棋盘：

====================

0 N-----------N

|\ /

| \ /

| \ /

1 | X ( )

| / \

| / \

|/ \

2 R-----------R

0 1 2

====================

这是一个3行，3列的网格。

一个bug

D: QiPan::findN

0 1 2

0 N - R

1 | R #

2 N - x

move N.

0 1 2

0 x - R

1 | R #

2 N – N

代码在移动N的时候，将0，0位置的N移动到2，2

新建了一个叫path的类，来定义，哪个棋子可以走到哪个位置。来限制N从00 位置跑到22位置。所以，如何实现呢？

Path类：

*def* \_\_init\_\_(*self*,*pos* = (0,0), *friends*= {(0,0)}, *num*=0, *way*=1):

        self.\_number = num

        self.\_way = way

        self.\_actPoint = pos

        self.\_friends = friends

想法：在走N的时候，

        self.\_xPos = (1,1)

        self.\_nPos = (0,0)

有n的点 self.\_nPos

先拿到x点，self.\_xPos

再去检查Path的friends点，

如果点x的位置在friends里，才可以走到这个x点。

否则不走这个点。

将原来的点置x，新点（原x点）置N。

二，找到能走的N，而不是随便找一个。

更新QiPan里的findN方法，

在map里循环找N，

如果返回值为N。

检查值为这个点的path的friends里是否有x。

有就返回这个值为N的点。

没有，就继续找N。

找不到是N，并且，friends里有x坐标的情况。

判负。

三，如何让它智能起来？？？

有选择的挪动N。