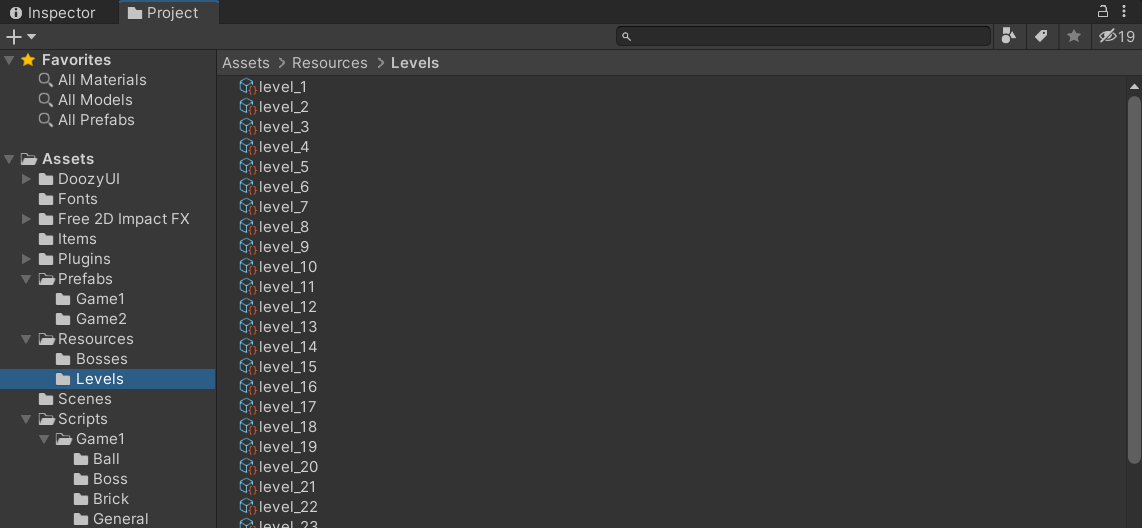
Thế mạnh của em là đã được tiếp xúc và làm việc với Game Engine Unity3D được hơn nửa năm thông qua Youtube, Discord và Document. Nên dự án này em đã sử dụng Unity3D để thực hiện.



Hình 2. 4: Giao diện của Unity3D

##### 2.2.2.3. File Asset

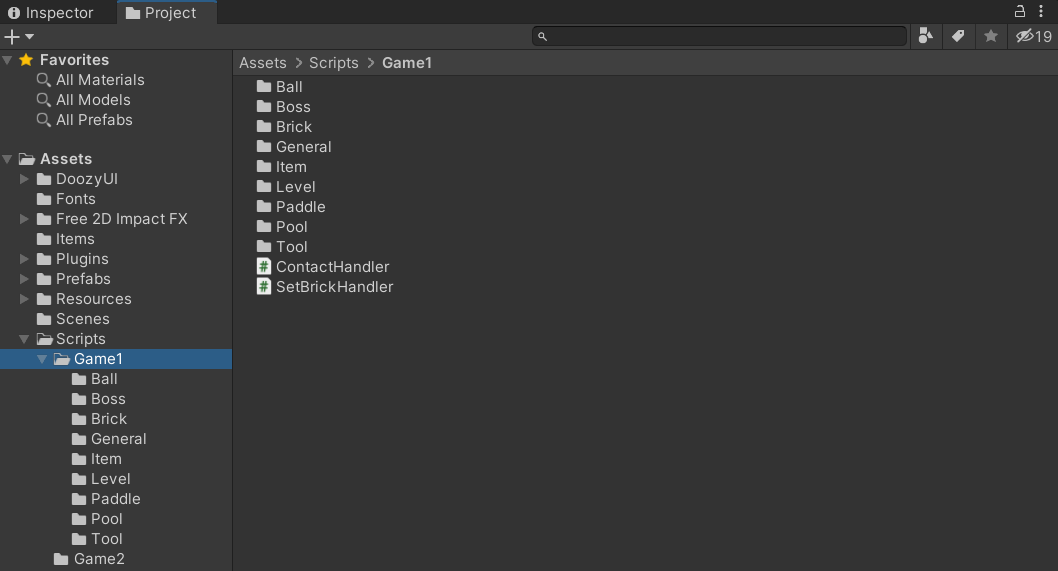
Folder Assets là nơi lưu trữ tất cả những File được dùng trong Game. Từ Scripts (Code), Prefabs, Scences, Sprites (Đồ hoạ 2D), Effects cho đến những Tool hỗ trợ cho việc lập trình và thiết kế,...



