

# DStauffman Library

David C. Stauffer

16 August 2016

# Agenda

- ▶ DStauffman overview
- ▶ Tic Tac Toe game
  - ▶ Interactive demo?

# What is the dstauffman library?

- ▶ Python library of useful utilities, plus fun games and apps written by David Stauffer.
- ▶ Available on GitHub as a public repository (<https://github.com/DStauffman/dstauffman>)
- ▶ Runs on Python v3.5+
- ▶ Mainly geared towards numerical analysis, so it requires Matplotlib, NumPy, pandas, PyQt4/5 and a handful of other libraries

# Library High Level View

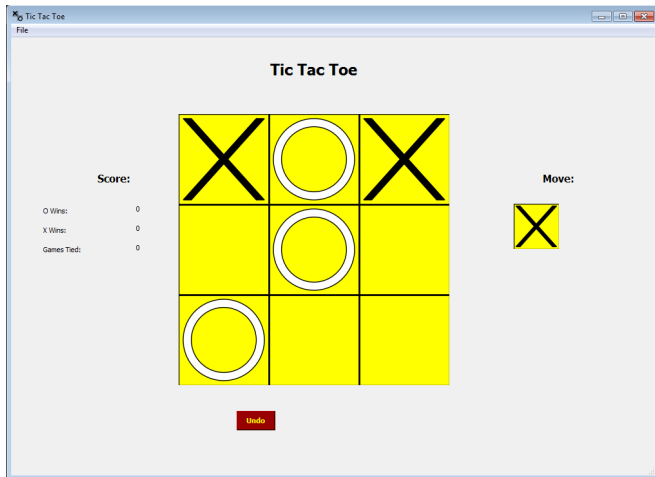
- ▶ Includes User's Guide, documents written in  $\text{\LaTeX}$  to continue open source software theme
- ▶ Code documentation using Sphinx and docstrings
- ▶ Unit test cases
- ▶ Main areas of code:
  - ▶ Generic utilities (folder & file manipulation, enum metaclasses, frozen class attributes, save & load decorators, etc.)
  - ▶ Quaternion methods
  - ▶ Image Processing (photo manipulation and renaming/resizing)
  - ▶ Batch Parameter Estimator for minimizing cost evaluation of arbitrary functions
  - ▶ Archery scoresheets and tournament brackets
  - ▶ Games
    - ▶ Pentago
    - ▶ Tic Tac Toe
    - ▶ Knight board
    - ▶ Brick/Rubik's Cube
    - ▶ Playing Cards and games

As of 15 August 2016:

- ▶ Number of Files: 135. Of which, 93 are Python code
- ▶ Total lines of code  $\approx$ 25K
- ▶ Of these: 13K executable (52%), 3K blank lines (13%), 9K comment lines (35%)
- ▶ Unit test cases currently cover 76% of code base, including some unfinished modules

# Tic Tac Toe

Interactive Demo (if we can get it up and running, otherwise see backup charts)



# Conclusions

The dstauffman library is a fun place to learn and test different Python tricks, while making it a public repository encourages good software practices such as code documentation and unit testing, and hopefully simultaneously provides code that is useful to other people.

Backup



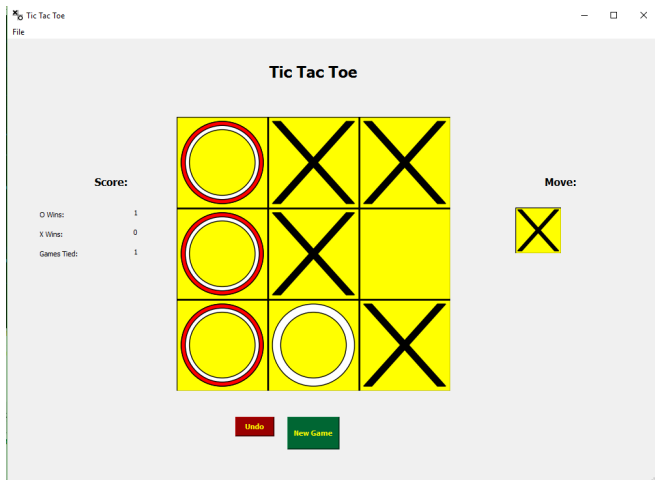
## Backup Details

This project is nice, because it's small enough to wrap your head around it, but makes use of a lot of Python features. These include:

- ▶ numpy arrays for tracking and evaluating board positions.
- ▶ PyQt5 (or PyQt4) for the graphics engine with event driven design.
- ▶ matplotlib plotting capabilities for drawing the board game.
- ▶ OOP design with classes for capturing game options, current state, and game history, with the ability to save and resume across sessions. Plus magic methods, like `__eq__` and `__lt__` to sort the best moves first.
- ▶ Unit tests and doc tests to demonstrate and verify the functions.

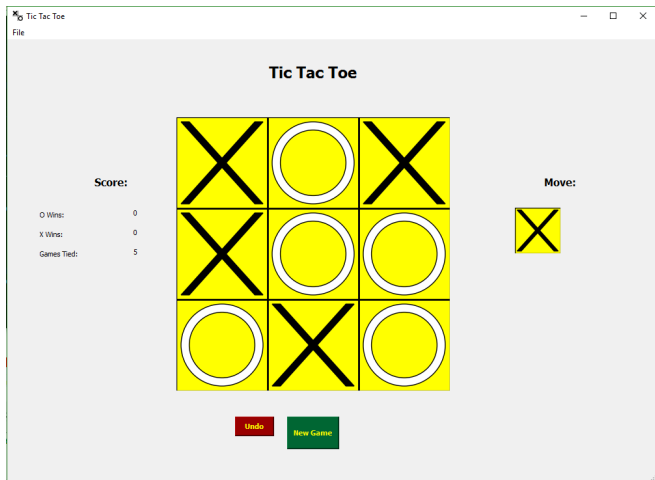
# Backup Details

You can play with two players and have the game keep track of the winning history.



# Backup Details

Alternatively, you can play against the AI, or even have the computer play against itself. (If you can beat the computer, I'll buy you lunch).



# Backup Details

You can adjust some options to show how the computer is graded the open squares and where the possible winning moves are. Relevant controls like “New Game” or “Redo” appear as necessary.

