

Name: _____

Partner: _____

Python Activity 42: Tic Tac Toe – Graphical User Interface

We've built the data layer for our game, now to build the GUI!

Learning Objectives

Students will be able to:

Content:

- Explain how a GUI can be wrapped around a data layer
- Summarize the event-handling programming paradigm
- Determine when to add helper methods or attributes to a class
- Explain how extending/inheriting a class reduces code redundancy

Process:

- Write code that implements a graphical user interface for a text-based game
- Write code that integrates many classes into a single, large project

Prior Knowledge

- Python concepts: user-defined classes, inheritance, tic-tac-toe

Critical Thinking Questions:

-  1. Follow along in the class lecture, and match the following methods/concepts on the left to their purpose on the right:

data layer	GUI element that we place on a screen
pygame	Provides an additional layer of abstraction
event listening	TextBox that notices clicking
wrapper class	A region to display text lists
Window	Simple rectangle to draw
Widget	Screen for placing graphics on
TextBox	External module for building games in python
Button	GUI element for holding text
Square	Programs wait for an event, and then handle it
TextList	Text-based representation of a project

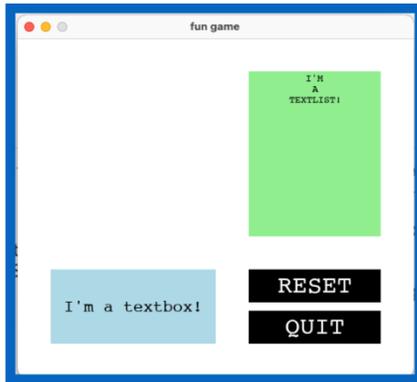
2. Follow along in class lecture, and fill out the *Class Object Models* below for Game: (*Hint: Be sure to include each method's return types, and the names of any parameters!*)

```
class Game
```

 Attributes:

 Methods:

3. Observe the screenshot below of a Game running:



a. How might we adapt this game interface to implement Tic-Tac-Toe?

b. How about adapting for Boggle?

4. Follow along in class lecture, and fill out the *Class Object Models* below for TTTgame: (*Hint: Be sure to include each method's return types, and the names of any parameters!*)

```
class TTTgame
```

Attributes:

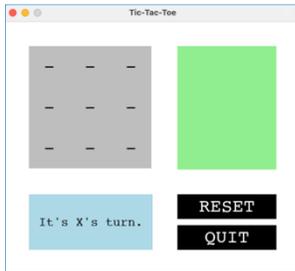
Methods:

5. Follow along in class lecture, and fill out the *Class Object Models* below for TTTbox: (*Hint: Be sure to include each method's return types, and the names of any parameters!*)

```
class TTTbox
```

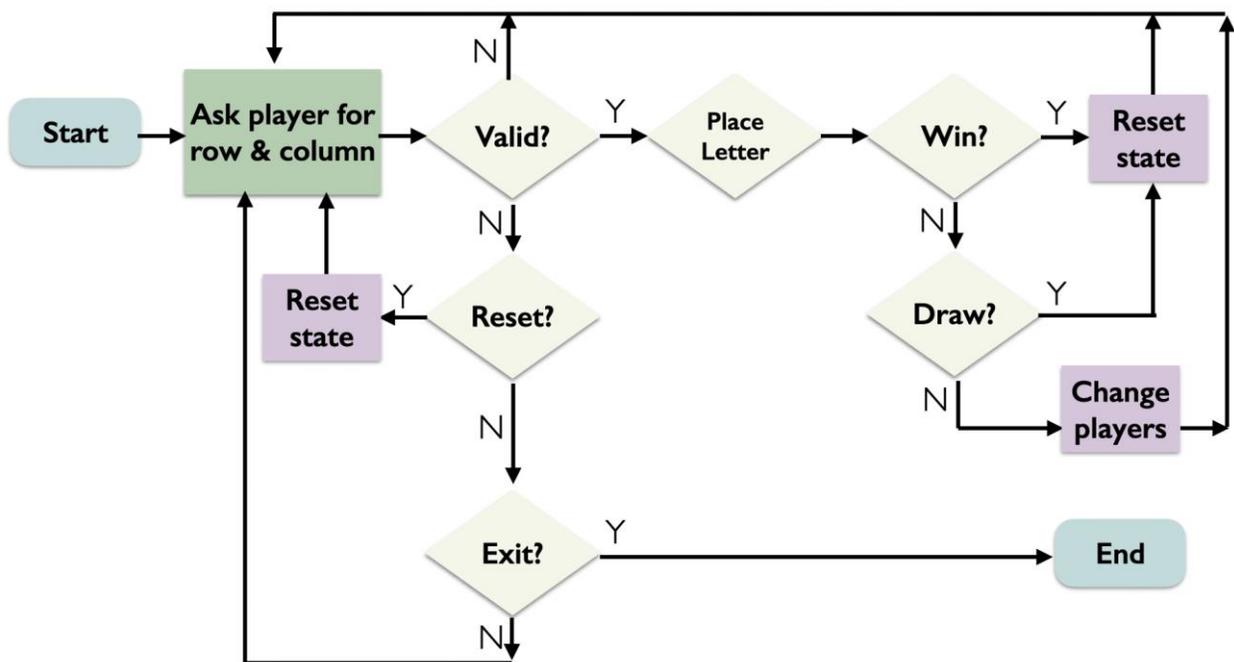
Attributes:

Methods:

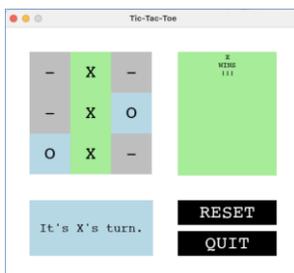


6. Examine the screenshot of an in-process TTTgame being developed. Given a mouse click's x & y position, how might we determine if the user clicked on the gray-hyphen-grid?

7. Follow along in lecture and note how the game logic for TTTgame differs from our game logic for play_text_game () from the previous POGIL. (play_text_game () diagram provided below)



8. When someone has won the Tic-Tac-Toe game, how might we lock the board so additional play area presses are ignored?



Application Questions: Use Python to check your work

(The Boggle Lab is a really good application of these concepts!)