Tic-Tac-Toe Project

v0.1

Generated by Doxygen 1.8.17

Chapter 1

Assignment 3 - Tic-Tac-Toe

JHU EN 605.604: Assignment 3 Author: Zhou, Yuchen (yzhou276 at jh dot edu)

1.1 Build

User can rebuild the main program using the following commands

```
mkdir build
cd build
cmake -S .. -B .
make
make install
```

1.2 Main Program

The main program is TicTacToeGame.cpp in the home directory

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 5

File Index

5.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 6

Namespace Documentation

6.1 TicTacToeDef Namespace Reference

Tic-Tac-Toe shared defintions.

Enumerations

- enum Marker : char { **EMPTY** = ' ', **O** = 'O', **X** = 'X' }

  Tic-Tac-Toe possible markers.
- enum Status { **O_WIN**, **X_WIN**, **DRAW**, **RUN** }

  Tic-Tac-Toe game status.

Variables

- const int BOARD_SIZE = 3

  Tic-Tac-Toe board size.
- const int MAXIMUM_PIECE = 9

  Tic-Tac-Toe maximum piece.

6.1.1 Detailed Description

Tic-Tac-Toe shared defintions.

Chapter 7

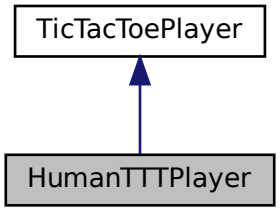
Class Documentation

7.1 HumanTTTPlayer Class Reference

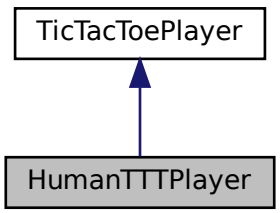
this class defines the movement of human Tic-Tac-Toe player

`#include <HumanTTTPlayer.h>`

Inheritance diagram for HumanTTTPlayer:



Collaboration diagram for HumanTTTPlayer:



Public Member Functions

- HumanTTTPlayer (const std::string &name="NoName", TicTacToeDef::Marker marker=TicTacToeDef::↩ Marker::X)

  Construct a new human Tic-Tac-Toe player object.
- std::pair< int, int > determineMove () const override

  prompt for the next move
- bool isHuman () const override

  return if the player is AI or human is declared by TicTacToePlayer

7.1.1 Detailed Description

this class defines the movement of human Tic-Tac-Toe player

Definition at line 21 of file HumanTTTPlayer.h.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 HumanTTTPlayer()

```
HumanTTTPlayer::HumanTTTPlayer (
            const std::string & name = "NoName",
            TicTacToeDef::Marker marker = TicTacToeDef::Marker::X )
```

Construct a new human Tic-Tac-Toe player object.

Parameters

name	registered player name
marker	player's marker

initializes a human Tic-Tac-Toe player's name and marker

Definition at line 20 of file HumanTTTPlayer.cpp.

7.1.3 Member Function Documentation

7.1.3.1 determineMove()

```
pair< int, int > HumanTTTPlayer::determineMove ( ) const  [override], [virtual]
```

prompt for the next move

Return values

std::pair<int,int>	coordinate

prompt the human player for next move's coordinate until the human player inputs a valid coordinate. The human player object does not have knowledge of the board, therefore this method is only responsible for getting a valid input from the human player.

Implements TicTacToePlayer.

Definition at line 33 of file HumanTTTPlayer.cpp.

7.1.3.2 isHuman()

```
bool HumanTTTPlayer::isHuman ( ) const  [override], [virtual]
```

return if the player is AI or human is declared by TicTacToePlayer

Return values

true	(always)

human Tic-Tac-Toe player should always return true

Implements TicTacToePlayer.

Definition at line 86 of file HumanTTTPlayer.cpp.

The documentation for this class was generated from the following files:

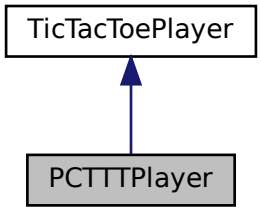
- include/HumanTTTPlayer.h
- src/HumanTTTPlayer.cpp

7.2 PCTTTPlayer Class Reference

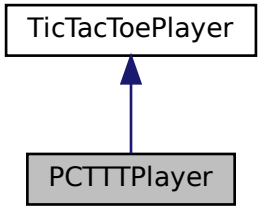
this class defines the movement of PC Tic-Tac-Toe player

```
#include <PCTTTPlayer.h>
```

Inheritance diagram for PCTTTPlayer:



Collaboration diagram for PCTTTPlayer:



Public Member Functions

- PCTTTPlayer (const std::string &name="NoName", TicTacToeDef::Marker marker=TicTacToeDef::Marker::X)
  Construct a new AI Tic-Tac-Toe player object.
- std::pair< int, int > determineMove () const override
  determine the next move (randomly chosen)
- bool isHuman () const override
  return if the player is AI or human is declared by TicTacToePlayer

7.2.1 Detailed Description

this class defines the movement of PC Tic-Tac-Toe player

Definition at line 21 of file PCTTTPlayer.h.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 PCTTTPlayer()

```
PCTTTPlayer::PCTTTPlayer (
            const std::string & name = "NoName",
            TicTacToeDef::Marker marker = TicTacToeDef::Marker::X )
```

Construct a new AI Tic-Tac-Toe player object.

Parameters

name	registered player name
marker	player's marker

initializes a PC Tic-Tac-Toe player's name and marker

Definition at line 19 of file PCTTTPlayer.cpp.

7.2.3 Member Function Documentation

7.2.3.1 determineMove()

```
pair< int, int > PCTTTPlayer::determineMove ( ) const  [override], [virtual]
```

determine the next move (randomly chosen)

Return values

std::pair<int,int>	coordinate

randomly generate a coordinate for the next move. The PC player object does not have knowledge of the board, therefore this method is only responsible for getting a valid input from the PC player.

