Description of Your Project

I will be simulating the board game battleship. Given the scope and time limitation of the project, I will not be adding all functionality that the game has, but will be making a limited, condensed version of the original game. There will only be three "battleships" the player can choose from (1x2, 1x3, 1x3) and the board will only be 6x6 instead of the usual 10x10.

The player will be able to see both their board and the 'other' player's board (they will be competing against the computer the whole time).

The game will have a start mode, where the player can select each ship and place them on the board, then begin game play. A turn will consist of the player clicking on the other player's board, guessing where they think a ship is. There will be an output where the ships were originally located displaying whether the guess was a hit, miss, or sink. The player's board will also change color to display this in a visual format. The computer's turn consist of a guess, displayed in the output box, and a change on the player's board. The game will run until one player's ships are all sunk, declaring the other player the winner.

Pseudocode for Algorithms and Program Flow

Code – High Level

Variables - Global:

playerDisplayBoard = empty, what gets displayed to screen, written in divs
computerDisplayBoard = empty, what gets displayed to screen
state = state of the game, 0=place ships, 1=player turn, 2=computer turn, 3=game end
playerShip1, playerShip2, playerShip3 = array(), gives coordinates to where the ships are on the player's board
computerShip1, computerShip2, computerShip3 = array(), gives coordinates to where the ships are on the computer's board

Start

Explain basic game to player in dialogue box.
Have player place their ships on board.

Player's Turn

Player clicks on computer's board.
Hit, Miss, or Sink is displayed in dialogue Box.
Changes color of computer's display board

Computer's Turn

Computer's selection is displayed in dialogue box.
Hit, Miss, or Sink is displayed in dialogue Box.
Changes color of play's display board

Game End

Fade out player and computer's boards.
Display who won.

Code – Low Level

Variables - Global:

```
playerDisplayBoard = array(array(<div id="p1"></div>));
computerDisplayBoard = array(array(<div id="c1"></div>));
state = 0;
playerShip1 = array('0', '0');
playerShip2 = array('0', '0', '0');
playerShip3 = array('0', '0', '0');
computerShip1 = array('0', '0');
computerShip2 = array('0', '0', '0');
computerShip13= array('0', '0', '0');
```

Start

```
state = 0;
document.getElementById('dialogueBox').innerHTML = "This is the game.  Blah, blah, blah";
<div id="playerShip1"><a href="#" onClick="placeShip('playerShip1');"></a></div>
function placeShip(divID)
{
 foreach
 (
  document.getElementById('playerBoardElement').innerHTML="onMouseOver='highlightElement()' ";
 )
}
```

Player's Turn

```
state = 1;
<div id="computerDisplayBoard"><a href="#" onClick="playerClick(divID)"></a></div>
document.getElementById("dialogueBox").innerHTML = "<p>Player's turn</p><p>Pick: " +
  getPlayerPick() + "</p><p>" + getHMS() + "</p>";
document.getElementByID(divID).appendChild(hitClass);
```

Computer's Turn

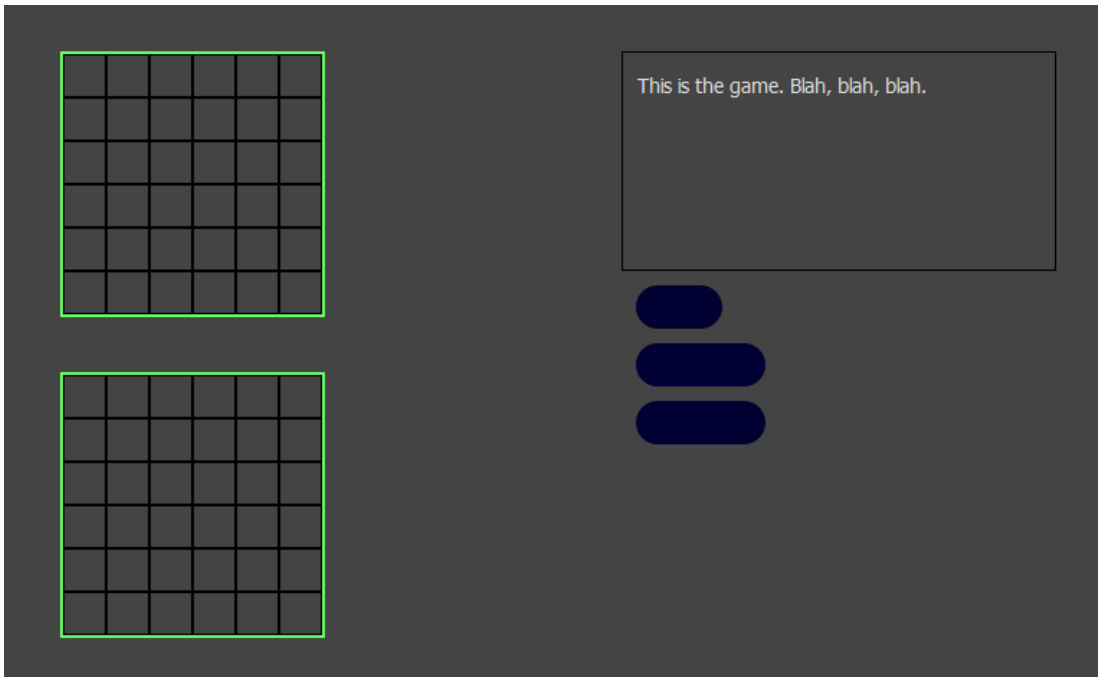
```
state = 2;
<div id="playerDisplayBoard"><a href="#" onClick="computerClick(divID)"></a></div>
document.getElementById("dialogueBox").innerHTML = "<p>Computer's turn</p><p>Pick: " +
  getComputerPick() + "</p><p>" + getHMS() + "</p>";
document.getElementByID(divID).appendChild(hitClass);
```

