本周我在GitHub上看到了一个有趣的人工智能小游戏，来自MIT，下面是我翻译的简介，以及链接和压缩包。<https://github.com/Zolomon/reversi-ai>。更详细的内容我还在进一步的学习之中。



reversi-AI是游戏reversi的实现，你可以在游戏中与计算机对战，或者观看两个人工智能相互对抗。

游戏是在python ( 3 . x)中实现的，由几个不同的组件组成。

在顶部是reversi.py文件，其中包含开始应用程序的常用“main”函数。这需要注意解析命令行参数和构造处理应用程序特定部分的游戏对象。

游戏类存在于游戏/game.py文件中。游戏对象创建了作为游戏的前端使用的板对象。游戏还初始化游戏所需的值。游戏对象的本质是其运行函数，它模仿普通的repl ( read - eval - print - loop)。它将呈现当前游戏状态，要求用户输入，处理输入，然后从第一步重复。

游戏对象保存关于两个控制器形状中的玩家的信息。控制器(在游戏/controllers.py中)可以是playercontroller或aicontroller。playercontroller将通过控制台解析用户输入到有效的move命令，或ctrl +d退出游戏。aicontroller将解析当前状态并将该信息传递给大脑(在游戏/ brain.py中)，然后在第二个线程中创建alphabetapruner (在游戏/ai.py中)的实例。

这允许用户在大脑执行其计算时轻松输出额外的信息或捕获用户输入。但是，只执行简单的输出来表示应用程序没有崩溃。

alphabetapruner ( ABP )是带有alpha -beta剪枝优化的极小极大算法的实现。ABP是使用一个整数列表来表示董事会，一个更优化的方法是只使用三个64位的整数和按位运算符，但是这是我想到的第一个表示。

在ABP的实施中，最有趣的功能是评价功能。这实际上可以被认为是整个人工智能的大脑，这里是实际状态评估的地方。使用下列评价；玩家是否有比对手更多的砖块，有多少玩家的砖块在边缘(边缘砖是重要的以控制对手的砖块的翻转)，有多少玩家砖块在角落，也非常重要的战略上。我发现维基百科有更重要的启发法，但没有足够的时间来描述它们。

游戏实例具有一个board对象，其中包含一个具有内部状态的块对象列表，该列表对象代表不同类型的可能的块(板、移动、白色、黑色)。在单色或256色RGB颜色中，如果-彩色命令行标志设置，则每片被绘制。

游戏通过给有效移动的x，y坐标的形状输入输入，使用x∈( a - h )，y∈( 1，8 )的公共格式，例如:D3、h1、a2、F8、H7等。

游戏结束时，不再向左移动，获胜者将由一计数砖块(更好的砖)来决定。