Implements TicTacToePlayer.

Definition at line 31 of file PCTTTPlayer.cpp.

7.2.3.2 isHuman()

```
bool PCTTTPlayer::isHuman ( ) const  [override], [virtual]
```

return if the player is AI or human is declared by TicTacToePlayer

Return values

false	(always)

PC Tic-Tac-Toe player should always return false

Implements TicTacToePlayer.

Definition at line 44 of file PCTTTPlayer.cpp.

The documentation for this class was generated from the following files:

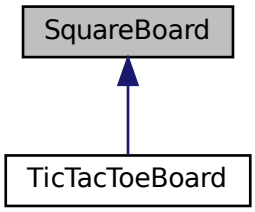
- include/PCTTTPlayer.h
- src/PCTTTPlayer.cpp

7.3 SquareBoard Class Reference

this class creates, manipulates, and manages a NxN square board

`#include <SquareBoard.h>`

Inheritance diagram for SquareBoard:



Public Member Functions

- SquareBoard (int size)

  Construct a new Square Board object.
- void cleanBoard ()

  This function resets all cells to empty (' ')
- void printBoard () const

  This function prints the board to console in default format.
- bool setCharacter (int row, int col, char character)

  This function sets cell at {row, col} to the input character.
- bool getCharacter (int row, int col, char &character) const

  This function passes the character in a cell by reference, then returns whether the input coordination is valid.
- int getBoardSize () const

  This function returns the size of the square board.
- const std::vector< std::vector< char > > & getBoard () const

  This function returns a constant reference to 2D square board variable.

Protected Member Functions

- void safeSetCharacter (int row, int col, char character)

  inline method to safely set a cell on the board
- bool isValidCoordinate (int row, int col) const

  inline method to valid the coordinate

Protected Attributes

- std::vector< std::vector< char > > board

  2D square board instance

7.3.1 Detailed Description

this class creates, manipulates, and manages a NxN square board

Definition at line 21 of file SquareBoard.h.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 SquareBoard()

```
SquareBoard::SquareBoard (
            int size )
```

Construct a new Square Board object.

Parameters

size	side length of the square

Construct a new NxN square board using 2D vectors. N is the size parameter user provided

Definition at line 22 of file SquareBoard.cpp.

7.3.3 Member Function Documentation

7.3.3.1 cleanBoard()

```
void SquareBoard::cleanBoard ( )
```

This function resets all cells to empty (' ')

cleans all cells of the square board

Definition at line 33 of file SquareBoard.cpp.

7.3.3.2 getBoard()

```
const vector< vector< char > > & SquareBoard::getBoard ( ) const
```

This function returns a constant reference to 2D square board variable.

Return values

const	vector<vector<char>>& 2D char vector

returns a constant reference to the 2D square board variable. The return reference is constant to prevent modification.

Definition at line 124 of file SquareBoard.cpp.

7.3.3.3 getBoardSize()

```
int SquareBoard::getBoardSize ( ) const
```

This function returns the size of the square board.

Return values

int	size of the square board

Definition at line 44 of file SquareBoard.cpp.

7.3.3.4 getCharacter()

```
bool SquareBoard::getCharacter (
            int row,
            int col,
            char & character ) const
```

This function passes the character in a cell by reference, then returns whether the input coordination is valid.

Parameters

row	row coordinate. Valid range: [0, boardSize]
col	column coordinate. Valid range: [0, boardSize]
character	character variable reference

Return values

true	if the coordinate is valid
false	if the coordinate is invalid

passes the character in the cell at input coordinate by the char reference. The reference will not be modified if the coordinate is illegal and therefore return false

Definition at line 108 of file SquareBoard.cpp.

7.3.3.5 isValidCoordinate()

```
bool SquareBoard::isValidCoordinate (
            int row,
            int col ) const  [inline], [protected]
```

inline method to valid the coordinate

Parameters

row	row coordinate
col	column coordinate

Return values

true	if the coordinate is legal
false	if the coordinate is illegal

Definition at line 105 of file SquareBoard.h.

7.3.3.6 printBoard()

```
void SquareBoard::printBoard ( ) const
```

This function prints the board to console in default format.

prints the square board in a specific format

Definition at line 53 of file SquareBoard.cpp.

7.3.3.7 safeSetCharacter()

```
void SquareBoard::safeSetCharacter (
            int row,
            int col,
            char character )  [inline], [protected]
```

inline method to safely set a cell on the board

Parameters

row	row coordinate
col	column coordinate
character	character to set to

Definition at line 92 of file SquareBoard.h.

7.3.3.8 setCharacter()

```
bool SquareBoard::setCharacter (
            int row,
            int col,
            char character )
```

This function sets cell at {row, col} to the input character.

Parameters

row	row coordinate. Valid range: [0, boardSize]
col	column coordinate. Valid range: [0, boardSize]
character	character to set to

Return values

true	if the coordinate is valid and set the cell
false	if the coordinate is invalid and does not set the cell

sets the cell at input coordinate to character. Returns false if the input coordinate is invalid and therefore does not update the board

Definition at line 91 of file SquareBoard.cpp.

The documentation for this class was generated from the following files:

- include/SquareBoard.h
- src/SquareBoard.cpp

7.4 TicTacToe Class Reference

This class create the top-level Tic-Tac-Toe game flow using TicTacToePlayer and TicTacToeBoard.

`#include <TicTacToe.h>`

Public Types

- enum GameMode { PVP = 1, PVE = 2, EVE = 3 }

  Tic-Tac-Toe game mode.

