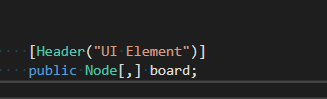
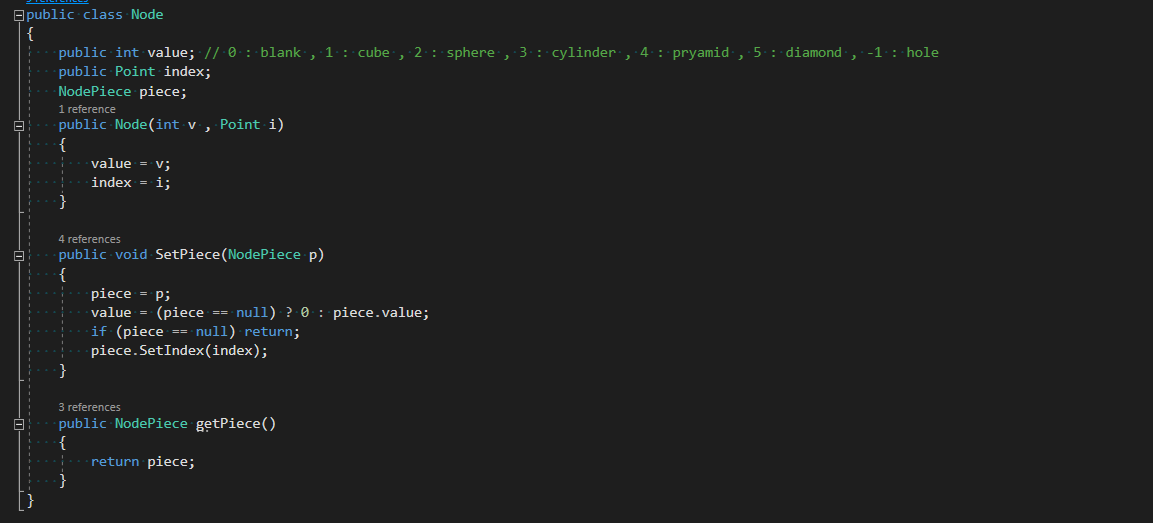
Game sẽ được làm trên canvas , do UI có dùng Rectransform có thể tùy ý chính sửa ảnh theo kích thước chuẩn

**Initialize :**

Tại mỗi block trong game là một node , tạo 1 mảng 2 node chiều cho game

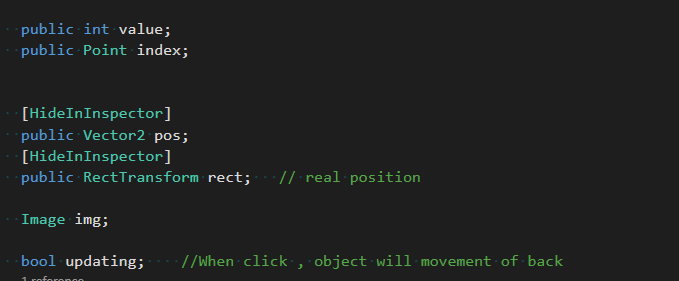


Với mỗi node có :



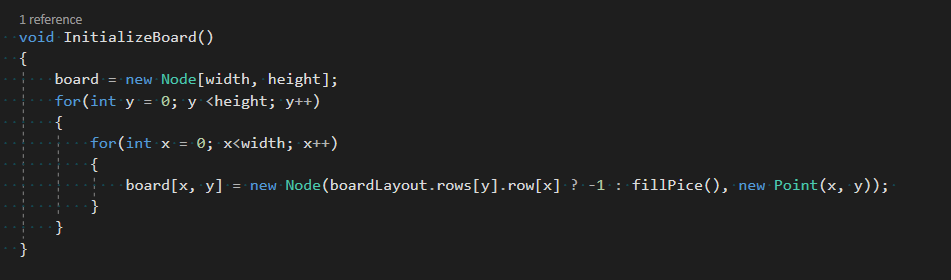
* Index : vị trí index của node trong mảng 2d
* Value : giá trị của node tại đó
* NodePiece : Một component do ta tạo ra , để ném vào mỗi object piece , component này sẽ quản lý trực tiếp object đó
* SetPiece : set cho nó mội piece mới
* getPiece : lấy ra component piece

