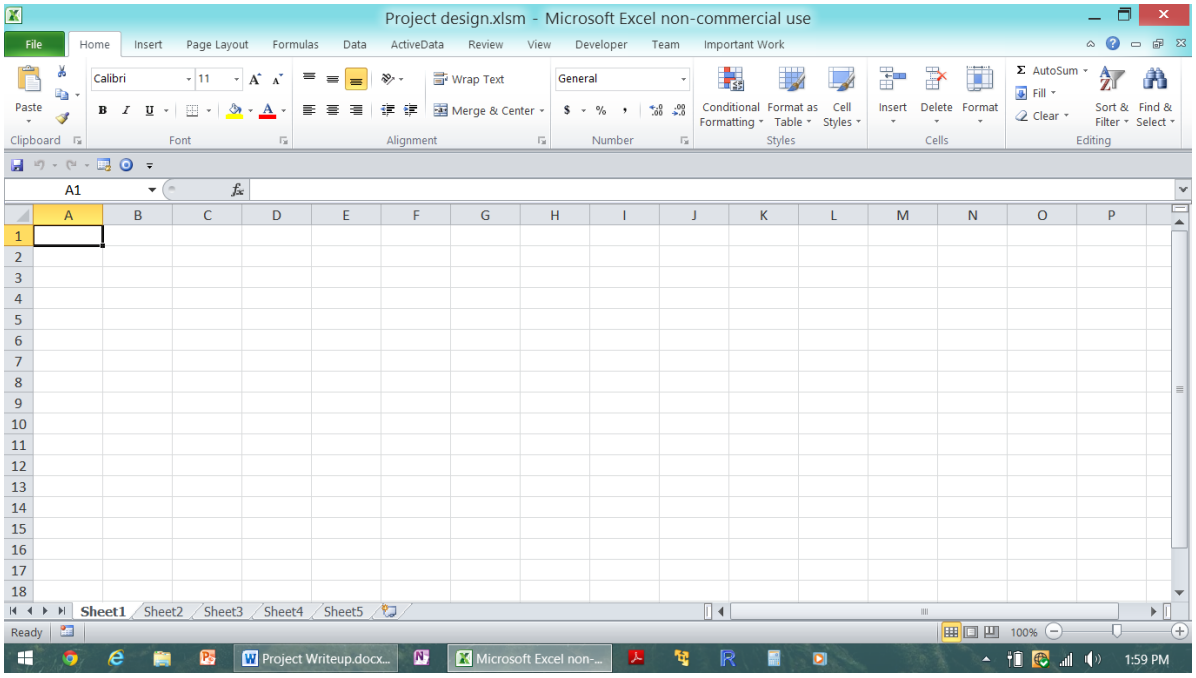
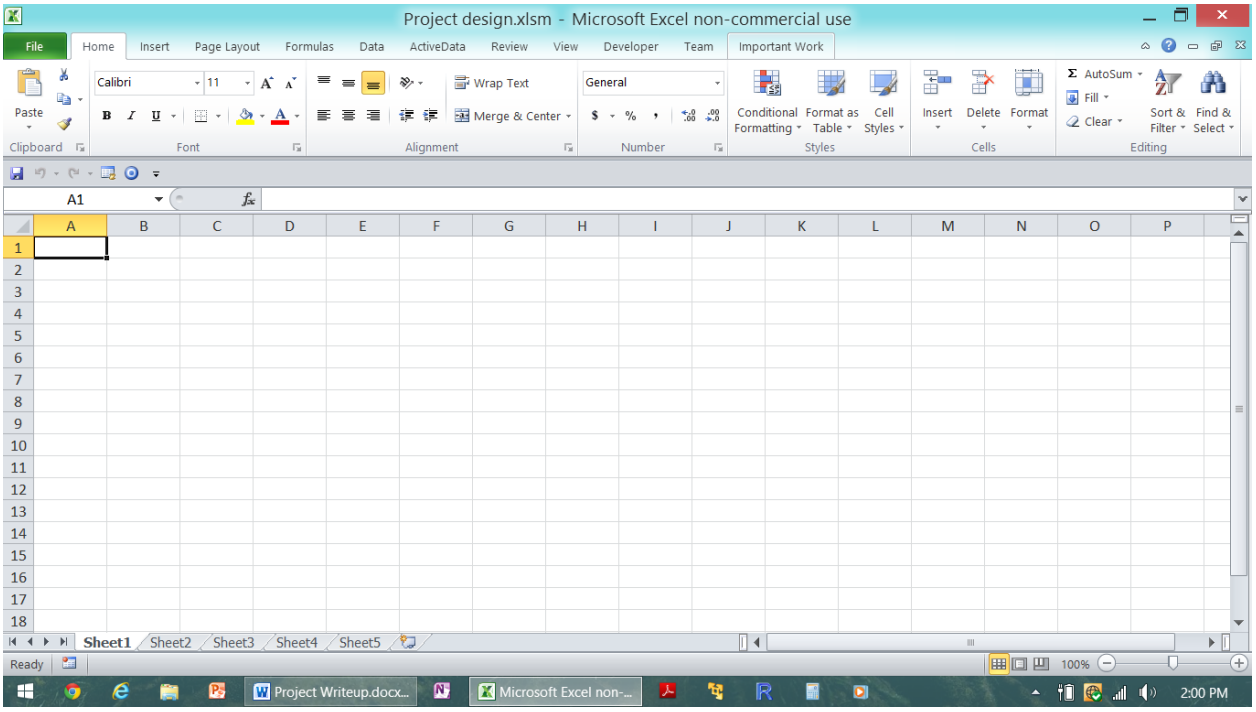
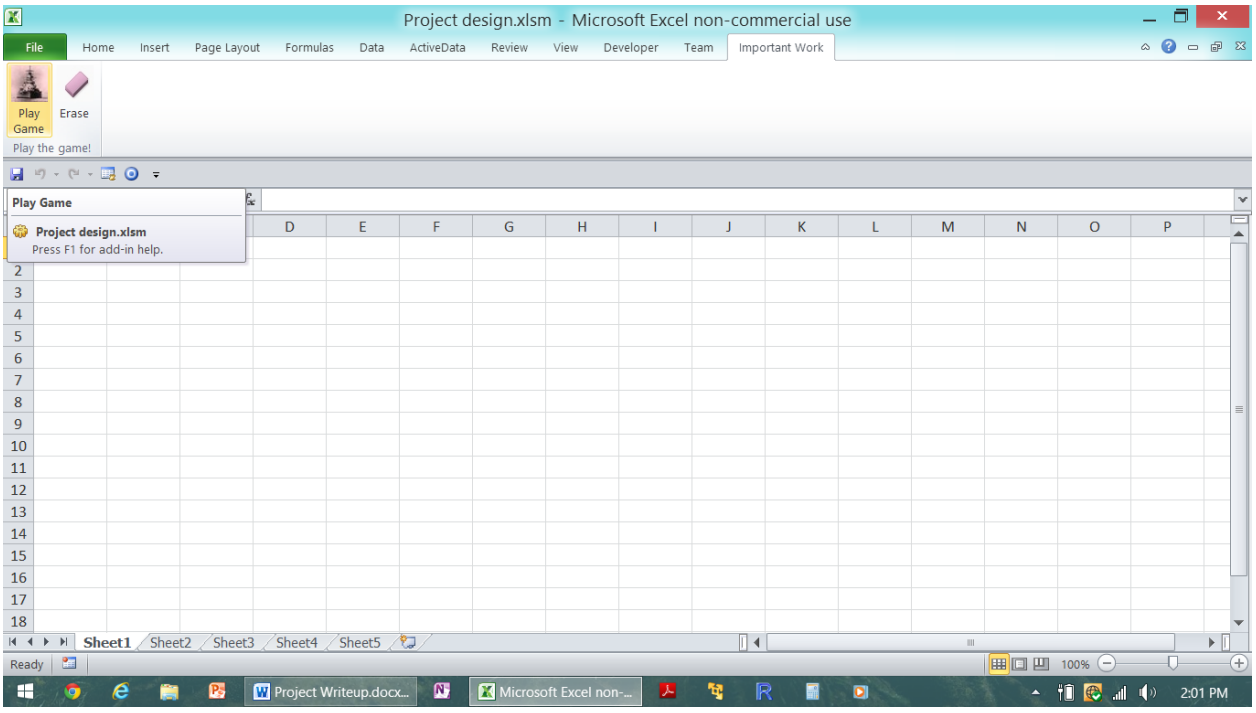
Have you ever been doing work and just need a break? Well my final project is meant to solve that exact problem. I have designed a game of Battleship that can be played in excel. If you just need a simple five minute break, you can pull up your excel file and have a little fun. My objective was to make this game so that it followed the exact rules of Battleship. I didn't want the player to be able to cheat at all. I also wanted to make sure that the computer would fire back at an intelligent level. If the computer found a hit, it would have to keep searching for the rest of the ship. If it found another ship while trying to complete the first ship it hit, it should finish off the new one then go back for the first ship it found. Although there were many complications that arose, I was able to make a working version of Excel Battleship.

