

ChessEngine.AI.Riddle

1.0.0

Generated by Doxygen 1.9.5



<b>1 Namespace Index</b>	<b>1</b>
1.1 Package List . . . . .	1
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy . . . . .	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List . . . . .	5
<b>4 Namespace Documentation</b>	<b>7</b>
4.1 ChessEngine Namespace Reference . . . . .	7
4.2 ChessEngine.AI Namespace Reference . . . . .	7
4.3 ChessEngine.AI.Riddle Namespace Reference . . . . .	7
4.4 ChessEngine.AI.Riddle.Utility Namespace Reference . . . . .	7
<b>5 Class Documentation</b>	<b>9</b>
5.1 ChessEngine.AI.Riddle.RiddleChessAI Class Reference . . . . .	9
5.1.1 Detailed Description . . . . .	10
5.1.2 Constructor & Destructor Documentation . . . . .	11
5.1.2.1 RiddleChessAI() . . . . .	11
5.1.3 Member Function Documentation . . . . .	11
5.1.3.1 CalculateMoveScore() . . . . .	11
5.1.3.2 OnBestMoveRequested() . . . . .	12
5.1.3.3 OnBestMoveSubmitted() . . . . .	12
5.1.3.4 StopThinking() . . . . .	12
5.1.3.5 SubmitBestMove() . . . . .	12
5.1.3.6 Thread_ScoreMoves() . . . . .	14
5.2 ChessEngine.AI.Riddle.RiddleChessAI.ScoredValidMove Class Reference . . . . .	14
5.2.1 Detailed Description . . . . .	14
5.3 ChessEngine.AI.Riddle.RiddleChessAI.SubmoveEntry Class Reference . . . . .	15
5.3.1 Detailed Description . . . . .	15
5.4 ChessEngine.AI.Riddle.RiddleChessAI.ValidMoveEntry Class Reference . . . . .	15
5.4.1 Detailed Description . . . . .	15
<b>Index</b>	<b>17</b>



# Chapter 1

## Namespace Index

### 1.1 Package List

Here are the packages with brief descriptions (if available):

<a href="#">ChessEngine</a>	7
<a href="#">ChessEngine.AI</a>	7
<a href="#">ChessEngine.AI.Riddle</a>	7
<a href="#">ChessEngine.AI.Riddle.Utility</a>	7



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

ChessAI	
ChessEngine.AI.Riddle.RiddleChessAI . . . . .	<a href="#">9</a>
ChessEngine.AI.Riddle.RiddleChessAI.ScoredValidMove . . . . .	<a href="#">14</a>
ChessEngine.AI.Riddle.RiddleChessAI.SubmoveEntry . . . . .	<a href="#">15</a>
ChessEngine.AI.Riddle.RiddleChessAI.ValidMoveEntry . . . . .	<a href="#">15</a>





## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">ChessEngine.AI.Riddle.RiddleChessAI</a>	
<a href="#">Riddle chess AI</a> . . . . .	9
<a href="#">ChessEngine.AI.Riddle.RiddleChessAI.ScoredValidMove</a>	
A scored valid move. . . . .	14
<a href="#">ChessEngine.AI.Riddle.RiddleChessAI.SubmoveEntry</a>	
Holds information about submoves at any depth. . . . .	15
<a href="#">ChessEngine.AI.Riddle.RiddleChessAI.ValidMoveEntry</a>	
Holds information about valid moves and attacks that may be made this turn. . . . .	15



## Chapter 4

# Namespace Documentation

### 4.1 ChessEngine Namespace Reference

### 4.2 ChessEngine.AI Namespace Reference

### 4.3 ChessEngine.AI.Riddle Namespace Reference

#### Classes

- class [RiddleChessAI](#)  
*Riddle chess AI.*

### 4.4 ChessEngine.AI.Riddle.Utility Namespace Reference

#### Classes

- class **RiddleMoveEvaluation**  
*A public static class that contains values for different chess moves and pieces.*



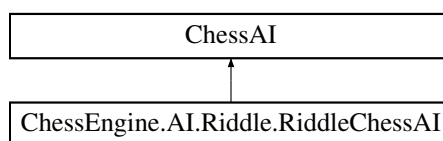
## Chapter 5

# Class Documentation

### 5.1 ChessEngine.AI.Riddle.RiddleChessAI Class Reference

[Riddle](#) chess [AI](#).