Public Member Functions

- TicTacToe ()

  Construct a new Tic Tac Toe object for PVP mode.
- TicTacToe (const std::string &humanPlayer1)

  Construct a new Tic Tac Toe object for PVE mode.
- TicTacToe (const std::string &humanPlayer1, const std::string &humanPlayer2)

  Construct a new Tic Tac Toe object for EVE mode.
- ~TicTacToe ()

  Destroy the Tic Tac Toe object.
- void startGame ()

  Start the Tic-Tac-Toe game engine.

7.4.1 Detailed Description

This class create the top-level Tic-Tac-Toe game flow using TicTacToePlayer and TicTacToeBoard.

Definition at line 27 of file TicTacToe.h.

7.4.2 Member Enumeration Documentation

7.4.2.1 GameMode

`enum TicTacToe::GameMode`

Tic-Tac-Toe game mode.

Enumerator

PVP	Human vs. Human.
PVE	Human vs. Computer.
EVE	Computer vs. Computer.

Definition at line 34 of file TicTacToe.h.

7.4.3 Constructor & Destructor Documentation

7.4.3.1 TicTacToe() [1/3]

```
TicTacToe::TicTacToe ( )
```

Construct a new Tic Tac Toe object for PVP mode.

Construct a new Tic-Tac-Toe objet for PC vs. PC. Dynamically allocate two PCTTTPlayer and feed a seed into the random number generator

Definition at line 23 of file TicTacToe.cpp.

7.4.3.2 TicTacToe() [2/3]

```
TicTacToe::TicTacToe (
            const std::string & namePlayer1 )
```

Construct a new Tic Tac Toe object for PVE mode.

Parameters

humanPlayer1	human player's name

Construct a new Tic-Tac-Toe objet for human vs. PC. Dynamically allocate one PCTTTPlayer and one and one HumanTTTPlayer, then feed a seed into the random number generator

Definition at line 44 of file TicTacToe.cpp.

7.4.3.3 TicTacToe() [3/3]

```
TicTacToe::TicTacToe (
            const std::string & namePlayer1,
            const std::string & namePlayer2 )
```

Construct a new Tic Tac Toe object for EVE mode.

Parameters

humanPlayer1	human player 1's name
humanPlayer2	human player 2's name

Construct a new Tic-Tac-Toe objet for human vs. human. Dynamically allocate two HumanTTTPlayer

Definition at line 63 of file TicTacToe.cpp.

7.4.3.4 ~TicTacToe()

```
TicTacToe::~TicTacToe ( )
```

Destroy the Tic Tac Toe object.

Destroy the Tic-Tac-Toe object

Definition at line 166 of file TicTacToe.cpp.

7.4.4 Member Function Documentation

7.4.4.1 startGame()

```
void TicTacToe::startGame ( )
```

Start the Tic-Tac-Toe game engine.

This is the main game loop.

1. Players take turns, starting with 'X'. On their turn, each player chooses an empty cell on the board and places their symbol ('X' or 'O') in that cell. After each move, the game checks for a win by determining
2. if the current player has aligned three of their symbols horizontally, vertically, or diagonally.
3. If no player has won and the grid is full, the game is declared a draw.
4. If a player wins, they are declared the winner, and the game ends.
5. If the game is a draw, no winner is declared, and the game ends.

Definition at line 89 of file TicTacToe.cpp.

The documentation for this class was generated from the following files:

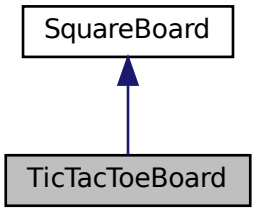
- include/TicTacToe.h
- src/TicTacToe.cpp

7.5 TicTacToeBoard Class Reference

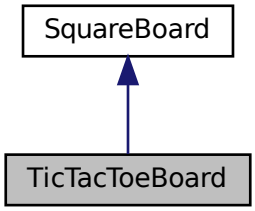
Represents a Tic-Tac-Toe board and inherits from SquareBoard.

`#include <TicTacToeBoard.h>`

Inheritance diagram for TicTacToeBoard:



Collaboration diagram for TicTacToeBoard:



Public Member Functions

- TicTacToeBoard ()
  Construct a new Tic-Tac-Toe Board object.
- bool makeMove (int row, int col, TicTacToeDef::Marker marker)
  place a new marker into the board
- void cleanBoard ()
  calls cleanBoard from SquareBoard and resets the number of pieces on the board
- int getNumPieceOnBoard () const
  gets the number of markers/cells occupied on the board
- bool checkWin (TicTacToeDef::Marker marker) const
  check if marker is winning
- bool checkDraw () const
  check if the board is draw
- bool isBoardFull () const
  checks if the board is full
- TicTacToeDef::Status getGameStatus () const
  Get the current status.

Additional Inherited Members

7.5.1 Detailed Description

Represents a Tic-Tac-Toe board and inherits from SquareBoard.

This class makes TTT move and checks the board status

Definition at line 24 of file TicTacToeBoard.h.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 TicTacToeBoard()

```
TicTacToeBoard::TicTacToeBoard ( )
```

Construct a new Tic-Tac-Toe Board object.

Construct a new Tic-Tac-Toe Board using the SquareBoard's constructor, then reset the number of markers on the board to 0

Definition at line 20 of file TicTacToeBoard.cpp.

7.5.3 Member Function Documentation

7.5.3.1 checkDraw()

```
bool TicTacToeBoard::checkDraw ( ) const
```

check if the board is draw

Return values

true	
false	

check if the game ties

Definition at line 116 of file TicTacToeBoard.cpp.

7.5.3.2 checkWin()

```
bool TicTacToeBoard::checkWin (
            TicTacToeDef::Marker marker ) const
```

check if marker is winning

Parameters

marker	(X, O, EMPTY)

Return values

true	
false	

check if the input marker is winning the game by checking if there are three cells of same marker in a row vertically, horizontally, or diagonally

Definition at line 80 of file TicTacToeBoard.cpp.