Hình 2. 5: File Asset trong dự án

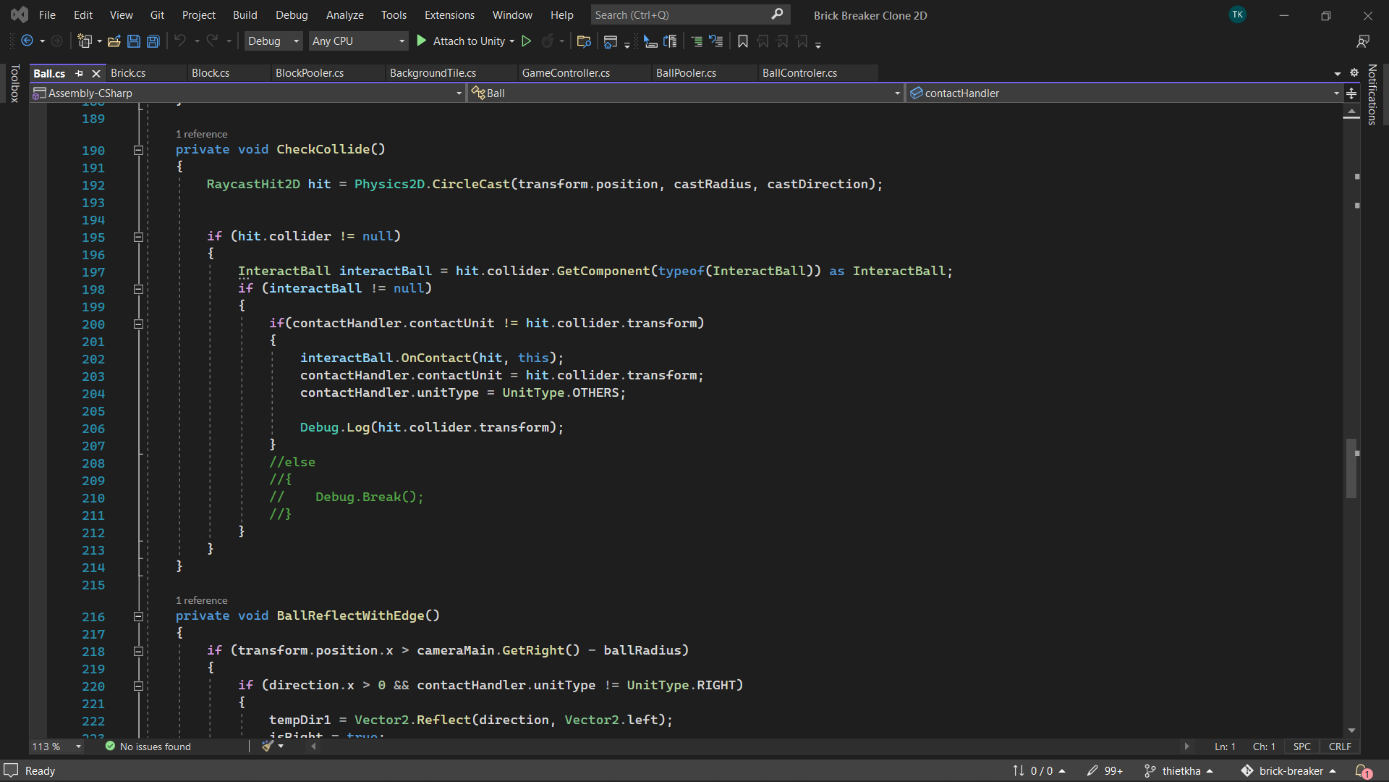
##### 2.2.2.4. Một số Script nổi bật

Vì hiện nay tổng số Script nếu không tính phần Tool và Mode đã là 27 Scripts và có những Script lên đến hơn 300 dòng nên em không thể báo cáo hết được chi tiết. Do đó em xin phép báo cáo một số chỗ nổi bật trong dự án.