**Component NodePiece :**

****

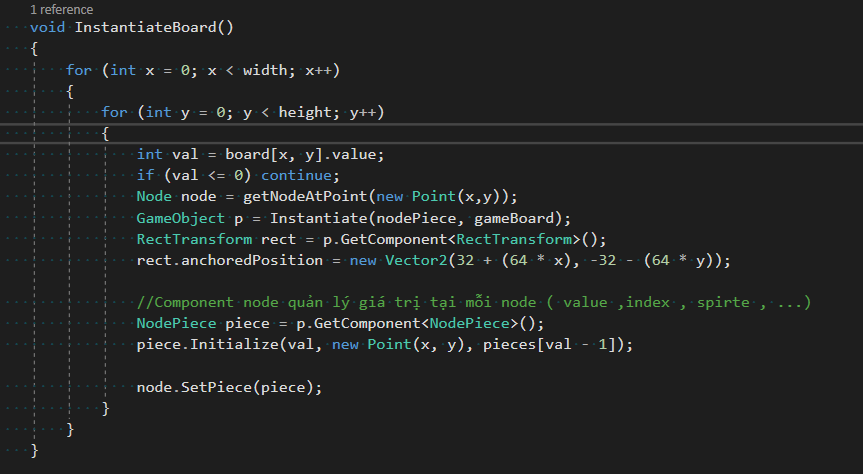
Do NodePiece la component quả lý object piece , cũng cần value và index , bool updating là biến quản lý vấn đề movement của object

**Render bảng ra màn hình game :**

****

*(khới tạo giá trị random cho board trước)*

Sau khi khởi tạo giá trị random , thì render object ra

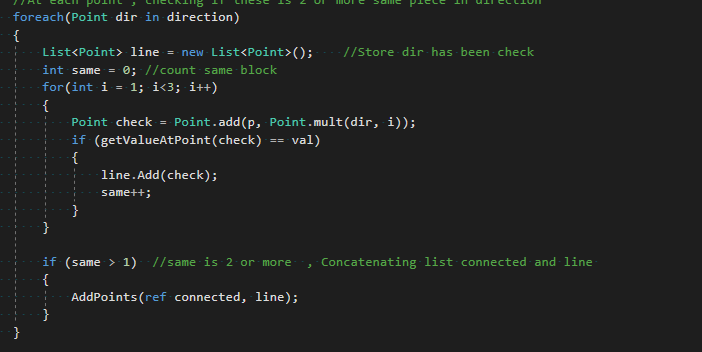


* Node node : lấy node tại vị trí x , y ra ( để gán NodePiece)
* P : một prefabs tạo sẵn , các object khác sẽ render dựa vào vị trí đầu tiên này
* Rect : lấy rectransform của p ra để gán lại tọa độ tính toán
* Lấy component nodepiece ra , initialize cho nó
* Sau cùng set nó lại vào node

**Thuật toán match : List<Point> isConnected(Point p , bool main)**

**Hàm trả về một list matching**

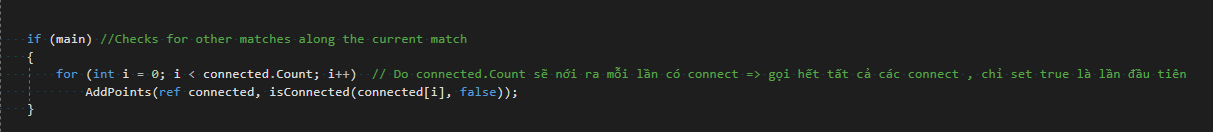
* Tạo một linklist **connected** , đánh dấu các node đã đủ điều kiện matching
* Foreach 4 hướng(up , left , down , right) . Tại mỗi hướng tạo một **List** để đánh dấu matching , sau đó for loop 2 lần ( đủ 3 để match )



* Như vậy khi có đủ 3 kết nối thì nó sẽ add vào connected , nhưng chưa đủ , nếu các piece sau cũng match thì sao , ví dụ : OOOOO