Inheritance diagram for ChessEngine.AI.Riddle.RiddleChessAI:



#### Classes

- class [ScoredValidMove](#)  
*A scored valid move.*
- class [SubmoveEntry](#)  
*Holds information about submoves at any depth.*
- class [ValidMoveEntry](#)  
*Holds information about valid moves and attacks that may be made this turn.*

#### Public Member Functions

- [RiddleChessAI](#) (ChessColor pTeam)  
*The constructor for [RiddleChessAI](#).*
- float [CalculateMoveScore](#) (Instance pInstance, MoveData pMoveData, int pMoveDepth, ref Cancellation↔TokenSource pCancellationToken, out GameOverReason pGameOverReason)  
*Gets the single-move score if the move described by pMoveData if played on the chess engine Instance, pInstance.*
- override void **OnBestMoveRequestUpdate** ()  
*Invoked after every 'OnUpdate' in any frame where 'IsBestMovePending == true' and 'IsBestMoveDelayed == false'. Logic to determine best moves should not be executed in this loop as a submission delay causes it to not be executed. It is best practice to submit ready best moves in this callback unless they were demanded, this will prevent any non-demanded submissions while there is a best move submission delay set.*
- override void [OnBestMoveRequested](#) (int pMaxDepth, float pMaxTime)

- Invoked when a best move is requested.*
- override void **OnUpdate** ()  
*Invoked every frame that the [AI](#) is updated.*
- override void **OnBestMoveDemanded** ()  
*Invoked after the [AI](#) demands a best move be submitted immediately.*
- override void **OnBestMoveSubmitted** (TileIndex pFrom, TileIndex pTo)  
*Invoked after the [AI](#) submits a best move.*

## Protected Member Functions

- void **StopThinking** (bool pWaitForThreads)  
*Forces the [AI](#) to stop thinking about a move.*
- void **SubmitBestMove** ()  
*Determines and submits a 'best move' instantly.*
- void **SubmitBestMove** (MoveData pMoveData)  
*Submits a 'best move' with the given move data.*
- void **Thread\_ScoreMoves** (int pThreadIndex, MoveData[] pMoves, SerializedChessInstance pPreMoveState, CancellationTokenSource pCancellationTokens)  
*Invoked on a thread. Given an array of moves to score.*

## Protected Attributes

- object **m\_LockObject**  
*The thread synchronization object.*
- CancellationTokenSource **m\_CancellationTokens**  
*The current cancellation token for the main thread.*
- List< MoveData > **m\_ValidMoves**  
*The valid moves for the [AI](#) on the current turn.*
- Thread[] **m\_Threads**  
*An array of threads used by the [AI](#) to calculate and submit moves.*
- ConcurrentDictionary< int, [ScoredValidMove](#) > **m\_ScoredMoveLookup** = new ConcurrentDictionary<int, [ScoredValidMove](#)>()  
*A concurrent queue where threads submit their highest scored moves. Key: int - The thread index. Value↔ : [ScoredValidMove](#) - The score of the move and the move data itself.*

## Properties

- bool **IsThinking** [get, protected set]  
*Returns true if the [AI](#) is currently thinking about its move, otherwise false if not making a move or finished thinking.*
- int **MaxThreads** = 4 [get, protected set]  
*The maximum # of threads the [AI](#) may use to calculate its next move.*

### 5.1.1 Detailed Description

[Riddle](#) chess [AI](#).

[Riddle](#) is a well balanced chess [AI](#) that takes as much time as he is allowed to carefully consider each move and it's consequences further in the game. Riddles difficulty scales with allowed think time and allowed think depth. [Riddle](#) may think on up to the hardware limit of threads or 'MaxThreads' threads at a time.

## 5.1.2 Constructor & Destructor Documentation

### 5.1.2.1 RiddleChessAI()

```
ChessEngine.AI.Riddle.RiddleChessAI.RiddleChessAI (
    ChessColor pTeam )
```

The constructor for [RiddleChessAI](#).

#### Parameters

<i>pTeam</i>	The team the <a href="#">AI</a> is playing as.
--------------	--

## 5.1.3 Member Function Documentation

### 5.1.3.1 CalculateMoveScore()

```
float ChessEngine.AI.Riddle.RiddleChessAI.CalculateMoveScore (
    Instance pInstance,
    MoveData pMoveData,
    int pMoveDepth,
    ref CancellationTokenSource pCancellationToken,
    out GameOverReason pGameOverReason )
```

Gets the single-move score if the move described by *pMoveData* if played on the chess engine *Instance*, *pInstance*.

#### Parameters

<i>pInstance</i>	
<i>pMoveData</i>	
<i>pMoveDepth</i>	The 'depth' (or how many moves in the future) the move is taking place.
<i>pCancellationToken</i>	A reference to a cancellation token that tracks whether or not move score calculation should be cancelled.
<i>pGameOverReason</i>	An output parameter that holds the reason for the game being over due to the move, or 'GameOverReason.NotOver' if the move did not end the game.

#### Returns

The score of the move *pMoveData* if played in the chess instance *pInstance*.

### 5.1.3.2 OnBestMoveRequested()

```
override void ChessEngine.AI.Riddle.RiddleChessAI.OnBestMoveRequested (
    int pMaxDepth,
    float pMaxTime )
```

Invoked when a best move is requested.