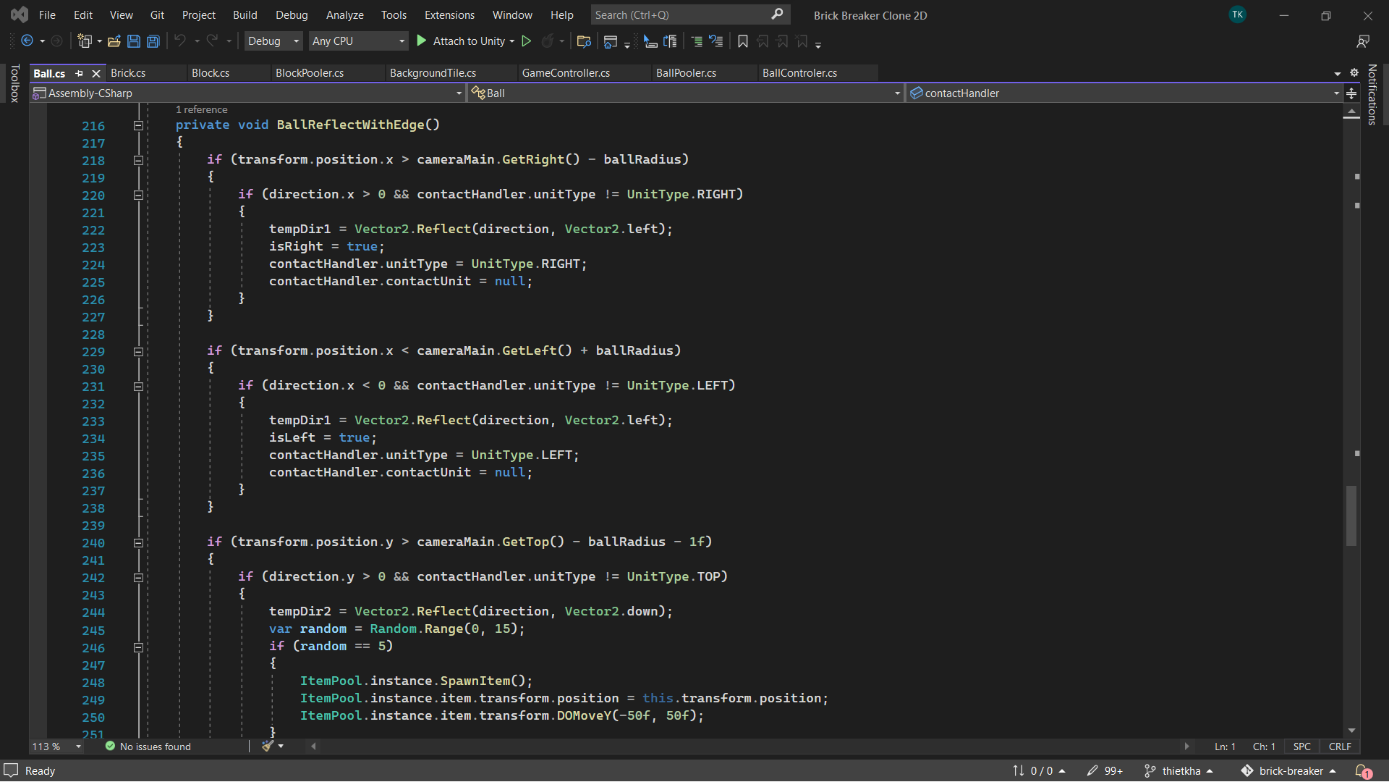
Hình 2. 6: Folder lưu trữ toàn bộ Scripts của dự án

**Ball Script:**



Hình 2. 7: Ball Script 1

Ở đây em tạo một phương thức CheckCollide() để kiểm tra xem bóng có va chạm hay không. Trong đó em sử dụng Physics2D.CircleCast() là một phương thức của Unity để tạo ra một Collider hình tròn xung quanh bóng và nếu có va chạm sẽ trả về vật đã va chạm với bóng sau đó Interface cho các Script của Object khác như Brick, Paddle.



Hình 2. 8: Ball Script 2

Trong phương thức tiếp theo là BallReflectWithEdge() vì các cạnh của màn hình không phải là Object nên không thể nào kiểm tra va chạm được nên trong phương thức này em đã lấy kích thước màn hình trừ đi bán kính của bóng để khi bóng chạm đến viền màn hình thì sẽ thực hiện một phương thức Reflect() dựa vào điều kiện là vị trí “x” của bóng đang ở đâu để tính và trả về một Vector phản xạ lại bằng Vector tới (hướng va chạm của bóng) với Vector normal (Vector vuông góc với mặt phẳng). Phương thức này bắt đầu từ dòng 216 đến 303 nên em không thể chụp hết được. Đoạn Script sau em thêm một số điều kiện để kiểm tra và thay đổi hướng phản xạ của bóng nếu như hướng phản xạ trùng với Vector normal.

**Link Google Drive file APK demo của dự án:** <https://drive.google.com/file/d/17HnoNZfeXtllxKaIhYLFCGHAOKv029-6/view?usp=sharing>