Game AI Design of Gomoku

09016425 隋越

09016493 刘云棣

Week 3

Report 1

Beta

1. 项目简介

设计一个五子棋游戏 AI，能够实现人机对战，并同其他的 AI 进行对战。具体内容有：

(1) 了解五子棋的游戏规则、棋型和获胜方法。

(2) 先制作一个较为简单的 AI，在测试和对战中对其进行完善和强化。

(3) 主要使用 C++进行程序编写，必要时也可使用其他语言。

2. 当前进度

(1) Week 3（截止至 3 月 18 日）进度

① 研究了五子棋的规则、棋型，并总结出了 7 种最常见的基本棋型——连五、活四、冲四、活三、眠三、活二、眠二。

② 通过评估函数来对棋盘上的位置打分。

③ 使用 Min-Max 搜索算法和 Alpha-Beta 剪枝原理来计算着点。

3. 详细说明

(1) Week 3

① 规则与棋型

暂时不考虑禁手。各种棋型的特点如下：

连五：五颗同色棋子连成一线，即获胜情形。

活四：有两个点可以形成连五。仅靠防守无法阻止连五。

冲四：只有一个点可以形成连五。容易防守。

活三：可以形成活四。利于进攻，需谨慎防守。

眠三：可以形成冲四，防守优先级不高。

活二：能够形成活三，较多时利于进攻。

眠二：能够形成眠三。

打分将以以上各种棋型为基础。

② 棋型打分

根据五子棋游戏中的实际情况，再增加双冲四、冲四活三、双活三、活三眠三、双活二、单子这几种情形。

对这些棋型进行打分，如：

连五：100 分；

活四、双冲四、冲四活三：90 分；

双活三：80 分；

活三眠三：70 分；

冲四：60 分；

活三：50 分；

双活二：40 分；

眠三：30 分；

活二：20 分；

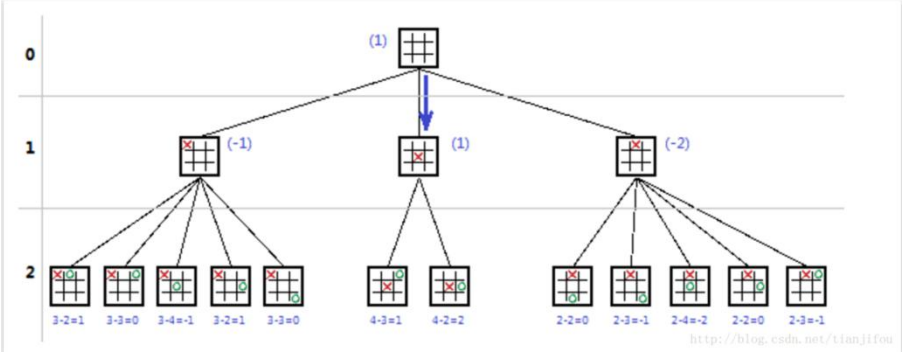
眠二：10 分；

单子：0 分。

再引入一个评估函数，用以上的打分机制来对棋盘位置进行打分。之后的搜索算法将以分值来确定最有利和最不利的着子位置。

③ 搜索算法

采用 Min-Max 搜索算法来确定最佳着子位置。 算法示意图如：



运行时要检查整个博弈树，因此效率较低。每次搜索更深一层时，树的大小就呈指数式增长。为了减少时间成本，采用 Alpha-Beta 剪枝原理来进行改进。

④ Alpha-Beta 剪枝原理

Alpha-Beta 剪枝原理旨在减少在其搜索树中，被 Min-Max 算法评估的节点数。这个思想是在搜索中传递两个值，第一个值是 Alpha，即搜索到的最好值，任何比它更小的值就没用了，因为策略就是知道 Alpha 的值，任何小于或等于 Alpha 的值都不会有所提高。

第二个值是 Beta，即对于对手来说最坏的值。这是对手所能承受的最坏的结果，因为我们知道在对手看来，他总是会找到一个对策不比 Beta 更坏的。如果搜索过程中返回 Beta 或比 Beta 更好的值，那就够好的了，走棋的一方就没有机会使用这种策略了。

在搜索着法时，每个搜索过的着法都返回跟 Alpha 和 Beta 有关的值，它们之间的关系非常重要，或许意味着搜索可以停止并返回。

如果某个着法的结果小于或等于 Alpha，那么它就是很差的着法，因此可以抛弃。由前文可知，在这个策略中，局面对走棋的一方来说是以 Alpha 为评价的。

如果某个着法的结果大于或等于 Beta，那么整个结点就作废了，因为对手不希望走到这个局面，而它有别的着法可以避免到达这个局面。因此如果我们找到的

评价大于或等于 Beta，就证明了这个结点是不会发生的，因此剩下的合理着法没有必要再搜索。

如果某个着法的结果大于 Alpha 但小于 Beta，那么这个着法就是走棋一方可以考虑走的，除非以后有所变化。因此 Alpha 会不断增加以反映新的情况。有时候可能一个合理着法也不超过 Alpha，这在实战中是经常发生的，此时这种局面是不予考虑的，因此为了避免这样的局面，我们必须在博弈树的上一个层局面选择另外一个着法。

（参考资料：http://www.xqbase.com/computer/search_alphabeta.htm）

4. 总结

(1) Week 3

在为时两周的学习中，我们主要以查阅资料为主，了解了五子棋相关的各种知识以及前辈们钻研出来的众多算法。对于一个棋盘游戏 AI 的设计，相当重要的一点就是要让 AI 认识到在不同情形下，各个位置着子对自己的胜利有多大帮助。因此，必须要根据五子棋游戏中的实际情况，为每个位置进行“打分”操作。这就涉及到相关的搜索算法，即如何找出最佳着子点。

为此，我们了解了一些搜索算法，并最终决定采用 Min-Max 搜索算法以及 Alpha-Beta 剪枝原理来确定最佳着子位置。

在接下来的两周时间里，我们计划设计出一个相对基础的 AI 来进行测试。这个 AI 将能与玩家进行对战。我们计划根据测试与对战中实际出现的问题来逐步完善对于 AI 的设计。