**Báo Cáo Tiến Độ: Đồ Án 1**

**Develop a game application  
Game: DoubleTee in Wonderland**

**GVHD**: ThS. Lê Văn Vinh

*Trường đại học Sư phạm Kỹ thuật TP.HCM*

**Nhóm 5:**

|  |  |
| --- | --- |
| **MSSV** | **Họ và tên** |
| Đinh Minh Thiện | 18110205 |
| Huỳnh Nhựt Thiên | 18110203 |

1. **Giới thiệu về đề tài.**
   1. **Giới Thiệu & Mục Đích**

**DoubleTee in Wonderland** là một trò chơi phiêu lưu giải đố độc lập miễn phí với các yếu tố giải trí được thiết kế bởi nhà phát triển dựa theo tựa game nổi tiếng Helltaker thiết kế bởi nhà phát triển Ba Lan Łukasz Piskorz, còn được gọi là vanripper.

* 1. **Nền tảng xây dựng ứng dụng**

Xây dựng trên ngôn ngữ lập trình C#, thiết kế và vận dụng các kiến thức về lập trình Unity.

* 1. **Tính năng chính**

- Background - giao diện

- Nhân vật

- Cốt truyện game (yếu tố phụ)

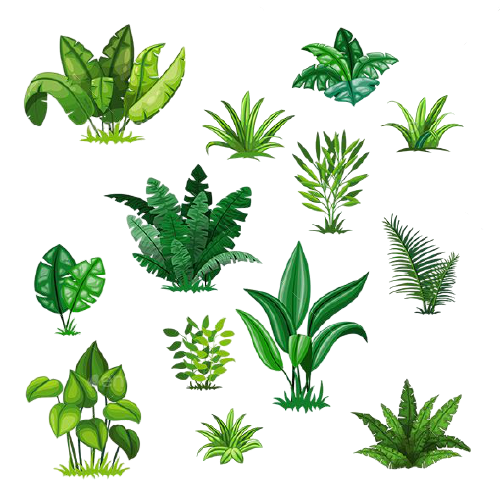
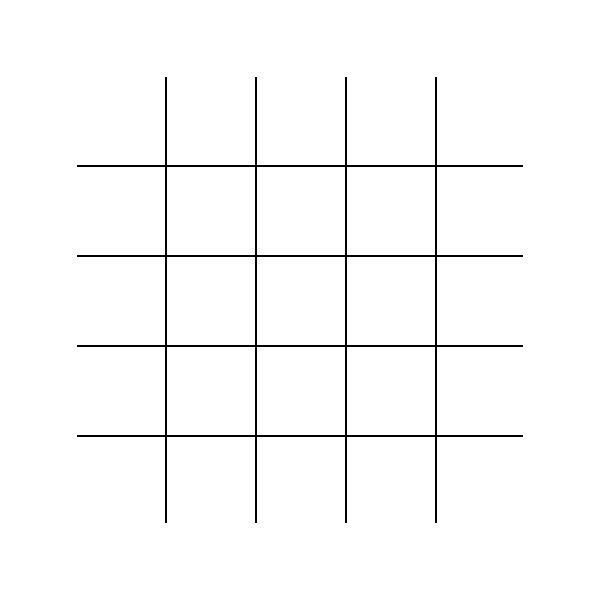
- Đồ họa

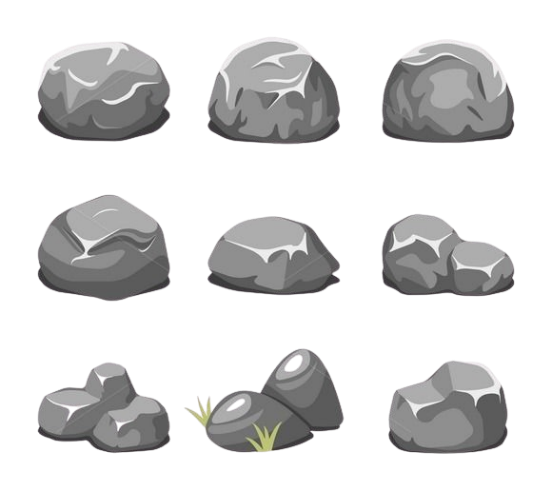
- Thuật toán và thiết kế cách chơi và câu đố của game

