#ifndef MCTS\_H

#define MCTS\_H

#include "../../Base.h"

#include "../../DataStructures/DoublyLinkedList.c"

#include "../../DataStructures/StringFields.h"

#include "../../DataStructures/MMath.h"

//#include "../../Simulation/Games/TickTackToe/TickTackToe.h"

// Program Header Information ////////////////////////////////////////

/\*\*

\* @file DoublyLinkedList.h

\*

\* @brief Header file for DoublyLinkedList.c

\* instantiate's The Structues:

\* -DLL\_Node\_t (DLL short for DoublyLinkedList)

\* -DLL\_Handle\_t (DLL short for DoublyLinkedList)

\*

\* @details Specifies:

\* -PCB(Process Control Block), That keeps track of simulated Programs

\* That Run on The Simulated Operating System.

\* -SystemManagement, Keeps track of all information durring RunTime

\*

\*

\* @note None

\*/

typedef void \*(\*PayerMove)(void \*Board,void\* Move);

typedef struct GameRules\_t

{

RunTimeFunction InitializeWorld;

RunTimeFunction Player0\_Moves;

PayerMove Player0;

RunTimeFunction Player1\_Moves;

PayerMove Player1;

PayerMove PlayerMove;

RunTimeFunction ManageGame;

RunTimeFunction CopyGame;

RunTimeFunction MakeMove;

RunTimeFunction Winner;

RunTimeFunction RollOut;

RunTimeFunction FreeGame;

//Create Node

} GameRules\_t;

typedef struct MCTS\_Node\_t

{

//float RolloutValue;

float AverageValue;

float Value;

float EndValue;

int NodeVisits;

int End;

int Depth;

void\* Board;

int Player;

bool LeafNode;

GameRules\_t\* GameRules;

DLL\_Handle\_t\* ChildNodes; // To MCTS\_Node

} MCTS\_Node\_t;

typedef struct MCTS\_Handle\_t

{

int NodeVisits;

int NodesToSearch;

int NodesCreated;

int Player;

int PlayersMove;

IMatrix\_t\* PosibleMoves;

void\* RollOutGame;

GameRules\_t\* GameRules;

MCTS\_Node\_t\* TransversedNode;

MCTS\_Node\_t\* MCTS\_Node0;

} MCTS\_Handle\_t;

MCTS\_Node\_t\* Create\_MCTS\_Node(GameRules\_t\* GameRules,int Depth);

#define MCTS\_GameRules ((GameRules\_t\*)MCTS\_Handle->GameRules)

#define MCTS\_P0Move\_IMatrix(X) (IMatrix\_t\*)(MCTS\_Handle->GameRules->Player0\_Moves(X))

#define MCTS\_P0Move\_DLL(X) MCTS\_P0Move\_IMatrix(X)

//Verison

void MCTS\_Heap\_V();

float Find\_MCTS\_UCB1(MCTS\_Node\_t\* MCTS\_Node,int ParentVisitCount);

float MCTS\_Rollout(MCTS\_Handle\_t\* MCTS\_Handle);

void Select\_MCTS\_NodeFromDLL\_UCB1(MCTS\_Node\_t\*\* MCTS\_Node);

void CreateMCTSNode\_DLL(MCTS\_Handle\_t\* MCTS\_Handle);

MCTS\_Handle\_t\* Create\_MCTS\_Handle(int SearchDepth,RunTimeFunction GameRules);

void Free\_MCTS\_Handle(MCTS\_Handle\_t\* MCTS\_Handle);

void Free\_MCTS\_Node(MCTS\_Node\_t\* MCTS\_Node);

void Simulate\_MCTS(MCTS\_Handle\_t\* MCTS\_Handle,void\* Board);

float Transverse\_MCTS(MCTS\_Handle\_t\* MCTS\_Handle);

void MCTS\_Heap\_T();

#endif //MCTS\_H