

---

# **chessboard Documentation**

***Release 1.2.0***

**Hou Lu**

**Oct 22, 2020**



<b>1</b>	<b>Chessboard</b>	<b>3</b>
<b>2</b>	<b>An example: comgames</b>	<b>7</b>
<b>3</b>	<b>Indices and tables</b>	<b>9</b>
<b>4</b>	<b>Installation</b>	<b>11</b>



This is a chessboard display module for board games in command line.



# CHAPTER 1

## Chessboard

This is what the chessboard looks like:

```
* 1 2 3 4 5 6 7 8 9 A
1 |X| | | | | | | |
2 | |O| | | | | | |
3 | | |X| | | | | |
4 | | | |O| | | | |
5 | | | | |X| | | |
6 | | | | | |O| | |
7 | | | | | | |X| |
8 | | | | | | | |O| |
9 | | | | | | | |X| |
A | | | | | | | |O| |
```

## 1.1 Init

The Chessboard class

```
chessboard.Chessboard(
    board_size=3,
    win=3,
    ch_off='O',
    ch_def='X',
    ch_blank=' ',
    user_number=2,
    game_name=None,
    pos=None,
    nested=False
)
```

- `board_size` defines the size of the chessboard
- `win` defines the number of chess pieces to win in a line

- `ch_off` defines the character of offensive player
- `ch_def` defines the character of defensive player
- `ch_black` defines the character of default place
- `user_number` defines the number of players (No use)
- `game_name` defines the built-in `game_name` (default `None`)

## 1.2 Instance Methods

Some methods to operate the chessboard is listed

```
self.set_pos(pos, check=False)
```

- `pos` are the coordinates of chess.
- `check` whether to check winner after this step
- **return** `True` if the current user wins, else, return the current coordinates

```
self.print_pos(coordinates=None, pos=None)
```

- Print the chessboard, if `pos` is given, print `pos`, else, print `self.pos`
- `coordinates` is a list of coordinates which will be printed in specific color.

```
self.rotate_board(angle, unit='radian')
```

- Rotate the chessboard *anticlockwise* for `angle` degree/radian (based on `unit`), using the center of the chessboard as the center of rotation, e.g.,

```
* 1 2 3
1|O|X| |
2| | | |
3| | | |
```

becomes

```
* 1 2 3
1| | |O|
2| | |X|
3| | | |
```

when call `self.rotate_board(270, 'angle')`

```
self.handle_input(input_str, check=False, place=True)
```

- Handle the input of user, can be *coordinates* or *commands*.
- `input_str` The input string.
- `check` Whether to check winner.
- `place` Whether to place a chess or only process the input
- **return** same as `self.set_pos` if `place` is `True`, else, return the current coordinates only.



```
self.validate_pos(pos)
```

- Validate the coordinates.
- pos should be in form (x, y)

```
self.validate_input(input_str, val_pos=True)
```

- Validate the user input.
- **input\_str**, valid user input is
  - x, y
  - u, 1
  - x (only for game *fourinarow*)(x and y are the *one-letter* coordinates)
- val\_pos indicate whether to validate the coodinates

```
self.undo(times=1)
```

- Undo
- times Undo times, default 1



## CHAPTER 2

---

### An example: comgames

---

## 2.1 Installation

```
pip install comgames
```

## 2.2 Usage

```
comgames
```

- Several kinds of board games are built-in.
  - *fourinarow*
  - *Gomoku*
  - *tictactoe*
  - *Reversi*
  - *normal*
- When *normal*, players are asked to input the size of the board and the number of winnings. Max size: 61 Max winning: < size

### 2.2.1 fourinarow

```
* 1 2 3 4 5 6 7
1 | | | | | | |
2 | | | | | | |
3 | | | | | | |
4 | | | O | | |
```

(continues on next page)

(continued from previous page)

5				O		X				
6			O		X		O			
7		O		X		X		O		X

2.2.2 Gomoku

*	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1															
2															
3															
4															
5															
6															
7							O								
8							X		O						
9							X		O						
A										O		X		X	
B											O				
C															
D															
E															
F															

2.2.3 tictactoe

*	1	2	3				
1		O		X		O	
2		X		O		X	
3		X		O		O	

2.2.4 Reversi

*	1	2	3	4	5	6	7	8
1								
2								
3								
4				O		X		
5				X		O		
6								
7								
8								

- [genindex](#)
- [modindex](#)
- [search](#)





## CHAPTER 4

---

### Installation

---

The *chessboardCLI* package is available on *pypi*

```
pip install chessboardCLI
```

Remember, the module name is *chessboard* and the package name is *chessboardCLI*.

Current Version: 1.2.0