King Me

Generated by Doxygen 1.9.1

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

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

Chapter 3

Class Documentation

3.1 board.Board Class Reference

Public Member Functions

- def __init__ (self, gui, board=None, turn="RED")

  The constructor initializes the member variables and calls setBoard.
- def resetBoard (self, gui)

  resetBoard resets the member variables and calls setBoard
- def setBoard (self, gui, board=None)

  setBoard sets the board state based on the colour of pieces the user chooses
- def move (self, piece, move, skipped)

  The move method takes a piece and moves it to the given location.
- def remove (self, piece)

  The remove method takes a piece and removes it from the board.
- def changeTurn (self)

  Changes the turn.
- def checkGameEnd (self)

  Checks to see if the boardState is in a win/loss state.
- def evaluateBoard (self)

  Determines the score of a position.
- def getPieces (self, colour)

  Given a colour, returns all the pieces of that colour.
- def getValidMoves (self, piece)

  Returns all of the valid moves that a piece can make.

Public Attributes

- turn

  turn stores the piece colour that can move
- boardState

  boardState stores the state of the board.
- red_pieces

  red_pieces stores all the red pieces that remain
- white_pieces

  white_pieces stores all the white pieces that remain
- winner

  stores the colour of the winner when the game ends

3.1.1 Constructor & Destructor Documentation

3.1.1.1 __init__()

```
def board.Board.__init__ (
              self,
              gui,
              board = None,
              turn = "RED" )
```

The constructor initializes the member variables and calls setBoard.

Parameters

gui	The GUI is passed in to relay information stored withing the GUI class to the setBoard method

3.1.2 Member Function Documentation

3.1.2.1 checkGameEnd()

```
def board.Board.checkGameEnd (
              self )
```

Checks to see if the boardState is in a win/loss state.

Checks to see if the player with the current turn has any moves remaining.

Returns

True when the boardState is in a win/loss state

False when the boardState is not in a win/loss state

3.1.2.2 evaluateBoard()

```
def board.Board.evaluateBoard (
              self )
```

Determines the score of a position.

Is used by the minmax function

Each player will try to maximize the difference in the number of pieces they have vs the number of pieces their opponent has

3.1.2.3 getPieces()

```
def board.Board.getPieces (
              self,
              colour )
```

Given a colour, returns all the pieces of that colour.

Parameters

colour	The colour of pieces to be returned

Returns

Returns the array of pieces, of the specified colour

3.1.2.4 getValidMoves()

```
def board.Board.getValidMoves (
              self,
              piece )
```

Returns all of the valid moves that a piece can make.

Parameters

piece	The piece to determine the valid moves for

Returns

moves {(row,col): [(row,col),...],...} A dictionary where the keys are the location (stored as a tuple) of all the valid moves the piece can make, the corresponding value is an array of locations (stored as tuples) representing the pieces that are jumped in making that move.

3.1.2.5 move()

```
def board.Board.move (
              self,
              piece,
              move,
              skipped )
```

The move method takes a piece and moves it to the given location.

It must move the piece to the specified location and remove all all the pieces that are jumped along the way

Parameters

piece	Is the piece that will be moving
move	(row,col) Is the location the piece will be moving to
skipped	[(row,col),(row,col),...] Is an array of the locations of the pieces that are jumped and need to be removed

3.1.2.6 remove()

```
def board.Board.remove (
              self,
              piece )
```

The remove method takes a piece and removes it from the board.

Parameters

piece	The piece being removed

3.1.2.7 resetBoard()

```
def board.Board.resetBoard (
              self,
              gui )
```

resetBoard resets the member variables and calls setBoard

Parameters

gui	The GUI is passed in to relay information stored withing the GUI class to the setBoard method

3.1.2.8 setBoard()

```
def board.Board.setBoard (
              self,
              gui,
              board = None )
```

setBoard sets the board state based on the colour of pieces the user chooses

Parameters

gui	The GUI is passed in to relay the piece colour choice made by the user

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

- board.py

3.2 game.Game Class Reference

Public Member Functions

- def __init__ (self, gui)

The init method for Game loads the board and gui to start a new game.

