

INFO2180 - Lab 4 (20 marks)

Due Date: **October 17, 2019**

Tic-Tac-Toe

Tic-tac-toe is typically a paper-and-pencil game for two players, X and O, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game.

In this lab we'll be using HTML, CSS and JavaScript to setup and play a simple game of Tic-Tac-Toe in a web browser.



You're given files **index.html** and **tic-tac-toe.css**. You should get the starter code by downloading the files from the following link:


<https://github.com/uwi-info2180/info2180-lab4-tic-tac-toe/archive/master.zip>

It contains a HTML and CSS file, but these should **NOT** be modified. You should only write a script file **tic-tac-toe.js** in the same directory as these files that provides all the event handling and behavior to make the Tic Tac Toe game work as specified below.

Create your Repository and Ensure you are using Github Pages to host you game

You are also required to create a Git repository on Github called **info2180-lab4** and after each exercise commit your code and push to Github. So let us start by creating that repository.

1. You can create a new Github repository by going to <https://github.com/new> or by clicking the "+" icon in the top right once you are logged into the website and select "New Repository".
2. For this lab, use the repository name as **info2180-lab4**.
3. Ensure that your repository is Public and that you select the option to initialize the project with a README file.
4. The remaining options can be left at their defaults. Then click on **Create Repository**.


 Search or jump to... Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner

Repository name *

 ylynfatt

/ info2180-lab2

Great repository names are short and memorable. Need inspiration? How about [redesigned-memory](#)?

Description (optional)

☒ Public

Anyone can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: **None**

Add a license: **None**

Create repository

5. On your newly created repository, click on the "Clone and Download" button and copy the URL in the box that appears.

uwi-info2180 / info2180-lab2

Unwatch 1 Star 0 Fork 0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

No description, website, or topics provided. [Edit](#)

[Manage topics](#)

1 commit

1 branch

0 releases

1 contributor

Branch: master


New pull request

Create new file

Upload files

Find File

Clone or download

 ylynfatt Initial commit

Latest commit 12294d2 1 minute ago

README.md

Initial commit

1 minute ago

README.md

info2180-lab2

6. Open your command prompt, navigate to the folder (e.g. using the `cd` command) where you would like to clone your repository and using the URL you copied in step 5, type the following:

```
$ git clone https://github.com/<your-username>/info2180-lab4.git
```

NOTE: Ensure you change the URL and use the one that you copied in Step 5 with your username.

7. Then navigate to your newly cloned repository by typing:

```
$ cd info2180-lab4
```

8. You can then type the command `ls` and you should see a `README.md` file.
9. Create a new branch called **gh-pages**.

```
$ git checkout -b gh-pages
```

10. Now download the starter files for the lab at the following URL:

<https://github.com/uwi-info2180/info2180-lab4-tic-tac-toe/archive/master.zip>

11. Unzip and copy the starter files from the link below into your **info2180-lab4** folder that you clone in Step 6 and do an initial commit.

```
$ git add .  
$ git commit -m "Added initial starter files"
```

12. Now begin the next set of exercises.

Exercise 1 - Layout the board

The first task is to write event-handling code so that when the page loads, each square in the 3x3 grid game board is styled appropriately by adding the right class. Write your

JavaScript code unobtrusively, **without** modifying either the **index.html** or **tic-tac-toe.css** files.

Hint: It might be helpful to set each **div** inside the game board to have the provided CSS class **square**, using the JavaScript **classList** property or **setAttribute()** method or **className** property that you learnt in your lecture.

Hint 2: You may also want to ensure you use the **onload** or **DOMContentLoaded** handler.

Once you have completed this exercise, take the opportunity to commit and push your code to Github.

Exercise 2 - Add an X or O to a square when clicked

For your next task, when a user clicks on a square in the grid it should alternate putting an **X** or an **O** onto the square that was clicked. Also ensure you add the **class "X"** or **"O"** to the square so that it is styled with the appropriate colour from the stylesheet.

Hint: You may want to initialize an empty array to keep track of the state of the game after each square is clicked so that you can use it later to check which user has won. You can also use the JavaScript **innerHTML** or **textContent** property to make the appropriate **X** or **O** show up in the appropriate **div**.

Once you have completed this exercise, take the opportunity to commit and push your code to Github.

Exercise 3 - Change the style when you move your mouse over a square

Now let us see if we can make it a little more interactive by changing the look of the square whenever a user moves their mouse over a square and then return it to the original style when the user's mouse leaves the square. You may notice in your **tic-tac-toe.css** file that there is a class called **hover**, which may be helpful.

Once you have completed this exercise, take the opportunity to commit and push your code to Github.

Exercise 4 - Check for the winner and update the status

When the user gets three X's or O's in a row they are declared the winner. Your next task is to check when either X or O has won and update the message on the page to say "***Congratulations! X is the Winner!***" or "***Congratulations! O is the Winner!***" depending on who wins. There is a **div** with an **id** of **status**, that this message should be placed in. You should also ensure that the class **you-won** is added to the status **div** as well.

Hint: You can use the JavaScript **innerHTML** or **textContent** property to help you to place the message in the **status div**.

Once you have completed this exercise, take the opportunity to commit and push your code to Github.

Exercise 5 - Restart the game

One annoying thing you may be noticing as you test the game so far is that it can't easily be reset to try again. So our next task will be to make it so that when the user clicks the New Game button, the game state will reset. That is, if the game squares have an X or an O then those are to be removed and the status message returns to the original message so that the user can try to play the game again.

Hint: You should use the **click** event.

Once you have completed this exercise, take the opportunity to commit and push your code to Github.

Exercise 6 - Disallow Cheating

Depending on how you coded the previous exercises, you may have a case where if a user clicks on a square that already has a value then they are able to change what was there before. Fix this by making it so that a user is not able to change the value of a square that already has an X or an O.

Once you have completed this exercise, take the opportunity to commit and push your code to Github.

Submission

To submit this lab you will be required to push your completed code to Github and also ensure you are using Github Pages (remember you will need to create a **gh-pages** branch). Submit your Github pages URL (e.g. <https://yourusername.github.io/info2180-lab4>) via the “Lab 4 Submission” link on OurVLE.