7.5.3.3 cleanBoard()

```
void TicTacToeBoard::cleanBoard ( )
```

calls cleanBoard from SquareBoard and resets the number of pieces on the board

Reset the Tic-Tac-Toe board and number of piece tracker

Definition at line 59 of file TicTacToeBoard.cpp.

7.5.3.4 getGameStatus()

```
TicTacToeDef::Status TicTacToeBoard::getGameStatus ( ) const
```

Get the current status.

Return values

TicTacToeDef::Status	(X_WIN, O_WIN, DRAW, IN)

get the status of the board. There are four possible status: O is winning, X is winning, there is a tie, and the game is still running

Definition at line 138 of file TicTacToeBoard.cpp.

7.5.3.5 getNumPieceOnBoard()

```
int TicTacToeBoard::getNumPieceOnBoard ( ) const
```

gets the number of markers/cells occupied on the board

Return values

int	number of markers on the board

Get the number of pieces on the board

Definition at line 69 of file TicTacToeBoard.cpp.

7.5.3.6 isBoardFull()

```
bool TicTacToeBoard::isBoardFull ( ) const
```

checks if the board is full

Return values

true	
false	

check if the Tic-Tac-Toe board is full

Definition at line 126 of file TicTacToeBoard.cpp.

7.5.3.7 makeMove()

```
bool TicTacToeBoard::makeMove (
            int row,
            int col,
            TicTacToeDef::Marker marker )
```

place a new marker into the board

Parameters

row	row coordinate. Valid range: [0, 2]
col	column coordinate. Valid range: [0, 2]
marker	(X, O, EMPTY)

Return values

true	if the movement is accepted by the board
false	if the movement is rejected by the board

place a new marker at the coordinate defined by row and col. Reject the movement if the coordinate is invalid to board, or if the cell is taken, or if the marker is EMPTY. Otherwise, accept the movement and update the board

Definition at line 33 of file TicTacToeBoard.cpp.

The documentation for this class was generated from the following files:

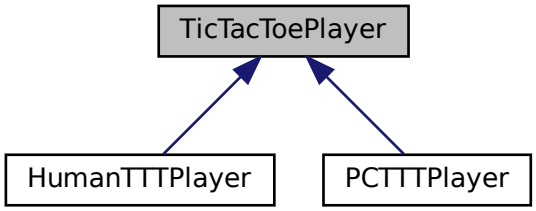
- include/TicTacToeBoard.h
- src/TicTacToeBoard.cpp

7.6 TicTacToePlayer Class Reference

This class registers the basic information of a Tic-Tac-Toe player and defines the prototypes of functions that inherited class need to implement.

`#include <TicTacToePlayer.h>`

Inheritance diagram for TicTacToePlayer:



Public Member Functions

- TicTacToePlayer (const std::string &name="NoName", TicTacToeDef::Marker marker=TicTacToeDef::↩ Marker::X)

  Construct a new Tic-Tac-Toe Player object.
- std::string getPlayerName () const

  Get player's name.
- TicTacToeDef::Marker getPlayerSymbol () const

  Get player's marker.
- virtual std::pair< int, int > determineMove () const =0

  virtual method to get the next move (must be implemented by derived classes)
- virtual bool isHuman () const =0

  return if the player is AI or human (must be implemented by derived classes)
- virtual ~TicTacToePlayer ()=default

  Destroy the Tic-Tac-Toe Player object.

7.6.1 Detailed Description

This class registers the basic information of a Tic-Tac-Toe player and defines the prototypes of functions that inherited class need to implement.

Definition at line 25 of file TicTacToePlayer.h.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 TicTacToePlayer()

```
TicTacToePlayer::TicTacToePlayer (
            const std::string & name = "NoName",
            TicTacToeDef::Marker marker = TicTacToeDef::Marker::X )
```

Construct a new Tic-Tac-Toe Player object.

Parameters

name	registered player name
marker	player's marker

initializes Tic-Tac-Toe player's name and marker

Definition at line 19 of file TicTacToePlayer.cpp.

7.6.2.2 ~TicTacToePlayer()

```
virtual TicTacToePlayer::~TicTacToePlayer ( )  [virtual], [default]
```

Destroy the Tic-Tac-Toe Player object.

7.6.3 Member Function Documentation

7.6.3.1 determineMove()

```
virtual std::pair<int, int> TicTacToePlayer::determineMove ( ) const  [pure virtual]
```

virtual method to get the next move (must be implemented by derived classes)

Return values

std::pair<int,int>	

Implemented in HumanTTTPlayer, and PCTTTPlayer.

7.6.3.2 getPlayerName()

```
string TicTacToePlayer::getPlayerName ( ) const
```

Get player's name.

Return values

std::string	player's name

return player's name

Definition at line 35 of file TicTacToePlayer.cpp.

7.6.3.3 getPlayerSymbol()

```
Marker TicTacToePlayer::getPlayerSymbol ( ) const
```

Get player's marker.

Return values

TicTacToeDef::Marker	player's marker

return player's marker

Definition at line 43 of file TicTacToePlayer.cpp.

7.6.3.4 isHuman()

```
virtual bool TicTacToePlayer::isHuman ( ) const  [pure virtual]
```

return if the player is AI or human (must be implemented by derived classes)

Return values

true	
false	

Implemented in HumanTTTPlayer, and PCTTTPlayer.

