DemoWithoutGUI说明

1. SDK准备

将SDK包中的 “AIWZQDll.dll”、”AIWZQDll.lib”、”Inter.h”复制到工作目录下，或者添加到搜索路径中。

并将 “Inter.h”中的4个涉及GUI显示的函数删除或注释掉：SetPieceWithGUI、SetPieceByAIAndShow、DrawBoard、DrawPieces

1. 新建一个控制台应用程序，并添加以下两句代码来引入SDK

#include "Inter.h"

#pragma comment(lib, "AIWZQDll.lib")

1. 编写调用程序
2. Login("user", "password")：使用辅助工具里的“神经网络连子棋.exe”或 [www.gnxxkj.com](http://www.gnxxkj.com) 注册用户，并将用户名和密码填入此处。该接口在整个程序运行过程中仅需调用一次。
3. 初始化“确定性神经网络”——自主研发的一种国产神经网络：

InitWithoutModelFile(15, 15, 5)：没有模型的情况下使用该接口，此例初始化棋盘大小为15x15，五连子获胜（即五子棋），也可以初始化为其他的合理数据，e.g 19x19大小、六连子等；

InitFromModelFile("model.mod")：已有模型的情况下，可以使用“确定性神经网络”模型来初始化，如此处的 model.mod。

初始化接口在整个程序运行过程中仅需调用一次。

1. StartNewGame()：开始新游戏。该接口在每次开始新一局游戏的时候调用。
2. SetPieceByAI()：调用“确定性神经网络”进行预测并落子。

若要使用人为操作落子，则可调用 SetPieceWithCoord(int nX, int nY)接口并传入要落子的位置（位置从0开始索引，且不能超过棋盘大小）；

棋盘大小可以通过TBOARD\* GetBoardData(int\* pnBoardWidth, int\* pnBoardHeight)接口获得。

1. IsGameOver()用于判断游戏是否已结束：若已结束，则可调用GetWinner()接口来查看谁获胜（-1表示黑棋胜，1表示白棋胜，0表示平局）。
2. SaveModel("model.mod")；用于保存当前已经学习出的“确定性神经网络”模型，该模型在每局结束的时候会重新学习。
3. SaveSteps("data.txt")：用于保存本次对局的棋局数据，每一步落子为一行，每行包含落子位置（x和y）和落子人（-1或1）。
4. TBOARD\* pData = GetBoardData(&nBoardWidth, &nBoardHeight)：获取到当前棋盘数据，可用于显示或其它操作。此例中用于输出在控制台上以显示当前棋局状态。至此可运行程序进行测试。