Tui ko cần biết mấy ông hiện thực trong MapLocation như thế nào, chỉ cần đưa ra đủ/ít nhất đống hàm public như sau:

* CCPoint[] getNextPoints(CCPoint currentPoint, int step) : cờ đang ở vị trí currentPoint và được phép đi step bước, khi gọi hàm này sẽ nhận dc 1 array các điểm sắp đi qua(cụ thể là array này có step phần tử). Hàm này dùng để vẽ amination di chuyển qua từng vị trí, cũng là dùng để lightup đường đi đó lên

**// CCPoint\* getNextPoints(int type,CCPoint currentPoint, int step)**

**//co loai co bao moi chon step may buoc duoc, de xet truong hop no ve dich luon**

**//dau hieu ket thuc mang la ccp(-1,-1);**

**//type:**

**//1:quan vang**

**//2:xanh la**

**//3:quan do**

**//4:quan xanh da troi**

**//cai nay hoi lech voi team no cua may, nhung tao so sua anh huong den nhung phan truoc do nen xai tam vay**

* CCPoint getInitPoint(int teamNo, int unitNo): con cờ thuộc team số 0,1,2,3 (teamNo) và unitNo là 0,1,2,3, không phân biệt theo màu gì cả, vì tương lai sẽ cho chọn team (heo gà ngựa trâu). Hàm này trả về vị trí trong chuồng (tức là chưa ra quân) của con cờ số unitNo của team số teamNo(0,1,2,3 gì đó). Hàm này dùng để hiện thực các hàm bên AnimalUnit(bao gồm hàm ra quân và hàm die-chết)
* **//ok**
* CCPoint getStartPoint(int teamNo) : 1 team có 1 vị trí ra quân, hàm này trả về vị trí đó
* **//0:quan vang**
* **//1:quan xanh la**
* **//2:quan do**
* **//3:xanh da troi**
* CCPoint getFinishPoint(int teamNo, int step): 1 team có 6 vị trí về đích(biến step), hàm này trả về 1 trong các vị trí đó. Hàm này dùng để hiện thực hàm finish() bên animalUnit
* **//ham getNextPoints da xu ly luon truong hop no ve dich roi**
* **//chi so la tu 0-5 nha(step)**
* **//**
* Một số hàm amination khác(cái này tui sẽ viết, và đã viết 1 số bao gồm lightup-select)
* Int havingUnitOnWay(CCPoint location, int step):Kiểm tra chướng ngại vật trên đường cho vị trí location, tầm vực step bước tiếp theo. Trả về 0 nếu không có, trả về 1 nếu chướng ngại vật nằm ở vị trí cuối cùng (tức là đá được) trả về -1 nếu bị chặn không đi được
* Bool havingUnitOnInitWay(int teamNo): vị trí ra quân của team số teamNo có bị chặn ko, có thì đá cmn đi(true)

Vì yêu cầu như trên, nên ban đầu tui đã setup cho MapLocation có 3 array chính

* Array 54 vị trí : wayLocation : đây là 54 vị trí trên đường đi vòng tròn
* **// map xung quanh co 56 vi tri chu ko pai la 54**
* Array 16 vị trí: initLocation: 16 vị trí trong chuồng/chia đều 4 phe mỗi phe 4 cái.

Team số n, vị trí số m sẽ có vị trí tại initLocation[n\*4+m]

* Array 24 vị trí :finishLocation: 24 vị trí về đích/ chia đều 4 phe mỗi phe 6 cái,

Team số n, vị trí số m sẽ có vị trí tại finishLocation[n\*4+m]

Thêm 3 cái int array để check con cờ (>=0tức là có con cờ đang ở đó, =-1 tức là rỗng)nếu không dùng 3 array này thì phải dùng box2d để xác định va chạm rất là mệt 😐, còn con số >=0 là sao? Vì ban đầu ta init 16 con tại 16 vị trí trong cái array 16 vị trí initLocation kia, như vậy cái array 16 int ban đầu sẽ là

[0,1,2,…..,14,15]

Array 54:

[-1,-1,-1,-1,…..-1,-1]

Array 24

[-1,-1,-1,….-1]

Coi như ta đánh số cho các con cờ = số thứ tự của nó trong array init luôn, sau đó con nào di chuyển tới đâu thì đánh số nó tới đó, ví dụ con cờ số 13 ra quân tại vị trí 0 của wayLocation thi 3 cái array trên theo thứ tự sẽ là

[0,1,2,3,4…..,12,-1,14,15]

[13,-1,-1,…-1]

[-1,-1,….-1]

Đó là cách tui định làm, để cho đơn giản và dễ đọc, mấy ông làm cách khác thì cũng được, nhưng phải cung cấp mấy hàm cơ bản để bên class khác gọi lấy kết quả, hiện tại bên MapLocation hiện thực quá mù mờ không ai rảnh để đi đọc code

1 số hàm để hỗ trợ game

* canBorn(k1,k2) : kết quả xúc xắc có thể ra quân dc ko ?
* canContinueRoll(k1,k2):kết quả xúc xắc có thể dc thêm lượt tiếp hay không
* …..