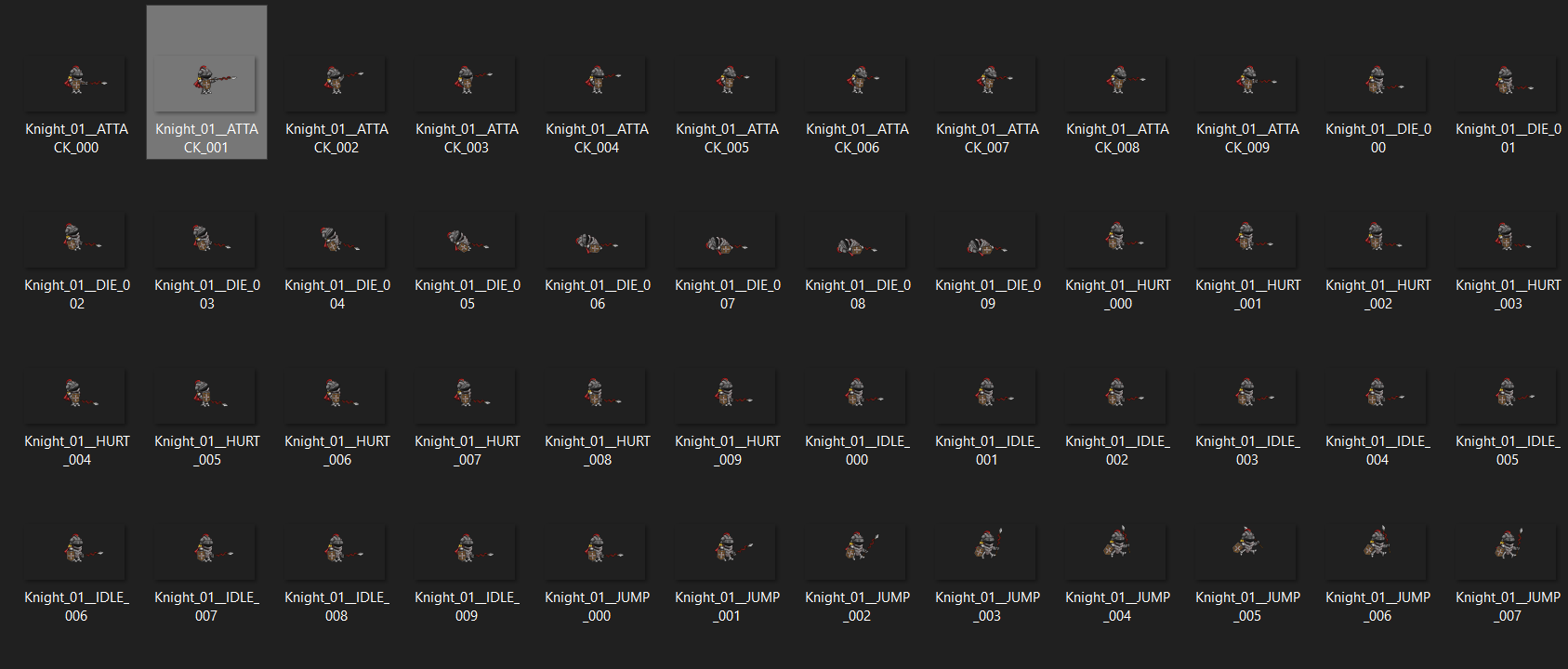
- Màn chơi thường và đặc biệt

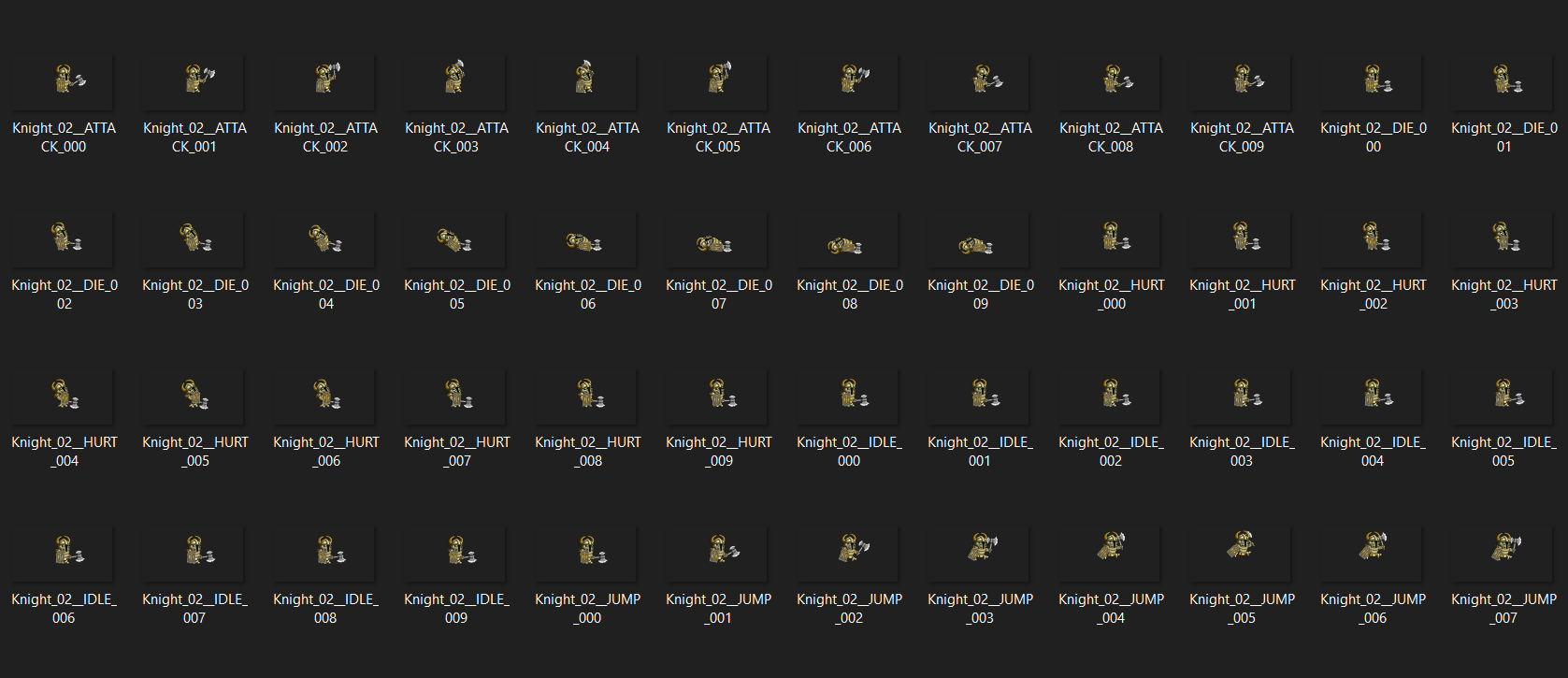
1. **Tiến độ thực hiện đề tài**

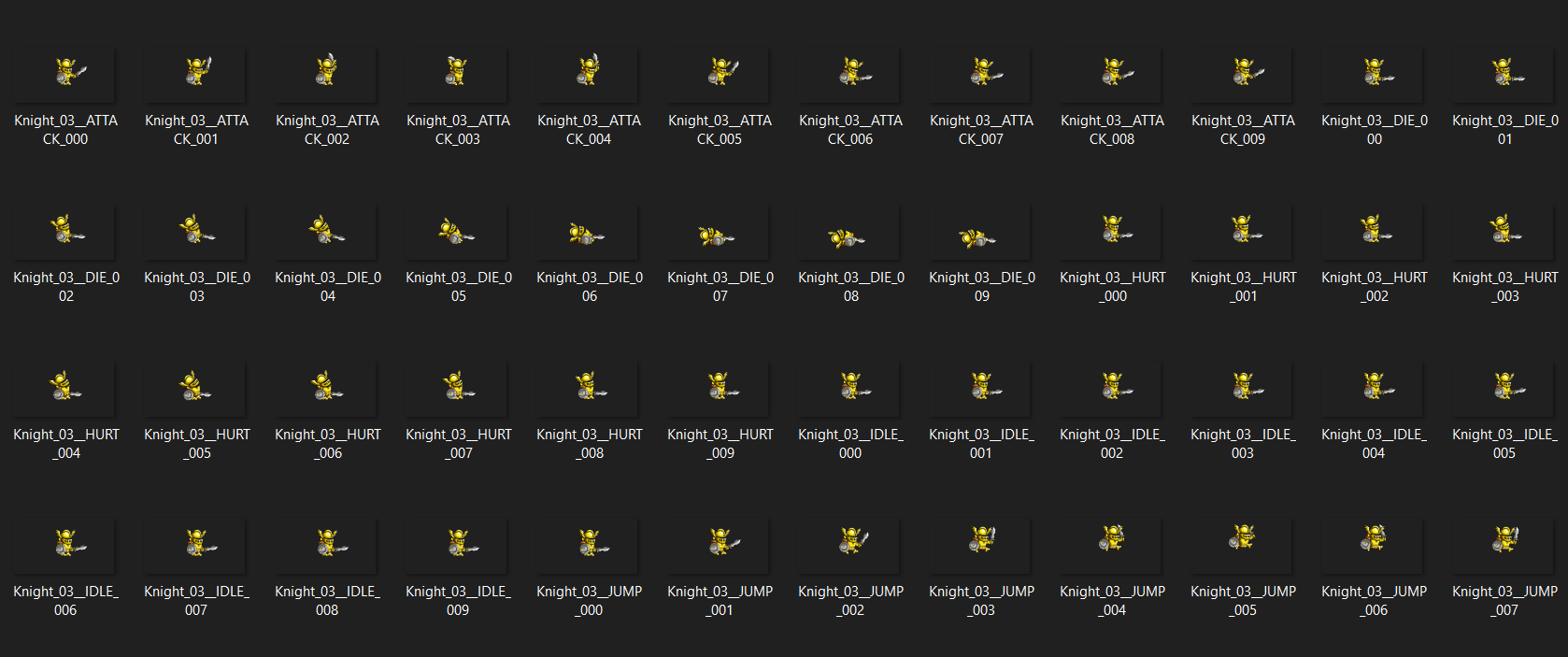
* **Tìm hiểu và làm các thao tác cơ bản trên unity qua các video trên mạng.**
* **Lên ý tưởng cũng như thống nhất về nội dung và giao diện của game**
* **Tìm kiếm hình ảnh: (các hình ảnh nhóm tình được để phụt vụ cho việc làm game)**