The documentation for this class was generated from the following files:

- include/TicTacToePlayer.h
- src/TicTacToePlayer.cpp

Chapter 8

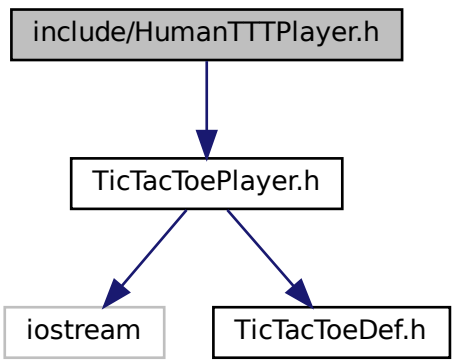
File Documentation

8.1 include/HumanTTTPlayer.h File Reference

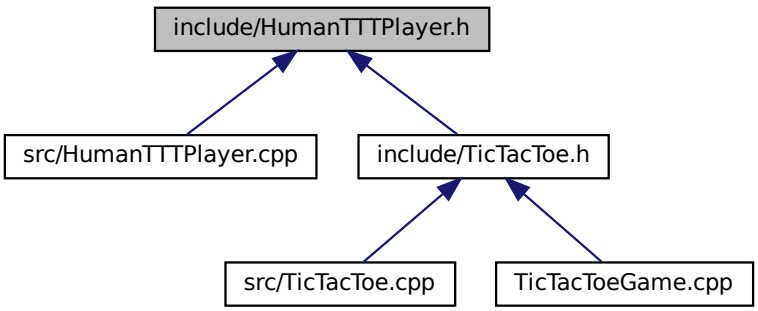
Human Tic-Tac-Toe player header file.

```
#include "TicTacToePlayer.h"
```

Include dependency graph for HumanTTTPlayer.h:



This graph shows which files directly or indirectly include this file:



Classes

- class HumanTTTPlayer

  this class defines the movement of human Tic-Tac-Toe player

8.1.1 Detailed Description

Human Tic-Tac-Toe player header file.

Author

Yuchen Zhou (yzhou276@jh.edu)

Version

0.1

Date

2024-09-14

Copyright

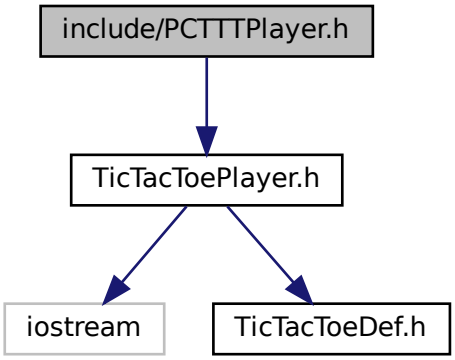
Copyright (c) 2024

8.2 include/PCTTTPlayer.h File Reference

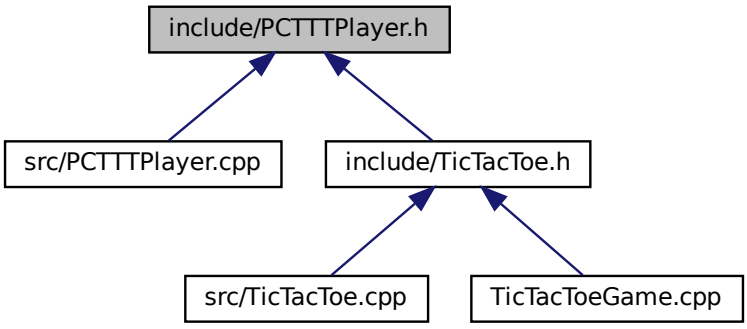
PC(AI) Tic-Tac-Toe player header file.

```
#include "TicTacToePlayer.h"
```

Include dependency graph for PCTTTPlayer.h:



This graph shows which files directly or indirectly include this file:



Classes

- class PCTTTPlayer

  this class defines the movement of PC Tic-Tac-Toe player

8.2.1 Detailed Description

PC(AI) Tic-Tac-Toe player header file.

Author

Yuchen Zhou (yzhou276@jh.edu)

Version

0.1

Date

2024-09-14

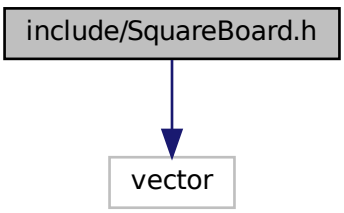
Copyright

8.3 include/SquareBoard.h File Reference

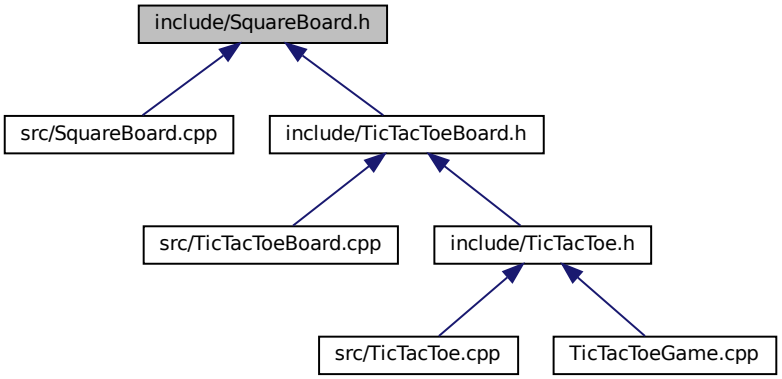
Square board header file.

```
#include <vector>
```

Include dependency graph for SquareBoard.h:



This graph shows which files directly or indirectly include this file:



Classes

- class SquareBoard

  this class creates, manipulates, and manages a NxN square board

8.3.1 Detailed Description

Square board header file.