- def reset_game (self)

  reset_game Initalizes the game to a new game and resets all current variables to initial state
- def start_AI (self)

  start_AI kickstarts the game if it is AI's turn first
- def select (self, square)

  select method handles the turns of the user(s)/AI

Public Attributes

- **board**
- **gui**
- **selected**
- **validMoves**
- **winner**

3.2.1 Constructor & Destructor Documentation

3.2.1.1 __init__()

```
def game.Game.__init__ (
              self,
              gui )
```

The init method for Game loads the board and gui to start a new game.

Parameters

gui	The gui class is used to handle the display/graphics

3.2.2 Member Function Documentation

3.2.2.1 select()

```
def game.Game.select (
              self,
              square )
```

select method handles the turns of the user(s)/AI

Handles/Requests the moves of the user(s) and AI based on whos turn it is and based on the game-mode+color selected. Also checks if game has ended yet or not.

Parameters

square	The current square that is selected on the board

3.2.2.2 start_AI()

```
def game.Game.start_AI (
              self )
```

start_AI kickstarts the game if it is AI's turn first

If the game mode is 1-Player and user's color is white, then AI must move first. start_AI is called for that purpose.

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

- game.py

3.3 GUI.GUI Class Reference

Public Member Functions

- def __init__ (self)

  The init method loads the graphics, sets the dimensions of the board, stores class variables and calls make_display()
- def make_display (self)

  make_display() creates the screen, sets the caption and adds the buttons to the screen
- def display_menu (self)

  display_menu() displays the menu buttons and resets the previous winner variable
- def display_board (self, board_state, turn)

  Given the state of a board, displays the board on the screen.
- def display_start (self)

  display_start() displays the blurred out board when starting the game or when the game is over and also displays the start button
- def display_choose_game_mode (self)

  display_choose_game_mode() displays the game modes on the main menu
- def display_choose_color (self)

  display_choose_color() displays the color choices for 1st-player on the main menu
- def display_selected (self, turn)

  display_selected Highlights and displays the selected piece on the board
- def reset_selected (self)

  reset_selected resets the self.selected variable to empty
- def pass_selected (self, piece)

  pass_selected stores the selected piece as a variable for GUI class to use
- def display_validMoves (self)

  display_validMoves Display valid moves of a piece on the board
- def reset_validMoves (self)

  reset_validMoves resets the valid moves dictionary to empty at end of turn

- def pass_validMoves (self, moves)

  pass_validMoves stores the valid moves for the user's selected piece
- def update_message (self, message)

  update_message sets the message to be displayed
- def display_piece (self, colour, row, col)

  display_piece displays a piece of the colour given, in the row and collumn given
- def calc_pos (self, row, col)

  calc_pos calculates the position on the screen of the top left corner of the square given
- def get_clicked_object (self, pos)

  get_clicked_object() is passed the position of a mouseclick and returns what was clicked on
- def get_square_clicked (self, pos)

  get_square_clicked() is called when the user clicks on the board.
- def display_tutorial (self)

  display_Tutorial Toggles the display of Tutorial on/off
- def display_newgame (self)

  display_newgame Display new game countdown on screen
- def display_winner (self, winner)

  display_winner Store the winner value for GUI class to use

Public Attributes

- **new_game**
- **tutorial**
- **start_game**
- **single_player**
- **color_selected**
- **selected**
- **moves**
- **previous_winner**
- **board_height**
- **board_width**
- **screen**
- **message**

3.3.1 Member Function Documentation

3.3.1.1 calc_pos()

```
def GUI.GUI.calc_pos (
              self,
              row,
              col )
```

calc_pos calculates the position on the screen of the top left corner of the square given

This function will be used by display piece to determine where on the screen to place the image

Parameters

row	The row number of the square
col	The collumn number of the square

Returns

(x,y) the coordinates of the top left corner of the square on the screen

3.3.1.2 display_board()

```
def GUI.GUI.display_board (
              self,
              board_state,
              turn )
```

Given the state of a board, displays the board on the screen.