Repeat in while loop until state=3

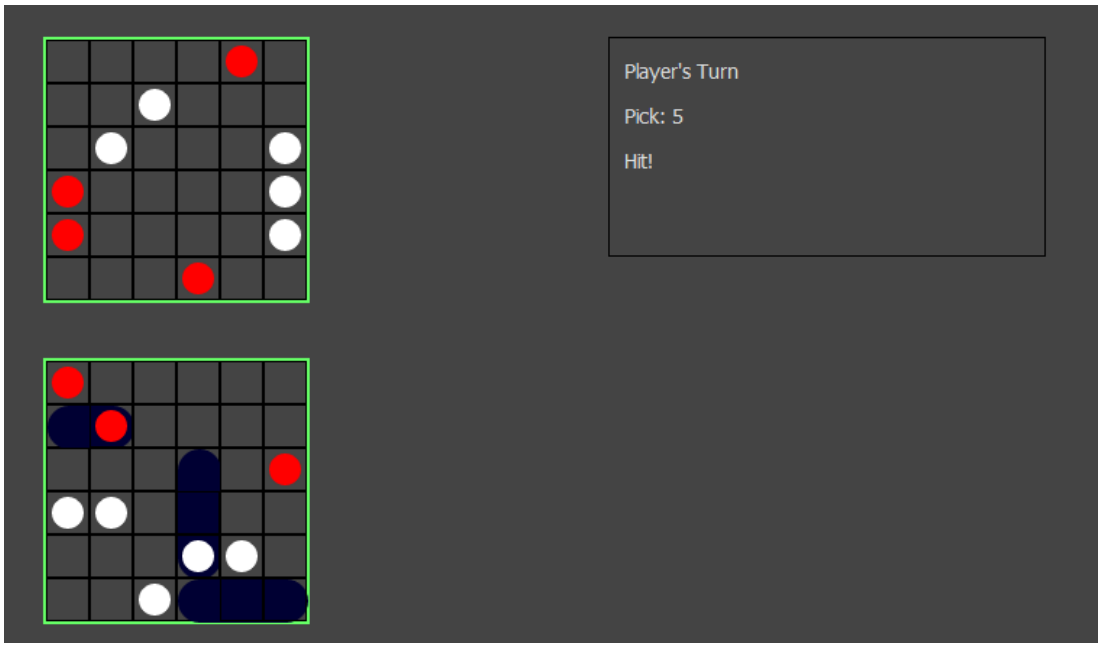
Game End

```
state = 3;
document.getElementByID('content').appendChild('displayWinner');
```

Visual Design Template or Example



This is an example of what the starting state will look like



This is an example of game play

Brief Description of How You Will Test Your Project

Every part of this game will be tested first through a proof of concept page. I will take the new functionality I will be adding and test a simple version on a separate page to prove that I will be able to accomplish the idea. I will then add parts to the game piece by piece. I will play the game at each of these levels, testing the boundary values at each point. After game completion, I will play the entire game fully a few more times to make sure everything is running smoothly.