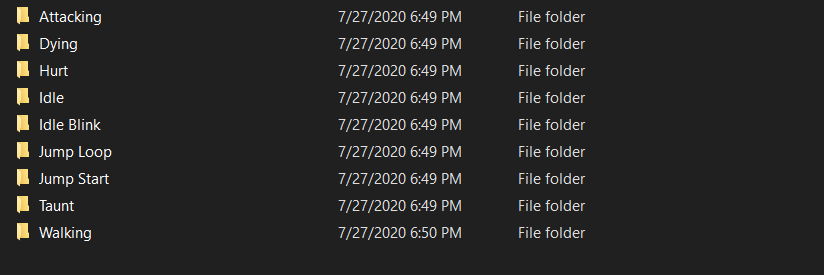


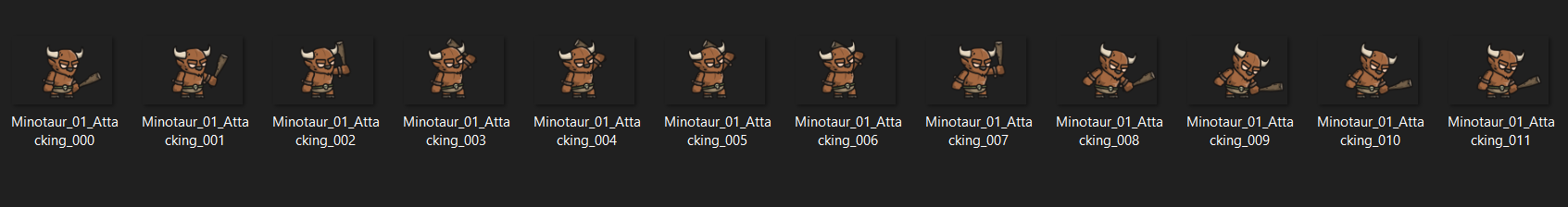




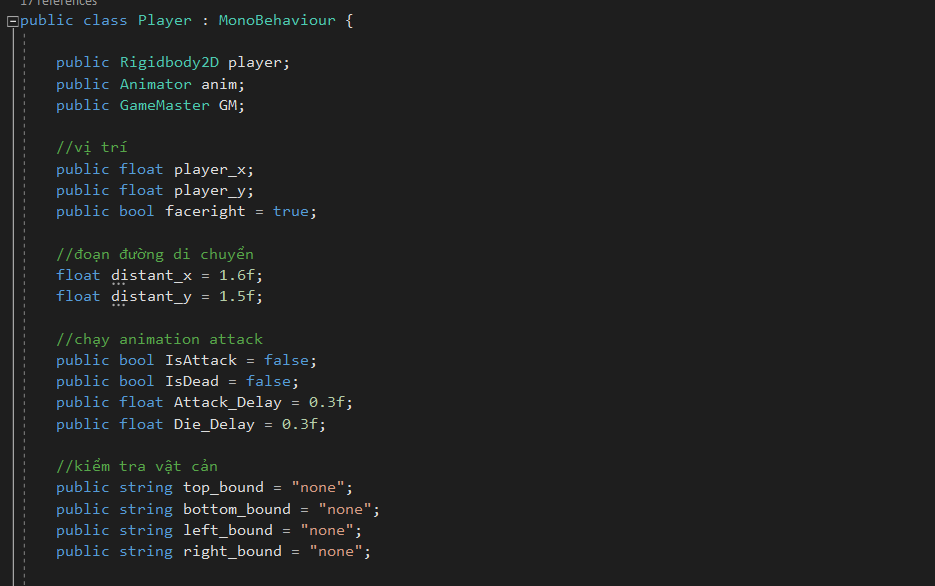




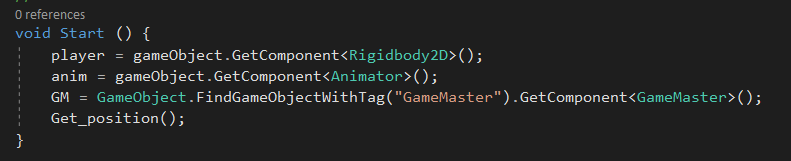




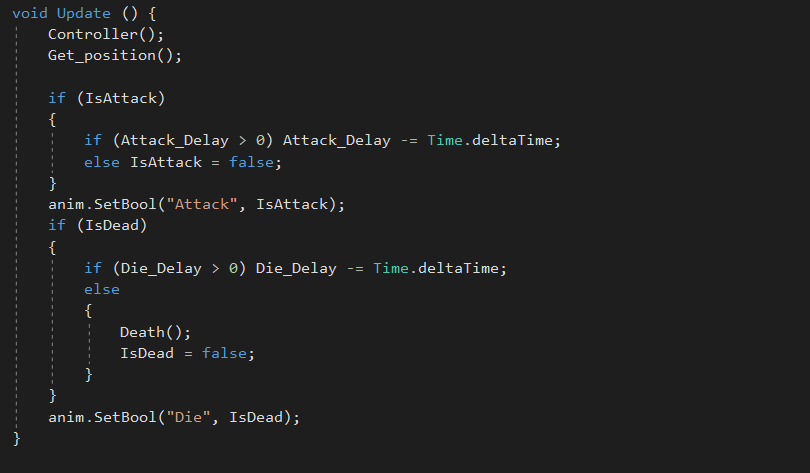
1. **Các class sử dụng trong đồ án**
   1. **Class trong folder Character**
      1. **Player**



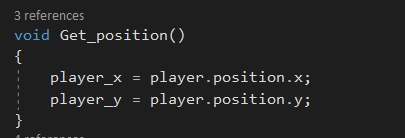
Các biến sử dụng



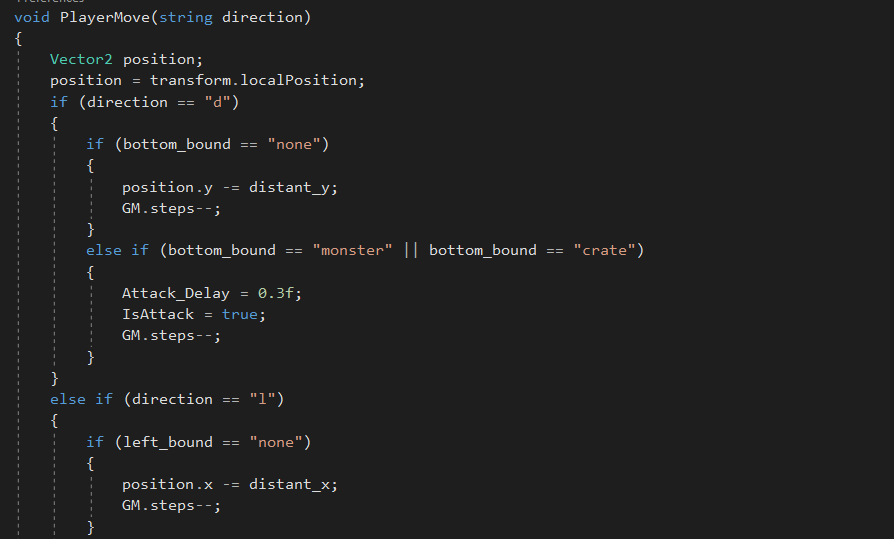
Hàm Start() khởi chạy 1 lần duy nhất khi bắt đầu màn game

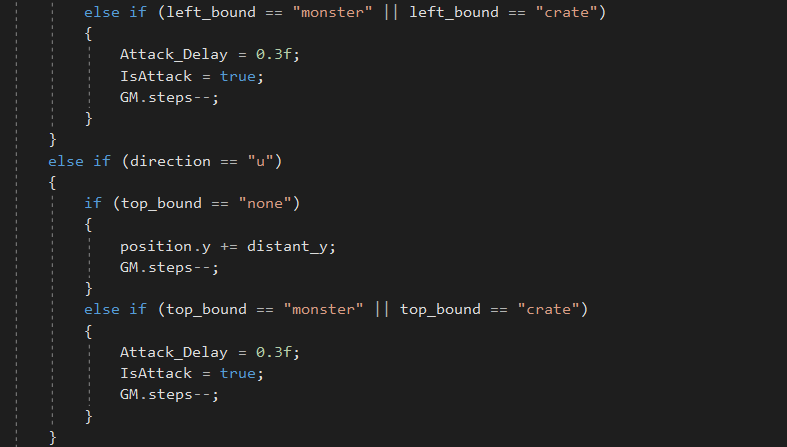


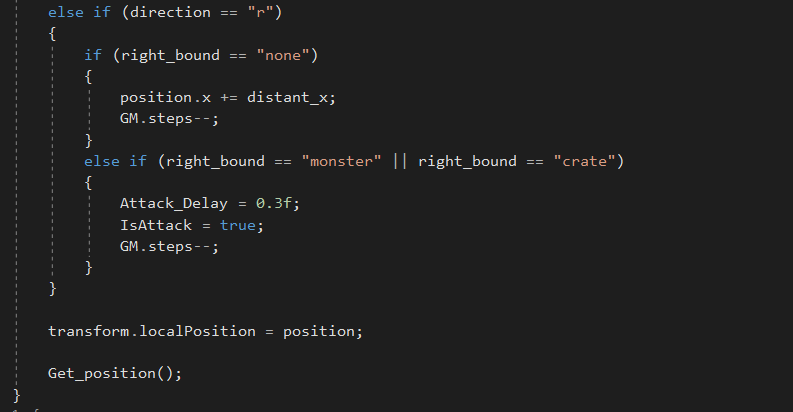
Hàm Update() được gọi liên tục trên từng frame