Loops through the board_state and calls display_piece to display the pieces

Parameters

board_state	Two dimensional array representing the state of the board
turn	A color string representing the current turn

3.3.1.3 display_choose_color()

```
def GUI.GUI.display_choose_color (
              self )
```

display_choose_color() displays the color choices for 1st-player on the main menu

The color choices are Red or White. Function highlights the current selected choice.

3.3.1.4 display_choose_game_mode()

```
def GUI.GUI.display_choose_game_mode (
              self )
```

display_choose_game_mode() displays the game modes on the main menu

Game modes available are 1-PLayer or 2-Player. Function highlights the current selected choice.

3.3.1.5 display_newgame()

```
def GUI.GUI.display_newgame (
              self )
```

display_newgame Display new game countdown on screen

Displays images that represent a countdown from 3 seconds to starting the game when a new game is selected

3.3.1.6 display_piece()

```
def GUI.GUI.display_piece (
              self,
              colour,
              row,
              col )
```

display_piece displays a piece of the colour given, in the row and collumn given

Parameters

colour	The colour of the piece to be displayed
row	The row to display the piece
col	The collumn to display the piece

3.3.1.7 display_selected()

```
def GUI.GUI.display_selected (
              self,
              turn )
```

display_selected Highlights and displays the selected piece on the board

Parameters

turn	String that represents the color of the current turn passed in

3.3.1.8 display_validMoves()

```
def GUI.GUI.display_validMoves (
              self )
```

display_validMoves Display valid moves of a piece on the board

Iterates through the valid moves array and highlight them on the board

3.3.1.9 display_winner()

```
def GUI.GUI.display_winner (
              self,
              winner )
```

display_winner Store the winner value for GUI class to use

Modifies/stores values based on the winner of the current game in order for the next screen to be displayed (I.e gameover screen, main menu, etc)

Parameters

winner	String representing the color of current game's winner

3.3.1.10 get_clicked_object()

```
def GUI.GUI.get_clicked_object (
              self,
              pos )
```

get_clicked_object() is passed the position of a mouseclick and returns what was clicked on

Parameters

pos	The tuple representing the mouseclick location on the screen

Returns

A string indicating what was clicked

3.3.1.11 get_square_clicked()

```
def GUI.GUI.get_square_clicked (
              self,
              pos )
```

get_square_clicked() is called when the user clicks on the board.

When the user clicks on the board the tuple containing the row and collumn of the corresponding square clicked on is returned

Parameters

pos	The position of the mouseclick

Returns

(col,row) The row and collumn corresponding to the square the User clicked on

3.3.1.12 pass_selected()

```
def GUI.GUI.pass_selected (
              self,
              piece )
```

pass_selected stores the selected piece as a variable for GUI class to use

Parameters

piece	The piece that is currently selected by user

3.3.1.13 pass_validMoves()

```
def GUI.GUI.pass_validMoves (
              self,
              moves )
```

pass_validMoves stores the valid moves for the user's selected piece

Parameters

moves	The valid moves that is computed by the minmax for the user piece selected

3.3.1.14 update_message()

```
def GUI.GUI.update_message (
              self,
              message )
```

update_message sets the message to be displayed

Parameters

message	The message to be displayed

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

- GUI.py

3.4 menu.menu Class Reference

Public Member Functions

- def tutorial (self, game)

  Class to display the message box (utilizing Tinker library) for the Tutorial once called upon.
- def new_game (self, game)

  Resets the game by reseting the pieces back to original spots.
- def **start_game** (self, game)
- def **select_game_mode** (self, game, mode)
- def **select_color** (self, game, color)

3.4.1 Member Function Documentation

3.4.1.1 new_game()

```
def menu.menu.new_game (
              self,
              game )
```

Resets the game by reseting the pieces back to original spots.