As I go over this project I decided to take a bunch of screen shots as I play through a couple games to show how everything works. The first picture below is a screen shot of a normal Excel workbook. I wanted to include this picture so that you could see how the program

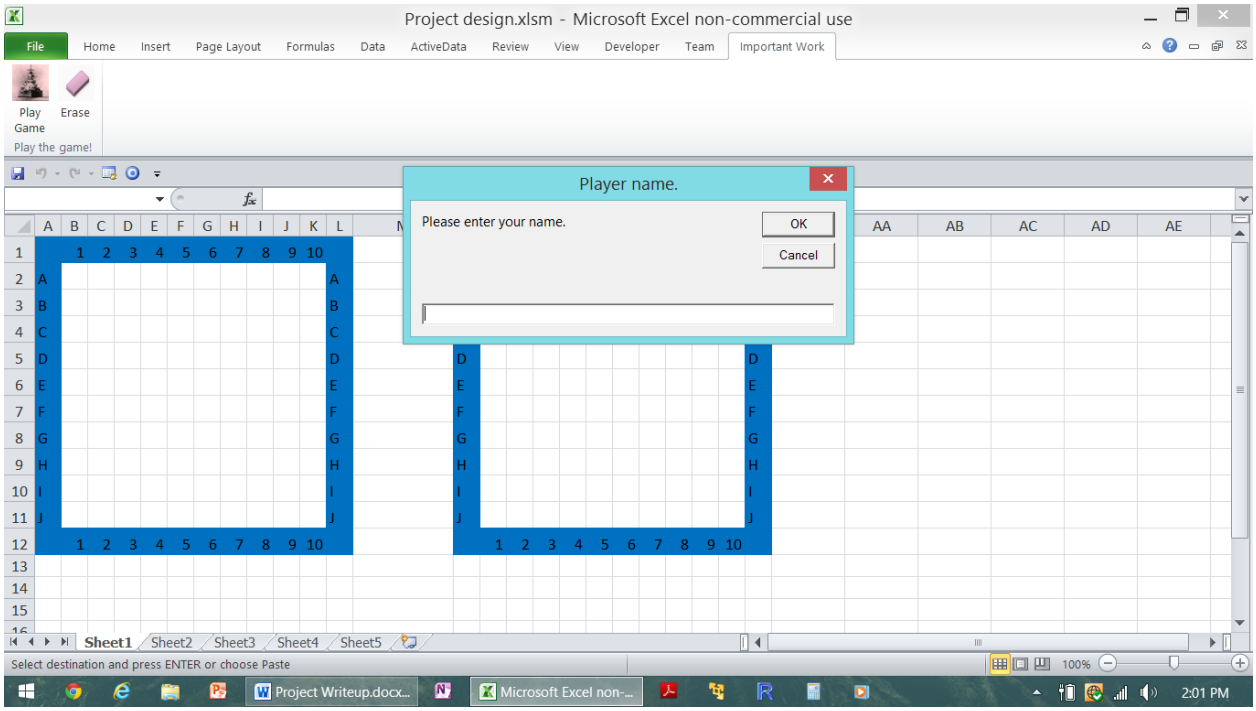


will change the workbook later. In order to get started, I would direct your attention to the upper right-hand