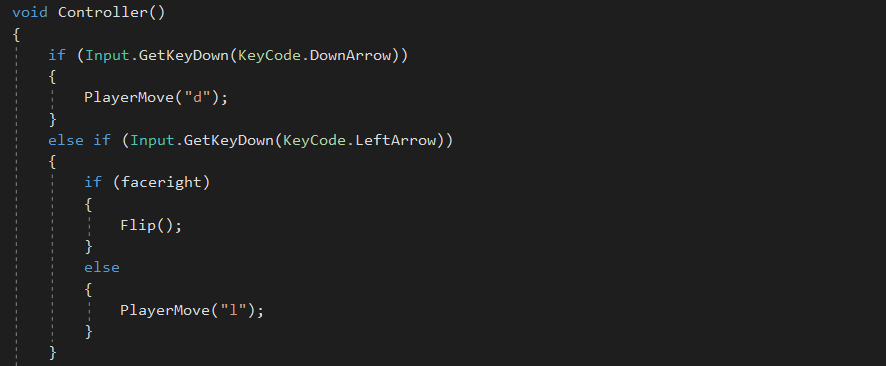
Hàm lấy vị trí Player

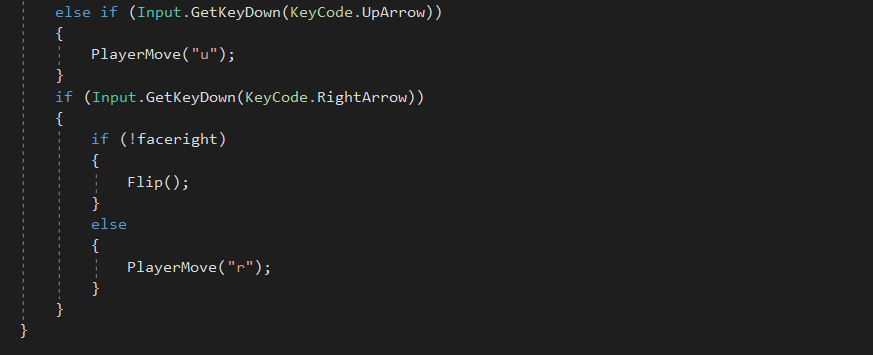






Hàm duy chuyển nhân vật



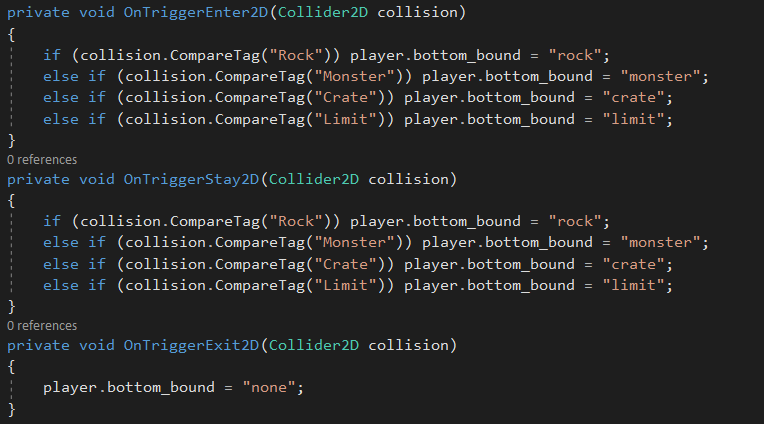


Hàm di chuyển nhân vật bằn bàn phím

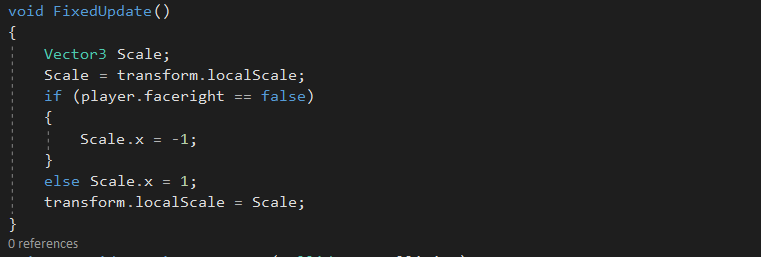


Hàm xoay nhân vật và xử lý khi nhân vật chết

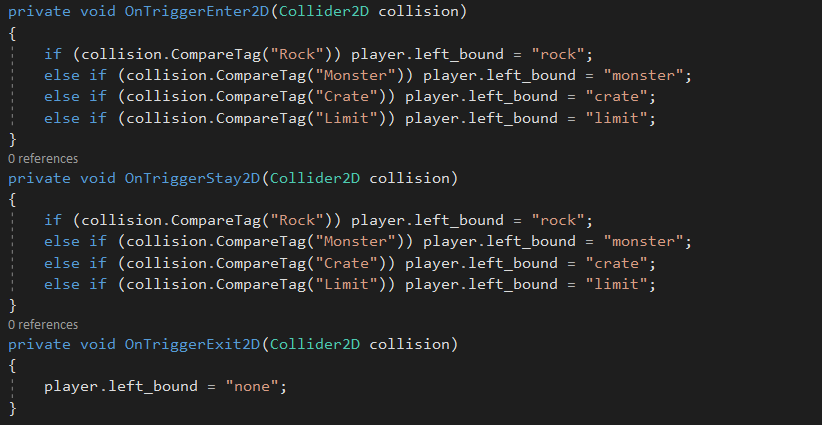
* + 1. **Bottom\_trigger**



* + 1. **Left\_trigger**

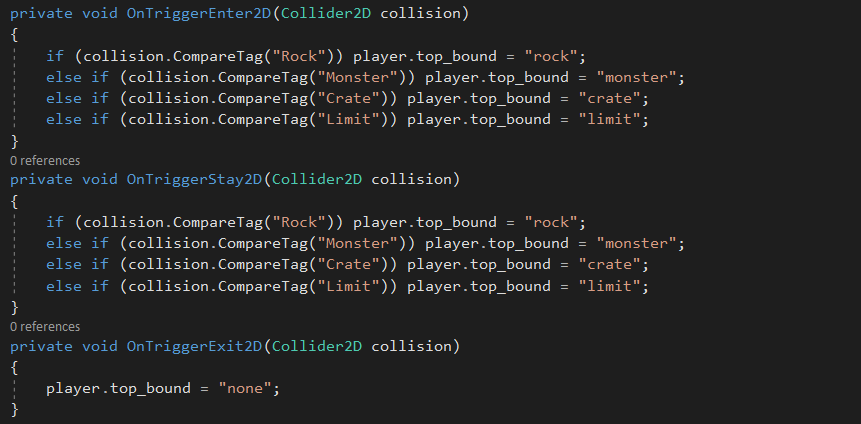


Hàm FixedUpdate() được gọi theo thời gian thực để đổi vị trí left\_trigger theo hướng xoay của nhân vật