Author

Yuchen Zhou (yzhou276@jh.edu)

Version

0.1

Date

2024-09-14

Copyright

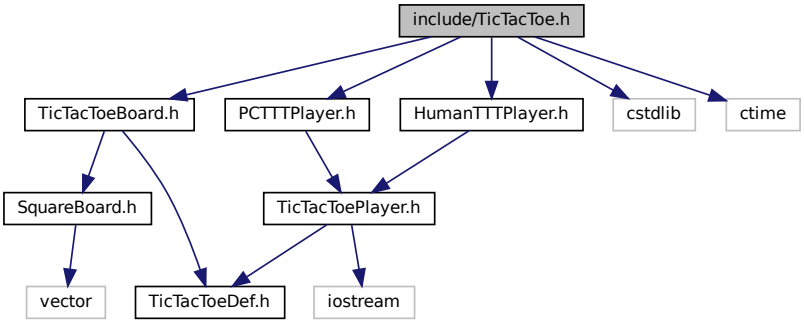
Copyright (c) 2024

8.4 include/TicTacToe.h File Reference

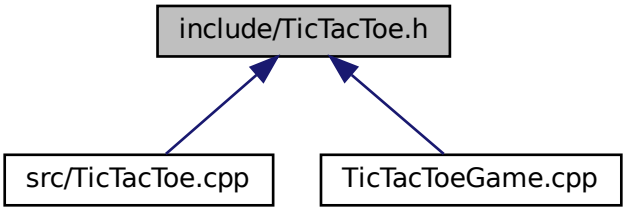
Tic-Tac-Toe top-level header file.

```
#include "TicTacToeBoard.h"
#include "PCTTTPlayer.h"
#include "HumanTTTPlayer.h"
#include <cstdlib>
#include <ctime>
```

Include dependency graph for TicTacToe.h:



This graph shows which files directly or indirectly include this file:



Classes

- class TicTacToe

  This class create the top-level Tic-Tac-Toe game flow using TicTacToePlayer and TicTacToeBoard.

8.4.1 Detailed Description

Tic-Tac-Toe top-level header file.

Author

Yuchen Zhou (yzhou276@jh.edu)

Version

0.1

Date

2024-09-14

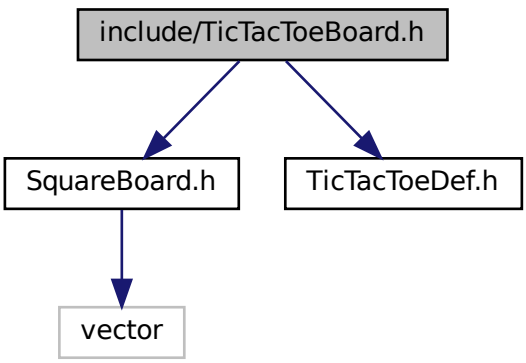
Copyright

8.5 include/TicTacToeBoard.h File Reference

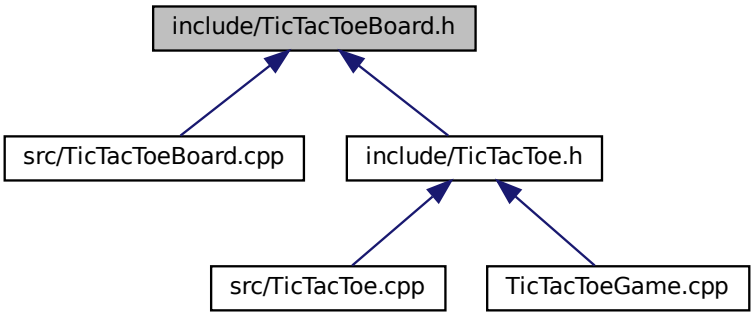
Tic-Tac-Toe board header file.

```
#include "SquareBoard.h"
#include "TicTacToeDef.h"
```

Include dependency graph for TicTacToeBoard.h:



This graph shows which files directly or indirectly include this file:



Classes

- class TicTacToeBoard

  Represents a Tic-Tac-Toe board and inherits from SquareBoard.

8.5.1 Detailed Description

Tic-Tac-Toe board header file.

Author

Yuchen Zhou (yzhou276@jh.edu)

Version

0.1

Date

2024-09-14

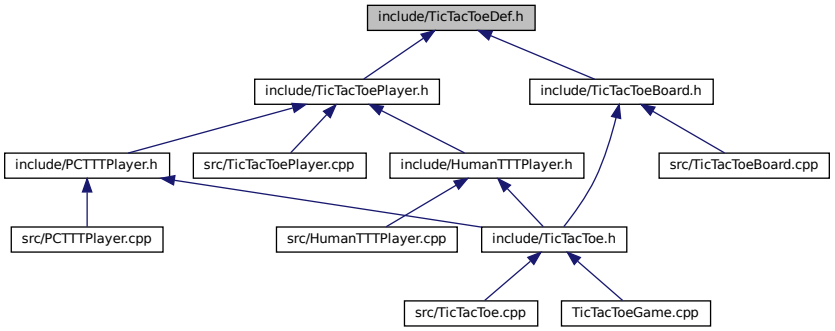
Copyright

Copyright (c) 2024

8.6 include/TicTacToeDef.h File Reference

Tic-Tac-Toe fixed definiton namespace.

This graph shows which files directly or indirectly include this file:



Namespaces

- TicTacToeDef

  Tic-Tac-Toe shared defintions.

Enumerations

- enum TicTacToeDef::Marker : char { **EMPTY** = ' ', **O** = 'O', **X** = 'X' }

  Tic-Tac-Toe possible markers.
- enum TicTacToeDef::Status { **O_WIN**, **X_WIN**, **DRAW**, **RUN** }

  Tic-Tac-Toe game status.

Variables

- const int TicTacToeDef::BOARD_SIZE = 3

  Tic-Tac-Toe board size.
- const int TicTacToeDef::MAXIMUM_PIECE = 9

  Tic-Tac-Toe maximum piece.

8.6.1 Detailed Description

Tic-Tac-Toe fixed definiton namespace.