corner of the Ribbon. I added a new tab just for the game. It is titled Important Work. This is to help prevent the boss from knowing how you get to your game. So if

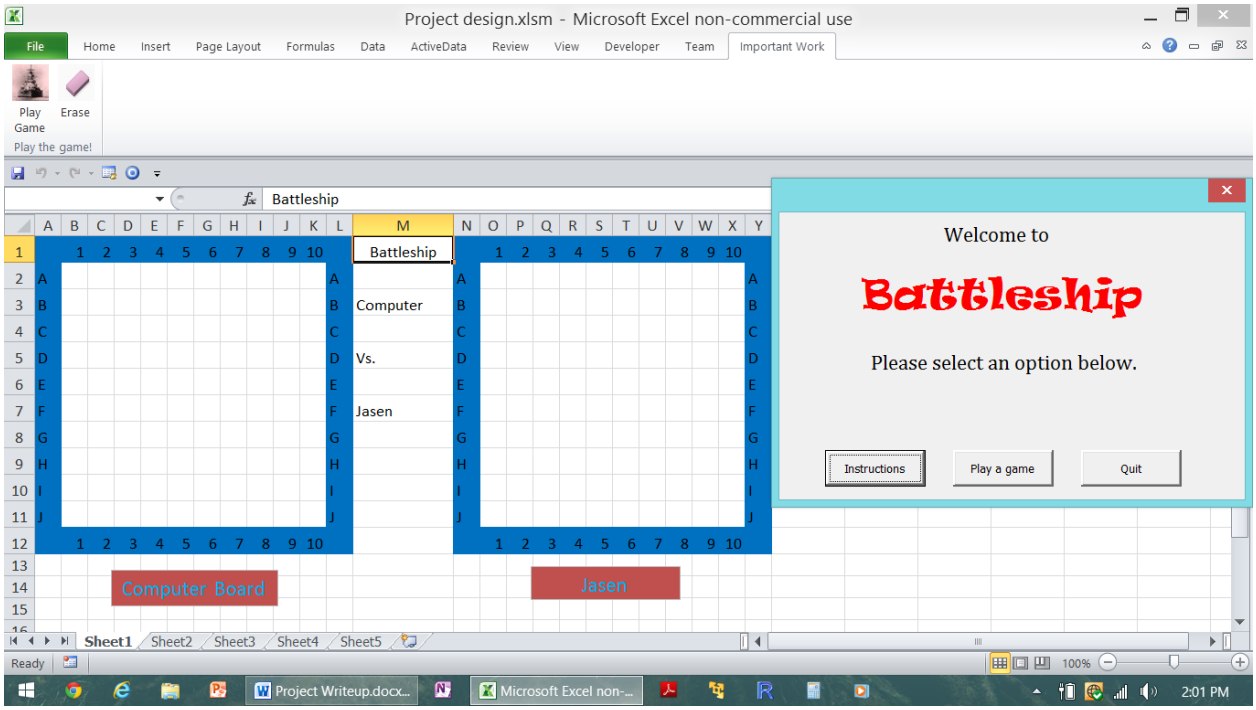




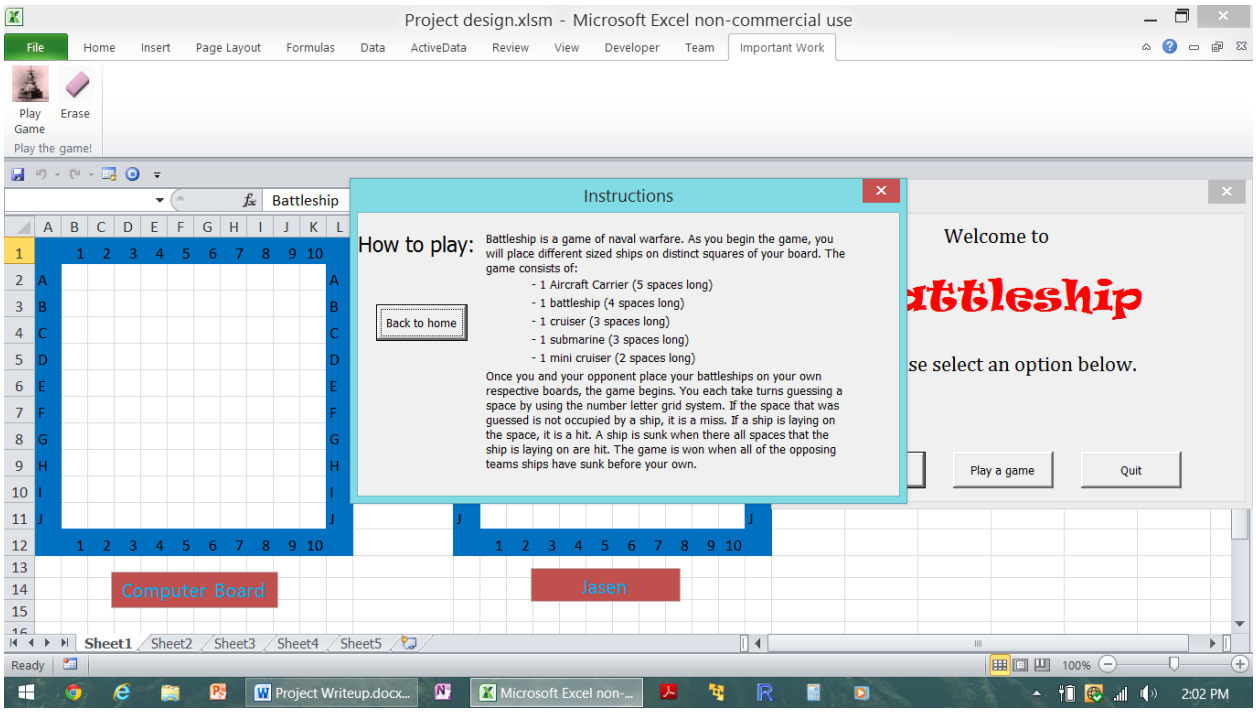
you need to be discrete, you can. If you click the Important Work tab, it opens up to two buttons. There is a button labeled Play Game and another labeled Erase. The