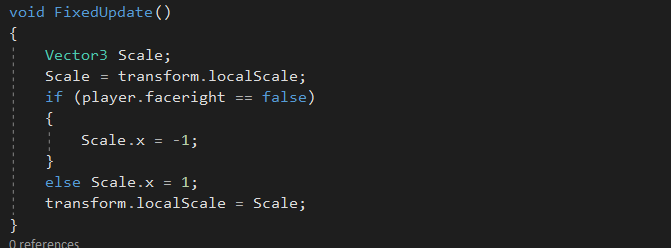


Xử lý va chạm với các vật thể khác

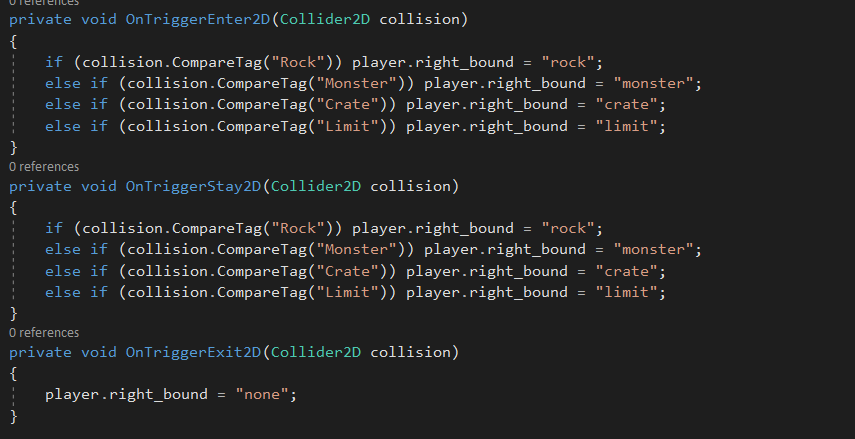
* + 1. **Top\_trigger**



* + 1. **Right\_trigger**

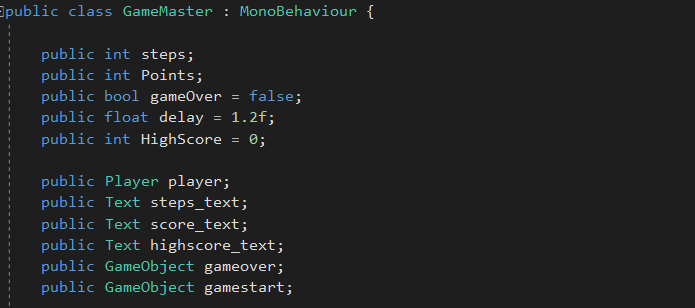


Hàm FixedUpdate() được gọi theo thời gian thực để đổi vị trí right\_trigger theo hướng xoay của nhân vật

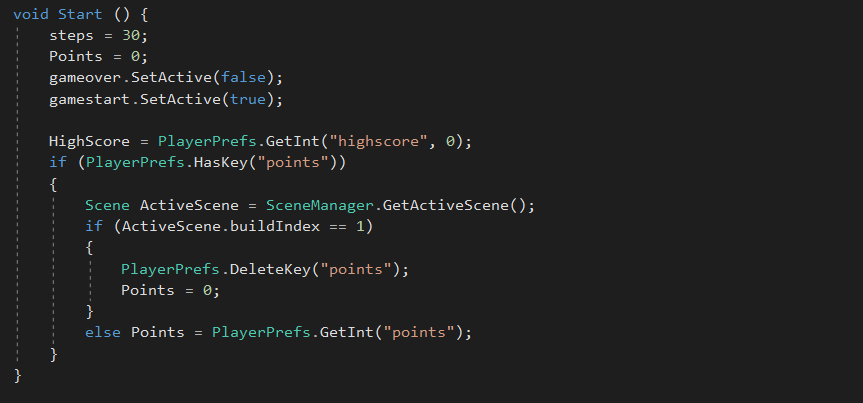


Xử lý va chạm với các vật thể khác

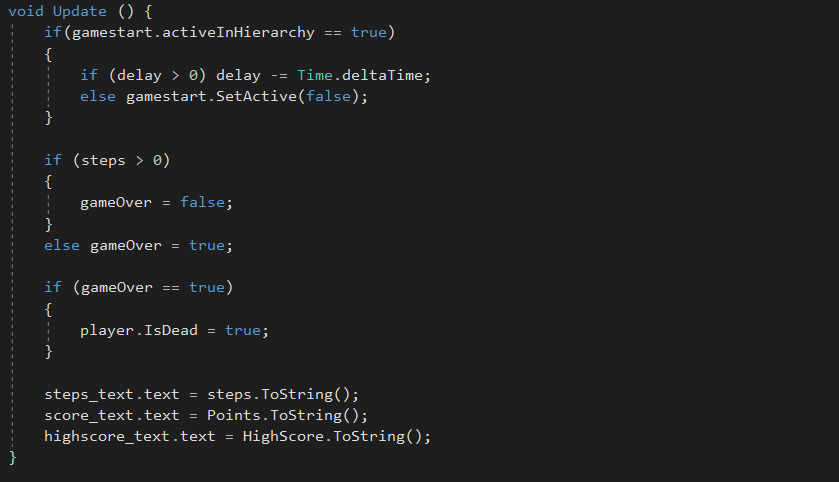
* 1. **Class trong folder Game**
     1. **GameMaster**



Các biến trong class GameMaster

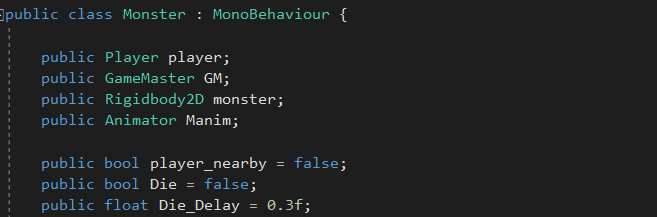


Hàm Start()

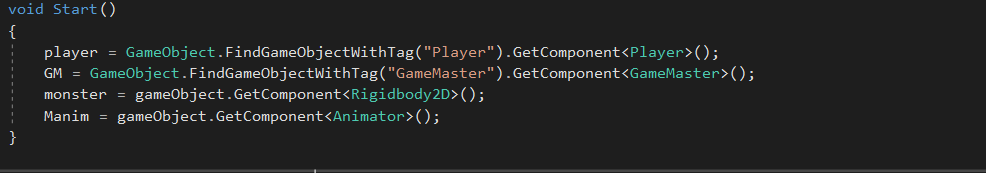


Hàm Update()