Parameters

board	The game board to be reset
width	The width of the game board
height	The height of the game baord
firstPlayer	The ID of the player

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

- menu.py

3.5 piece.piece Class Reference

Public Member Functions

- def __init__ (self, row, col, color, direction, king=False)

  Class to represent a checkers piece.
- def makeKing (self)

  Represent a checkers king piece.
- def move (self, row, col)

  Move the checkers piece to the desired location.

Public Attributes

- **row**
- **col**
- **color**
- **king**
- **direction**

3.5.1 Constructor & Destructor Documentation

3.5.1.1 __init__()

```
def piece.piece.__init__ (
              self,
              row,
              col,
              color,
              direction,
              king = False )
```

Class to represent a checkers piece.

Parameters

row	The row location of the piece.
col	The column location of the piece.
color	The color of the piece.
direction	The direction of the piece.
king	Boolean whether the piece is a king or not

3.5.2 Member Function Documentation

3.5.2.1 move()

```
def piece.piece.move (
              self,
              row,
              col )
```

Move the checkers piece to the desired location.

Parameters

row	The row location of the move.
col	The column location of the move.

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

- piece.py

Chapter 4

File Documentation

4.1 board.py File Reference

The Board class handles the state of the board and well as the moving of the pieces.

Classes

- class board.Board

4.1.1 Detailed Description

The Board class handles the state of the board and well as the moving of the pieces.

Reference: `https://github.com/techwithtim/Python-Checkers-AI`

Author

Ardhendu, Dylan, Thaneegan

Date

April 5th 2021

4.2 game.py File Reference

Following module handles the logic of the game.

Classes

- class game.Game

4.2.1 Detailed Description

Following module handles the logic of the game.

The logic of the game includes initializing the board and gui and handling the turns of users/AI.

Author

Ardhendu, Dylan, Thaneegan

Date

April 4th 2021

4.3 GUI.py File Reference

Following module handles the graphical user interface for the checkers game.

Classes

- class GUI.GUI

4.3.1 Detailed Description

Following module handles the graphical user interface for the checkers game.

Reference: `https://github.com/binary-b/python-checkers/tree/master/img`

Author

Ardhendu, Dylan, Thaneegan

Date

April 4th 2021

4.4 menu.py File Reference

This class represents the Menu for the game. The menu contains the functionality for Tutorial and New Game.

Classes

- class menu.menu

4.4.1 Detailed Description

This class represents the Menu for the game. The menu contains the functionality for Tutorial and New Game.

Author

Ardhendu, Dylan, Thaneegan

Date

March 15th 2021

4.5 minmax.py File Reference

The minmax file determines the best move the AI can make.

Functions

- def minmax.minmax (currentBoard, maxPlayer, depth)

    Minmax determines the best move that a player can make.

4.5.1 Detailed Description

The minmax file determines the best move the AI can make.

Reference: `https://github.com/techwithtim/Python-Checkers-AI`

Author

Ardhendu, Dylan, Thaneegan

Date

April 5th 2021

4.5.2 Function Documentation

4.5.2.1 minmax()

```
def minmax.minmax (
              currentBoard,
              maxPlayer,
              depth )
```

Minmax determines the best move that a player can make.

Parameters

currentBoard	Is a board object representing the state of the game to be evaluated
maxPlayer	is a boolean value. It is true when the player calling minmax is the player maximizing the score (white pieces in our case). It is negative when the player calling minmax is minimizing the score.
depth	Is the recursive depth that the algorithm will search. It is an exponential algorithm so the higher the recursive depth the slower the AI performs.

Returns

bestBoard.evaluateBoard(), bestBoard The minmax function returns the score of the best move, as well as the board after the best move has been made.

4.6 piece.py File Reference

Class to represent a checkers piece.

Classes

- class piece.piece

4.6.1 Detailed Description

Class to represent a checkers piece.

Author

Ardhendu, Dylan, Thaneegan

Date

April 4th 2021

Index