**对抗搜索算法的实验分析**

**实验零：补全代码，并将填空内容写在实验报告中**

alpha\_beta\_serach.hpp

// 更新alpha或者beta的值：

alpha = new\_value;

beta = new\_value;

// 剪枝

if ( beta < alpha )

break;

贴代码有某些行缩进和换行有问题，难以阅读，下面是代码截图，源代码见附代码



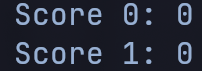
general\_game\_search.hpp

// ……

best\_cumulative\_rewards[ state.active\_player() ] < cumulative\_rewards[ state.active\_player() ]



运行截图：



两个终局状态：





monte\_carlo\_search.hpp

// ……

// TODO：子结点初始累计收益 values 为模拟得到的值——1行

values = simulate\_from( next\_state );

/\*\*\*\*\*\*TODO：选择UCT值最大的子结点继续探索\*\*\*\*\*/

selection.submit( value\_sums\_of[ node->index() ][ child ] / visit\_count\_of[ child ] + exploration \* sqrt( 2 \* log( visit\_count\_of[ node->index() ] ) / visit\_count\_of[ child ] ) );

/\*\*\*\*\*\*\* TODO： 按平均价值贪心选择（需要参考root\_state的接口）\*\*\*\*\*\*/

selection.submit( value\_sums\_of[ child ][ root\_state.active\_player() ] / visit\_count\_of[ child ] );