## Utiliser la DLL *IPlayable* dans votre projet comme référence (ne pas retaper le code!)

Il est donné ici pour expliciter l'interface

/\* Interface pour tournoi d'IA sur Othello

\* Révision 3 (06.12.2018 OHU)

\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace IPlayable

{

public interface IPlayable

{

/// <summary>

/// Returns the IA's name

/// </summary>

/// <returns>true or false</returns>

**string GetName();**

/// <summary>

/// Returns true if the move is valid for specified color

/// </summary>

/// <param name="column">value between 0 and 8</param>

/// <param name="line">value between 0 and 6</param>

/// <param name="isWhite"></param>

/// <returns>true or false</returns>

**bool IsPlayable(int column, int line, bool isWhite);**

/// <summary>

/// Will update the board status if the move is valid and return true

/// Will return false otherwise (board is unchanged)

/// </summary>

/// <param name="column">value between 0 and 7</param>

/// <param name="line">value between 0 and 7</param>

/// <param name="isWhite">true for white move, false for black move</param>

/// <returns></returns>

**bool PlayMove(int column, int line, bool isWhite);**

/// <summary>

/// Asks the game engine next (valid) move given a game position

/// The board assumes following standard move notation:

///

/// A B C D E F G H I

/// [ ][0 1 2 3 4 5 6 7 8] (first index)

/// 1 0

/// 2 1

/// 3 2 X

/// 4 3 X

/// 5 4

/// 6 5

/// 7 6

///

/// Column Line

/// E.g.: D3, F4 game notation will map to {3,2} resp. {5,3}

/// </summary>

/// <param name="game">a 2D board with integer values: 0 for white 1 for black and -1 for empty tiles. First index for the column, second index for the line</param>

/// <param name="level">an integer value to set the level of the IA, 5 normally</param>

/// <param name="whiteTurn">true if white players turn, false otherwise</param>

/// <returns>The column and line indices. Will return {-1,-1} as PASS if no possible move </returns>

Tuple<int, int> GetNextMove(int[,] game, int level, bool whiteTurn);

/// <summary>

/// Returns a reference to a 2D array with the board status

/// </summary>

/// <returns>The 7x9 tiles status</returns>

**int[,] GetBoard();**

/// <summary>

/// Returns the number of white tiles on the board

/// </summary>

/// <returns></returns>

**int GetWhiteScore();**

/// <summary>

/// Returns the number of black tiles

/// </summary>

/// <returns></returns>

int GetBlackScore();

}

}