#### Parameters

<i>pMaxDepth</i>	
<i>pMaxTime</i>	

### 5.1.3.3 OnBestMoveSubmitted()

```
override void ChessEngine.AI.Riddle.RiddleChessAI.OnBestMoveSubmitted (
    TileIndex pFrom,
    TileIndex pTo )
```

Invoked after the [AI](#) submits a best move.

#### Parameters

<i>pFrom</i>	
<i>pTo</i>	

### 5.1.3.4 StopThinking()

```
void ChessEngine.AI.Riddle.RiddleChessAI.StopThinking (
    bool pWaitForThreads ) [protected]
```

Forces the [AI](#) to stop thinking about a move.

#### Parameters

<i>pWaitForThreads</i>	Should the main thread wait for all worker threads to return? (Wait for best moves to be submitted.)
------------------------	--

### 5.1.3.5 SubmitBestMove()

```
void ChessEngine.AI.Riddle.RiddleChessAI.SubmitBestMove (
    MoveData pMoveData ) [protected]
```



Submits a 'best move' with the given move data.

## Parameters

<i>pMoveData</i>	
------------------	--

**5.1.3.6 Thread\_ScoreMoves()**

```
void ChessEngine.AI.Riddle.RiddleChessAI.Thread_ScoreMoves (
    int pThreadId,
    MoveData[] pMoves,
    SerializedChessInstance pPreMoveState,
    CancellationTokenSource pCancellationToken ) [protected]
```

Invoked on a thread. Given an array of moves to score.

## Parameters

<i>pThreadId</i>	The index of the thread.
<i>pMoves</i>	
<i>pPreMoveState</i>	
<i>pCancellationToken</i>	

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

- RiddleChessAI.cs

**5.2 ChessEngine.AI.Riddle.RiddleChessAI.ScoredValidMove Class Reference**

A scored valid move.

**Public Attributes**

- float **score**  
*The score for the move.*
- MoveData **moveData**  
*The data that describes the move.*

**5.2.1 Detailed Description**

A scored valid move.

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

- RiddleChessAI.cs

## 5.3 ChessEngine.AI.Riddle.RiddleChessAI.SubmoveEntry Class Reference

Holds information about submoves at any depth.

### Public Attributes

- int **validMoveIndex**  
*The index of the move that is valid this turn for the [AI](#) in the 'valid moves' array.*
- float **cumulativeScore**  
*The cumulative score from 'valid moves' entry at 'validMoveIndex' to and including this move.*
- SerializedChessInstance **postMoveState**  
*The state of the board after the move.*

### 5.3.1 Detailed Description

Holds information about submoves at any depth.

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

- RiddleChessAI.cs

## 5.4 ChessEngine.AI.Riddle.RiddleChessAI.ValidMoveEntry Class Reference

Holds information about valid moves and attacks that may be made this turn.

### Public Attributes

- SerializedChessInstance **preMoveState**  
*The state of the board before the move.*
- MoveData **moveData**  
*The data for the valid move.*
- List< [SubmoveEntry](#) > **submoveEntries**  
*A list of SubMoveEntrys that are overridden every frame while a best move is being considered. This ensures that only the most recent scoring data is considered*

### 5.4.1 Detailed Description

Holds information about valid moves and attacks that may be made this turn.

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

- RiddleChessAI.cs



# Index

- CalculateMoveScore
  - ChessEngine.AI.Riddle.RiddleChessAI, [11](#)
- ChessEngine, [7](#)
- ChessEngine.AI, [7](#)
- ChessEngine.AI.Riddle, [7](#)
- ChessEngine.AI.Riddle.RiddleChessAI, [9](#)
  - CalculateMoveScore, [11](#)
  - OnBestMoveRequested, [11](#)
  - OnBestMoveSubmitted, [12](#)
  - RiddleChessAI, [11](#)
  - StopThinking, [12](#)
  - SubmitBestMove, [12](#)
  - Thread\_ScoreMoves, [14](#)
- ChessEngine.AI.Riddle.RiddleChessAI.ScoredValidMove,
  - [14](#)
- ChessEngine.AI.Riddle.RiddleChessAI.SubmoveEntry,
  - [15](#)
- ChessEngine.AI.Riddle.RiddleChessAI.ValidMoveEntry,
  - [15](#)
- ChessEngine.AI.Riddle.Utility, [7](#)
- OnBestMoveRequested
  - ChessEngine.AI.Riddle.RiddleChessAI, [11](#)
- OnBestMoveSubmitted
  - ChessEngine.AI.Riddle.RiddleChessAI, [12](#)
- RiddleChessAI
  - ChessEngine.AI.Riddle.RiddleChessAI, [11](#)
- StopThinking
  - ChessEngine.AI.Riddle.RiddleChessAI, [12](#)
- SubmitBestMove
  - ChessEngine.AI.Riddle.RiddleChessAI, [12](#)
- Thread\_ScoreMoves
  - ChessEngine.AI.Riddle.RiddleChessAI, [14](#)