Author

Yuchen Zhou (yzhou276@jh.edu)

Version

0.1

Date

2024-09-14

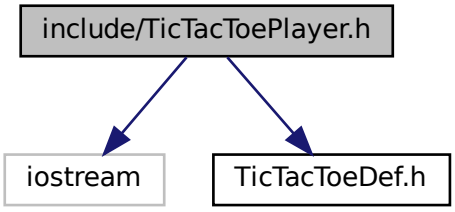
Copyright

8.7 include/TicTacToePlayer.h File Reference

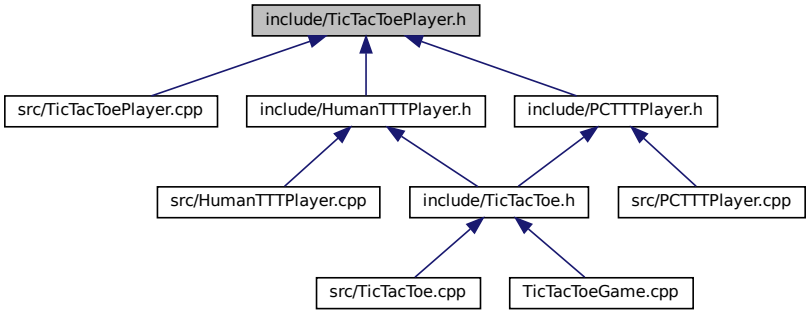
Tic-Tac-Toe Player header file.

```
#include <iostream>
#include "TicTacToeDef.h"
```

Include dependency graph for TicTacToePlayer.h:



This graph shows which files directly or indirectly include this file:



Classes

- class TicTacToePlayer

  This class registers the basic information of a Tic-Tac-Toe player and defines the prototypes of functions that inherited class need to implement.

8.7.1 Detailed Description

Tic-Tac-Toe Player header file.

Author

Yuchen Zhou (yzhou276@jh.edu)

Version

0.1

Date

2024-09-14

Copyright

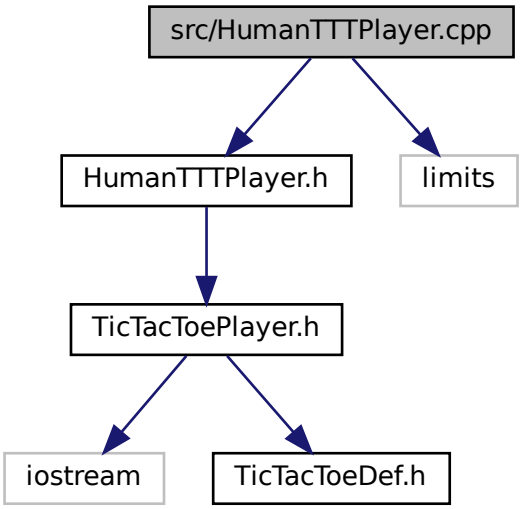
Copyright (c) 2024

8.8 src/HumanTTTPlayer.cpp File Reference

Implementation of HumanTTTPlayer class.

```
#include "HumanTTTPlayer.h"
#include <limits>
```

Include dependency graph for HumanTTTPlayer.cpp:



8.8.1 Detailed Description

Implementation of HumanTTTPlayer class.

Author

Yuchen Zhou (yzhou276@jh.edu)

Version

0.1

Date

2024-09-14

Copyright

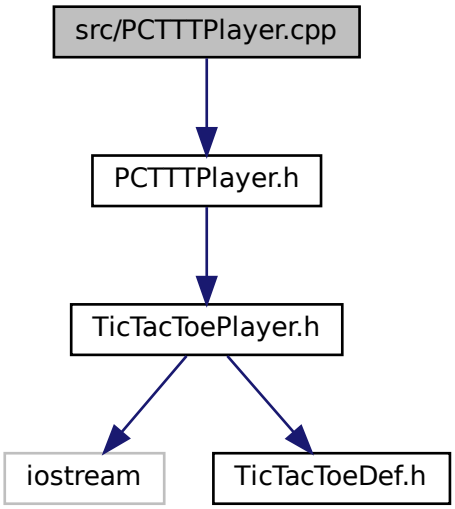
Copyright (c) 2024

8.9 src/PCTTTPlayer.cpp File Reference

Implementation of PCTTTPlayer class.

```
#include "PCTTTPlayer.h"
```

Include dependency graph for PCTTTPlayer.cpp:



8.9.1 Detailed Description

Implementation of PCTTTPlayer class.

Author

Yuchen Zhou (yzhou276@jh.edu)

Version

0.1

Date

2024-09-14

Copyright

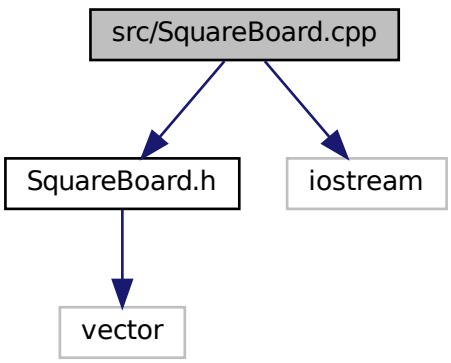
Copyright (c) 2024

8.10 src/SquareBoard.cpp File Reference

Implementation of SquareBoard class.

```
#include "SquareBoard.h"
#include <iostream>
```

Include dependency graph for SquareBoard.cpp:



8.10.1 Detailed Description

Implementation of SquareBoard class.