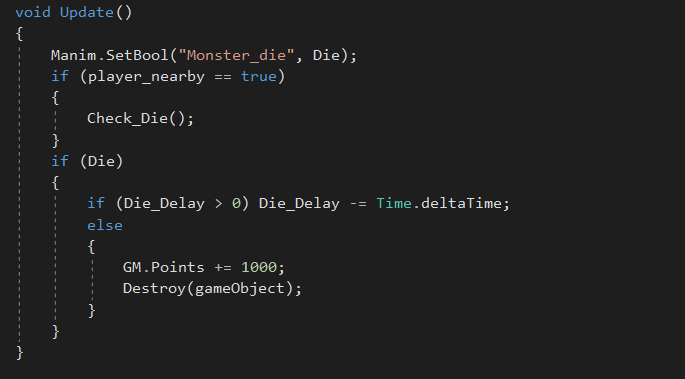
* + 1. **GameOverUI (note: chưa làm xong)**
  1. **Các Class trong folder Monster**
     1. **Monster**



Các biến trong class Monster



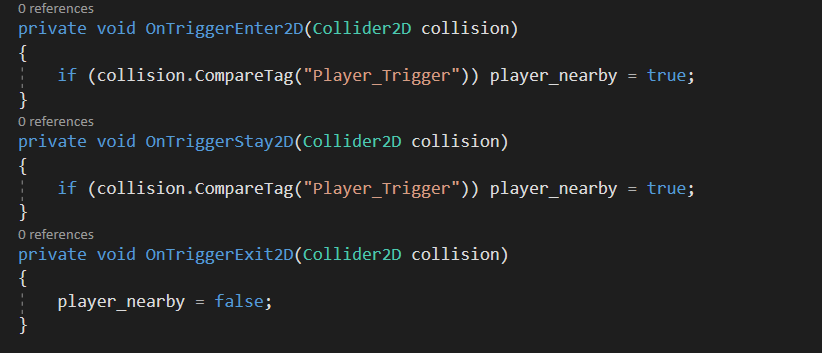
Hàm Start()



Hàm Update()

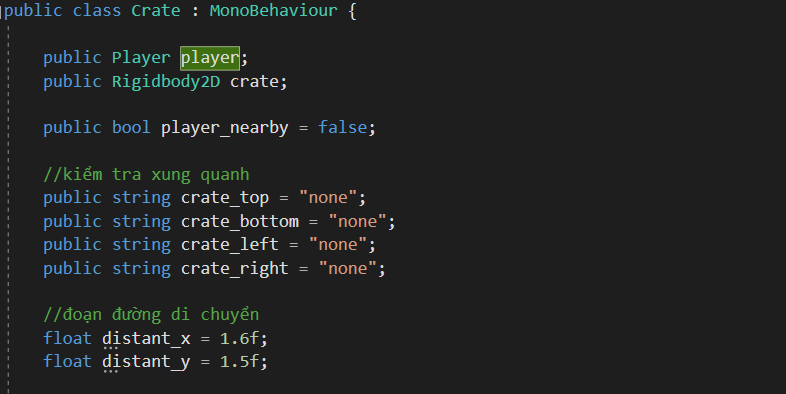


Hàm kiểm tra liệu quái có bị đánh chết



Hàm xử lý va chạm với trigger của Player

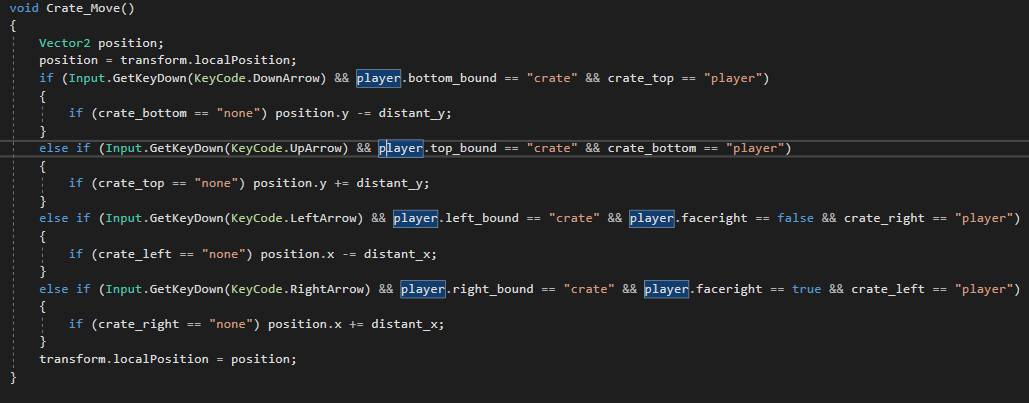
* 1. **Các class trong folder Objects**
     1. **Crate**



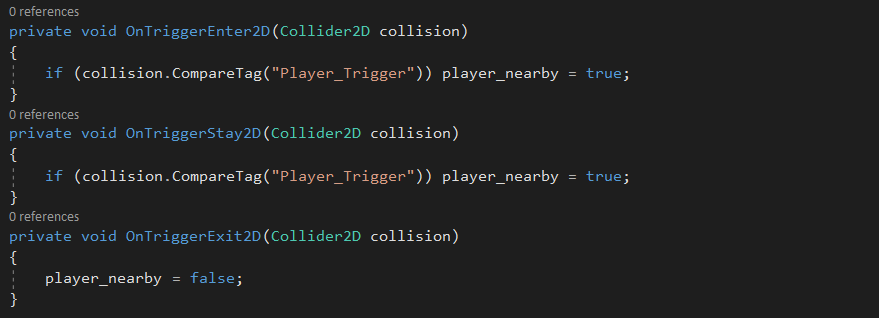
Các biến trong class Crate



Hàm Start() và Update()