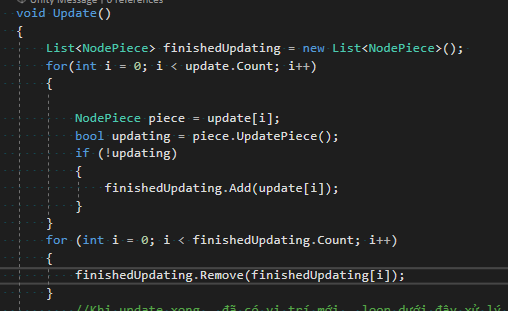
Khi thực hiện thuật toán như trên thì chỉ có 3 O đầu là đc đưa vào list

* Gọi đệ quy : tiếp tục gọi lại hàm như sau :

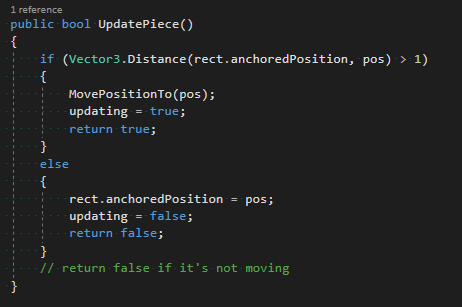


Đệ quy chỉ gọi 1 lần duy nhất , để gọi lại các piece khác , tại mỗi piece đã add vào connected nó sẽ kiểm tra matching và add vào connected 1 lần nữa , lúc này connected.Count sẽ phồng ra(tăng độ dài) và cứ thế kiểm tra cho đến hết các piece đc matching

**Di chuyển piece :**

****

Loop Update liên tục , khi tồn tại phần tử trong list update thì lập tức for loop sẽ thực hiện cập nhật updating = piece.UpdatePiece()

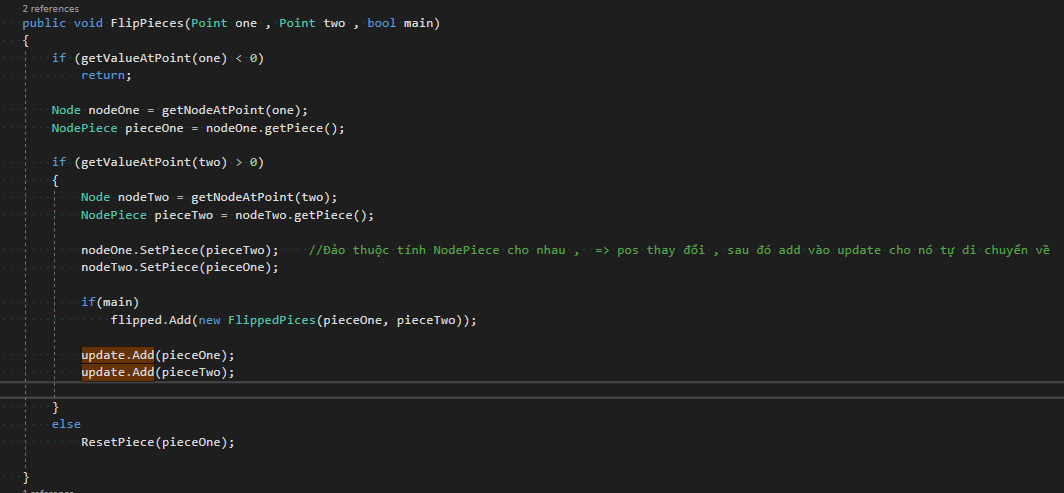


Hàm này kiểm tra nếu chi sai lệch 1 pixel ( tức là đã có tác động di chuyển thì nó sẽ lerp tới pos ) cho đến khi pixel ( khoảng cách ) < 1 thì return flase kết hợp function Update.

* Chỉ cần trong list update có phần tử thì updatePiece sẽ đc thực hiện và nếu chưa đến đc pos thì return true , liên tục di chuyển cho đến khi đến đc vị trí thì remove

**TAO TÁC Flip ( add vào update để di chuyển )**

Để add NodePiece vào update -> Khi ta flip



Thao tác flip thực hiện khi vừa buông chuột ra , nó sẽ trao đổi 2 thuộc tính NodePiece , bao gồm pos cũ , value , nhưng index thì không đổi (tức là node sẽ mang thằng NodePiece xuống làm component của nó và goi method SetPiece trong đó có SetIndex ( để cập nhật pos mới , sau đó update loop liên tục để nó quay về vị trí pos)

Cho nên khi list update có phần tử , nó if(Distance > 1) => Quay về vị trí ban đầu.