# Module 2 Tasks

We have two main goals in Module 2: Fixing bugs and refactoring the Flood Fill code; and producing a Tic-Tac-Toe game using Flood Fill as a reference:

1. **Bug fix and refactor module 2 code**: The Flood Fill game template provided in module 2 includes some improvements over the module 1 template, but it also contains some bugs and some poorly structured JavaScript. For this task you must:
   1. Install ESList (requires installing npm) and use it to clean up the poorly structured JavaScript (i.e. linting errors identified by ESLint – not bugs)
   2. Identify and fix **three** bugs that I have intentionally left in the module 2 code. I suggest using console and debugging tools as described in our readings this week.
   3. Improve the module 2 code by reducing the number of calls to the render()function.
2. **Develop a Tic Tac Toe game:** The fundamental structures of Flood Fill would work equally well for a game of Tic Tac Toe, with the major difference that there is no requirement for a flood fill (or similar) algorithm in Tic Tac Toe. For this task you must create a **2-player** (meaning assume both players are sitting together and playing using the same computer and mouse – no need for ‘AI’ or remote play options) Tic Tac Toe game that includes the following:
   1. Use the module 2 code as a base – keep as much as you can that would be reasonable for a Tic Tac Toe game (i.e. history, undo, render and updateGridAt – with necessary changes)
   2. Make sure to install ESLint to keep your code clean (note that all project submissions going forward will \*require\* the inclusion of ESLint).
   3. Use canvas to draw your game (note you don’t necessarily \*need\* Xs and Os – you can just fill in cells with two colours, if that’s easier)

# Milestone 1 – Requirements

The requirements below follow the general template steps provided in the Milestone Overview document. The blank spaces below should be used to complete the steps as outlined.

## Task 1 – Bug fix and refactor module 2 template code

### Define the problem

Write at least one paragraph that describes the ‘user’-facing problem we are attempting to solve by implementing Task 1. In the case of Task 1, you may think of the ‘user’ as another developer who might be working on your code.

**Problem definition:**

|  |
| --- |
|  |

### Identify potential solutions

For this task the ‘identify potential solutions’ component will follow a slightly different approach. Rather than identifying two potential solutions to the problem you’ve stated above, I want you to outline the steps that you expect you will take to address the following three sub-tasks:

1. Explain how you will identify structural code problems in the module 2 starter code (hint: ESLint!)
2. Explain the process you will take to identify bugs in the code (hint: think about the console and debugger – there are three bugs!)
3. Explain what your approach will be to refactoring the code to reduce the number of calls to the render() function.

**Structural code problems:**

|  |
| --- |
|  |

**Code bugs:**

|  |
| --- |
|  |

**Refactoring:**

|  |
| --- |
|  |

### Analyze potential solutions

Write a series of pros and cons that help to explain why you believe your approach to the second sub-task, *explain the process you will take to identify bugs in the code*, was a good a good approach (i.e. explain why it worked, and either what could be improved upon). Try to identify at least three pros and three cons.

**Analysis of first potential solution:**

|  |  |
| --- | --- |
| **Pros** | **Cons** |
|  |  |

## Task 2 – Create a Tic Tac Toe game

### Define the problem

Write at least one paragraph that describes at a high level what you will need to create a Tic Tac Toe game. Keep the description user-facing, meaning that you should think about the UI and how the players will engage with the game.

**Problem definition:**

|  |
| --- |
|  |

### Identify coding requirements

Use this space to identify all code changes that you expect you will need to make in the process of developing the Tic Tac Toe game. This means identifying structures that you will use from the module 2 code, structures that you will need to modify, and structures that you will need to add. Do your best to be as specific as possible.

**Code requirement:**

|  |
| --- |
|  |

### Analyze potential solutions

Use this space to identify components of your solution that you believe will be easier to develop, and components that you believe will be more difficult to develop. You should endeavour to be as complete as possible and categorize all components identified in the Code Requirements section above. Provide a short explanation justifying why you think the component will be easier or more challenging.

**Easier components:**

|  |
| --- |
|  |

**More challenging components:**

|  |
| --- |
|  |