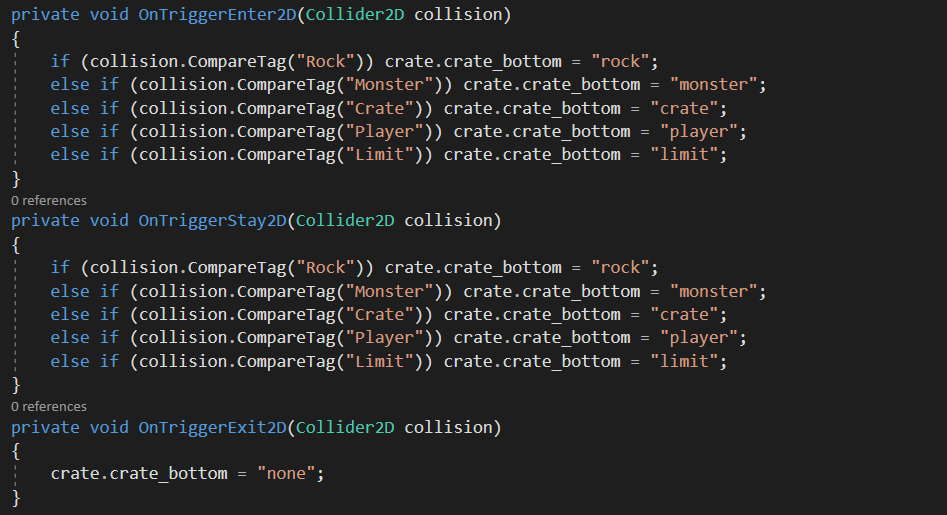


Hàm xử lý di chuyển cho Crate

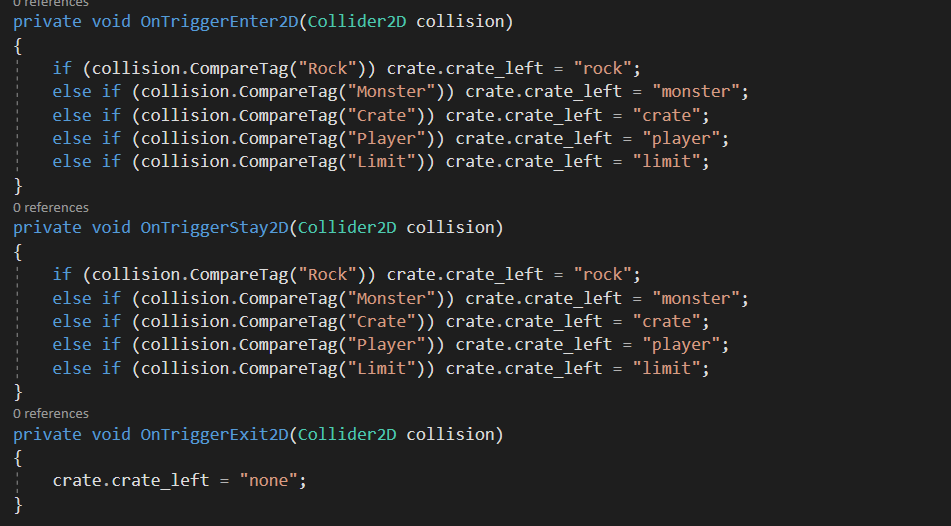


Hàm xử lý va chạm với trigger của Player

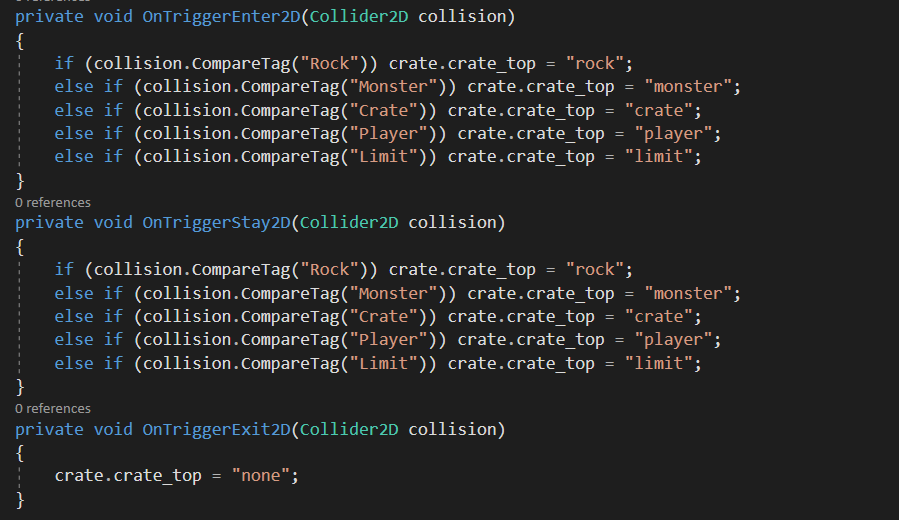
* + 1. **Crate\_bottom**



* + 1. **Crate\_left**



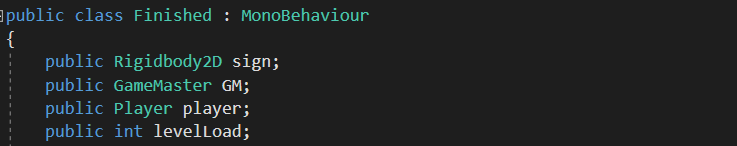
* + 1. **Crate\_top**



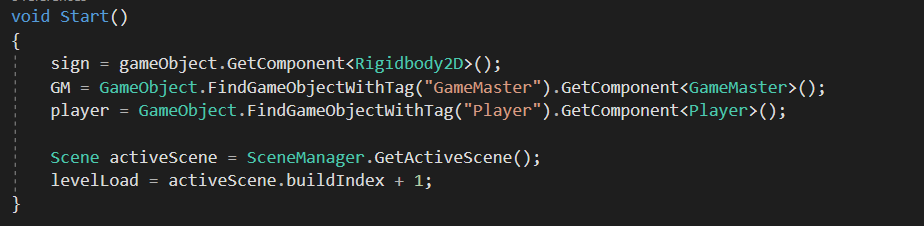
* + 1. **Crate\_right**



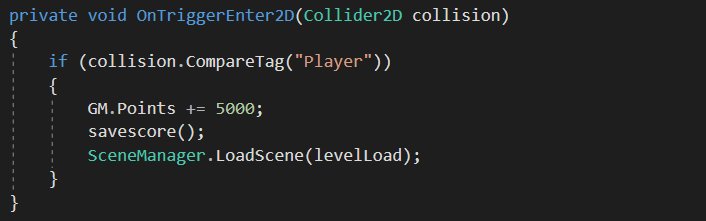
* + 1. **Finished**



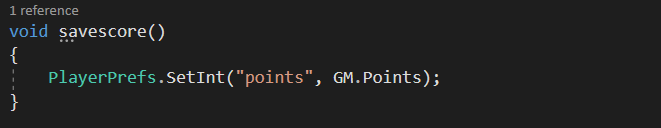
Các biến của class Finished



Hàm Start()



Hàm xử lý va chạm với Player



Hàm lưu điểm