Author

Yuchen Zhou (yzhou276@jh.edu)

Version

0.1

Date

2024-09-14

Copyright

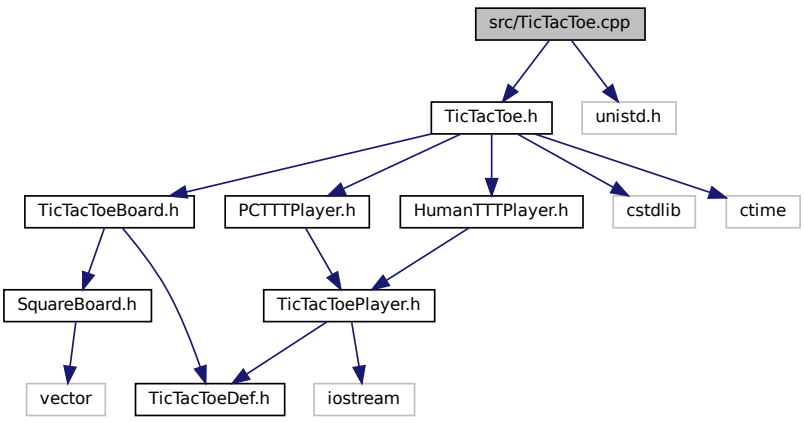
Copyright (c) 2024

8.11 src/TicTacToe.cpp File Reference

Implementation of TicTacToe class.

```
#include "TicTacToe.h"
#include <unistd.h>
```

Include dependency graph for TicTacToe.cpp:



8.11.1 Detailed Description

Implementation of TicTacToe class.

Author

Yuchen Zhou (yzhou276@jh.edu)

Version

0.1

Date

2024-09-14

Copyright

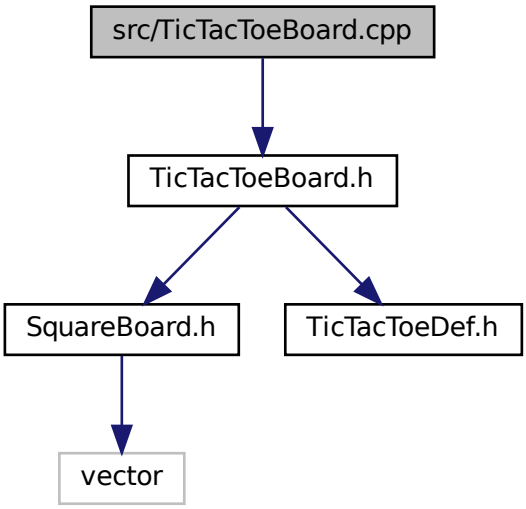
Copyright (c) 2024

8.12 src/TicTacToeBoard.cpp File Reference

Implementation of TicTacToeBoard class.

```
#include "TicTacToeBoard.h"
```

Include dependency graph for TicTacToeBoard.cpp:



8.12.1 Detailed Description

Implementation of TicTacToeBoard class.

Author

Yuchen Zhou (yzhou276@jh.edu)

Version

0.1

Date

2024-09-14

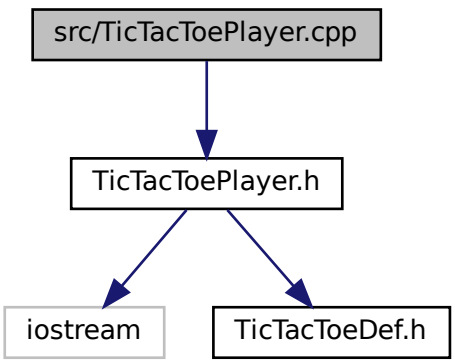
Copyright

8.13 src/TicTacToePlayer.cpp File Reference

Implementation of TicTacToePlayer class.

```
#include "TicTacToePlayer.h"
```

Include dependency graph for TicTacToePlayer.cpp:



8.13.1 Detailed Description

Implementation of TicTacToePlayer class.

Author

Yuchen Zhou (yzhou276@jh.edu)

Version

0.1

Date

2024-09-14

Copyright

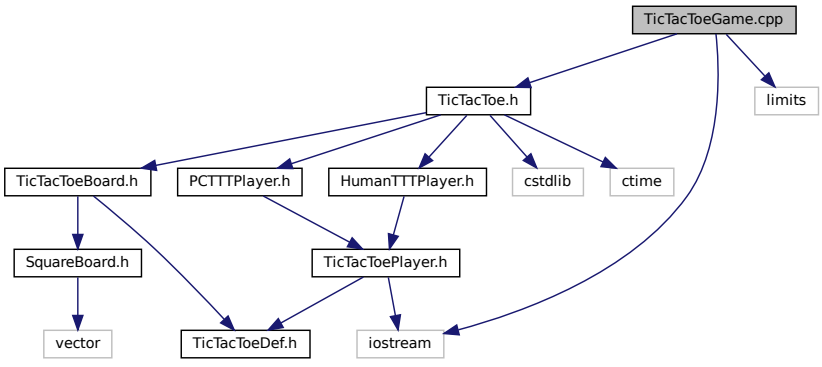
Copyright (c) 2024

8.14 TicTacToeGame.cpp File Reference

Tic-Tac-Toe Game.

```
#include "TicTacToe.h"
#include <limits>
#include <iostream>
```

Include dependency graph for TicTacToeGame.cpp:



Functions

- int main ()

  Tic-Tac-Toe game for client.

8.14.1 Detailed Description

Tic-Tac-Toe Game.

Author

Yuchen Zhou (yzhou276@jh.edu)

Version

0.1

Date

2024-09-14

Copyright

Copyright (c) 2024

8.14.2 Function Documentation

8.14.2.1 main()

```
int main ( )
```

Tic-Tac-Toe game for client.

runs the Tic-Tac-Toe game in the console. After the welcome message, user can select among the three the Tic-↩ Tac-Toe game modes. If the selected mode is valid, the program will dynamically allocate the game object then start the game. Else, the program will print the error message to console and exit.

Return values

int	

Definition at line 28 of file TicTacToeGame.cpp.

Index