Tic-tac-toe Code Challenge

Author: Sandra Davis

Table of Contents

[1. INTRODUCTION 3](#_Toc392074824)

[2. LANGUAGES 3](#_Toc392074825)

[3. GUI DESIGN 3](#_Toc392074826)

[4. TESTING SOFTWARE 4](#_Toc392074827)

[5. INSTALLATION 5](#_Toc392074828)

# INTRODUCTION

The purpose of this document is to go over the design and installation of the tic-tac-toe application.

# LANGUAGES

The tic-tac-toe application is written in the following languages:

* Django
* JavaScript
* HTML5
* CSS

# GUI DESIGN

I created the grid with HTML5. The grid is 400px by 400px.

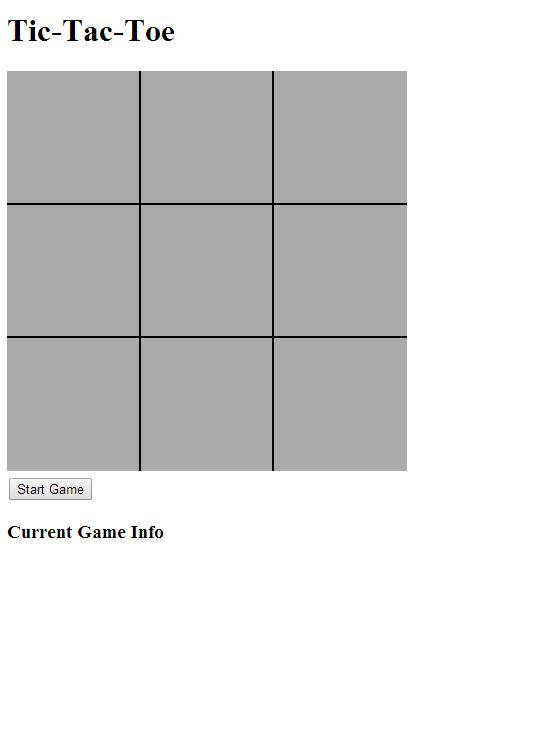


Figure : Tic Tac Toe Application

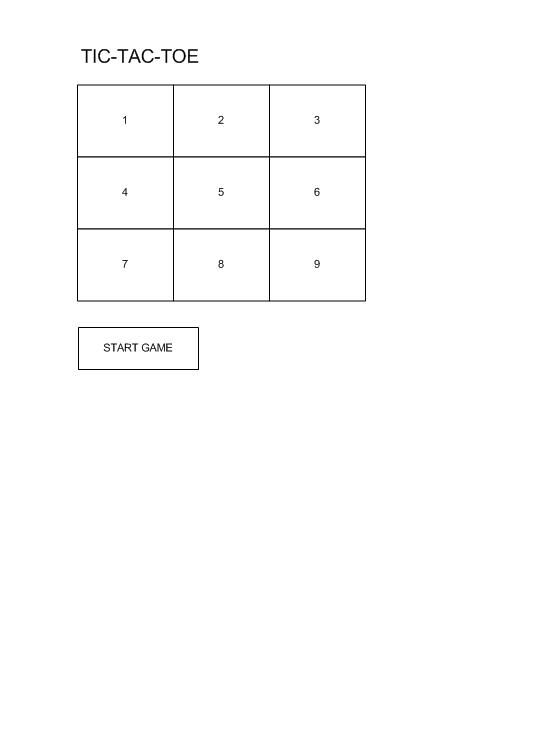
By drawing the canvas, I was able to set specific positions for drawing the vertical and horizontal lines. This made it easier to configure where each square would be and to determine which square the player chose when he clicked. I would do this by grabbing the (x,y) coordinates of the mouse click on the canvas.

Vertical Line 2

266

Vertical Line 1

133



Horizontal Line 2

266

Horizontal Line 1

133

Figure : Grid Design

As you can see in Figure 2, the horizontal and vertical lines are drawn at a specific coordinate location. So, if the player clicked in square 1, the x coordinate would have to be less than 133 and the y coordinate would have to be less than 133.

# TESTING SOFTWARE

I tested the application using different types of software.

* Django Test Cases
* Selenium (installed locally)
  + Ran into problems writing some test cases with this. Due to using the windows.prompt() for asking user whether he would like to be ‘X’ or ‘O’.
* Selenium IDE (FireFox Plugin)
  + Began testing the functionality of the application using the plugin. I was able to test and confirm that the windows.prompt() was working correctly.
  + Ran into problems trying to test the clicks on the HTML5 “canvas”. Since the “canvas” is divided into 9 squares. It wouldn’t allow me to test just where I clicked at on the canvas.
    - This made me do some more researching to find another way to test my JavaScript code.
* Qunit
  + Wrote some test cases to test the JavaScript functions.
  + This allowed me to confirm that the grid positions were correct.

# INSTALLATION

* Download the code and place it somewhere on your computer.
* To start the application:
  + Change to the <downloaded\_dir>/tic-tac-toe/ directory in git shell
  + Start the Runserver: python manage.py runserver
* Open your browser to: localhost:8000