Erase button will clear the board, and will be demonstrated later. For now we are going to just press the play



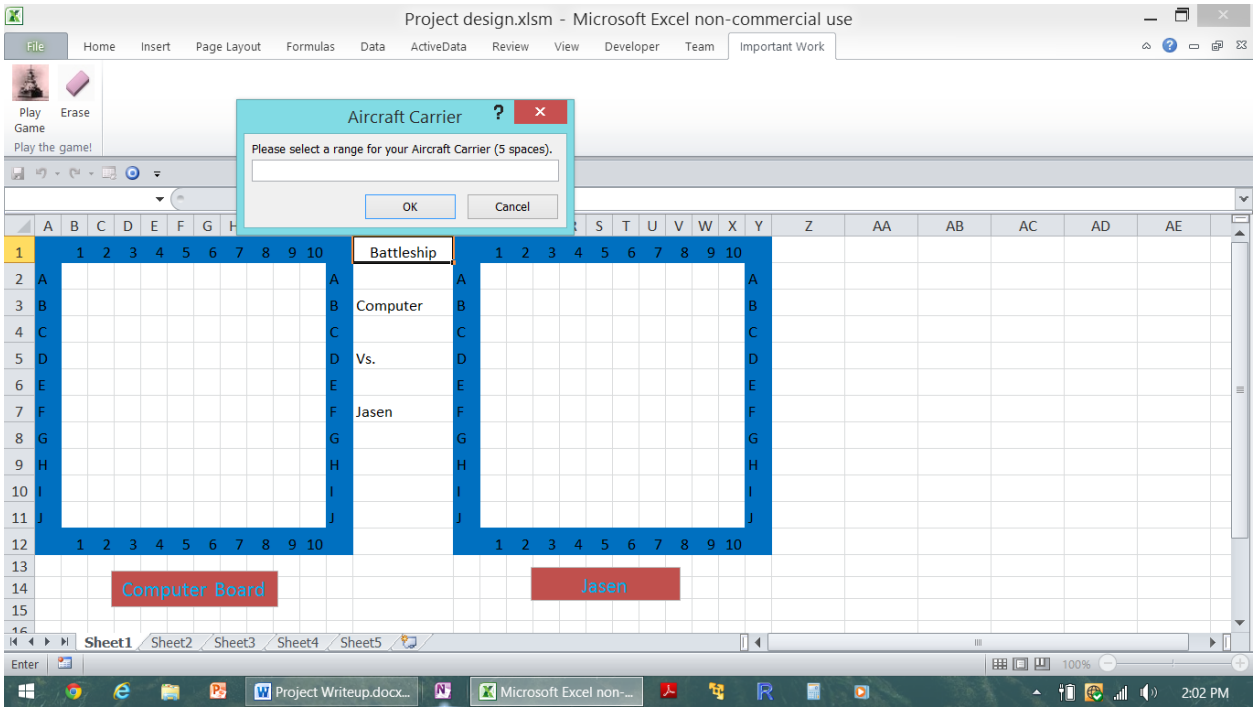
game button and have some fun! Once you press the Play Game button, the game board will automatically be created. A prompt will then appear asking for your name. You can put anything you want as the name, but for this example I will just use my own. All I did was type



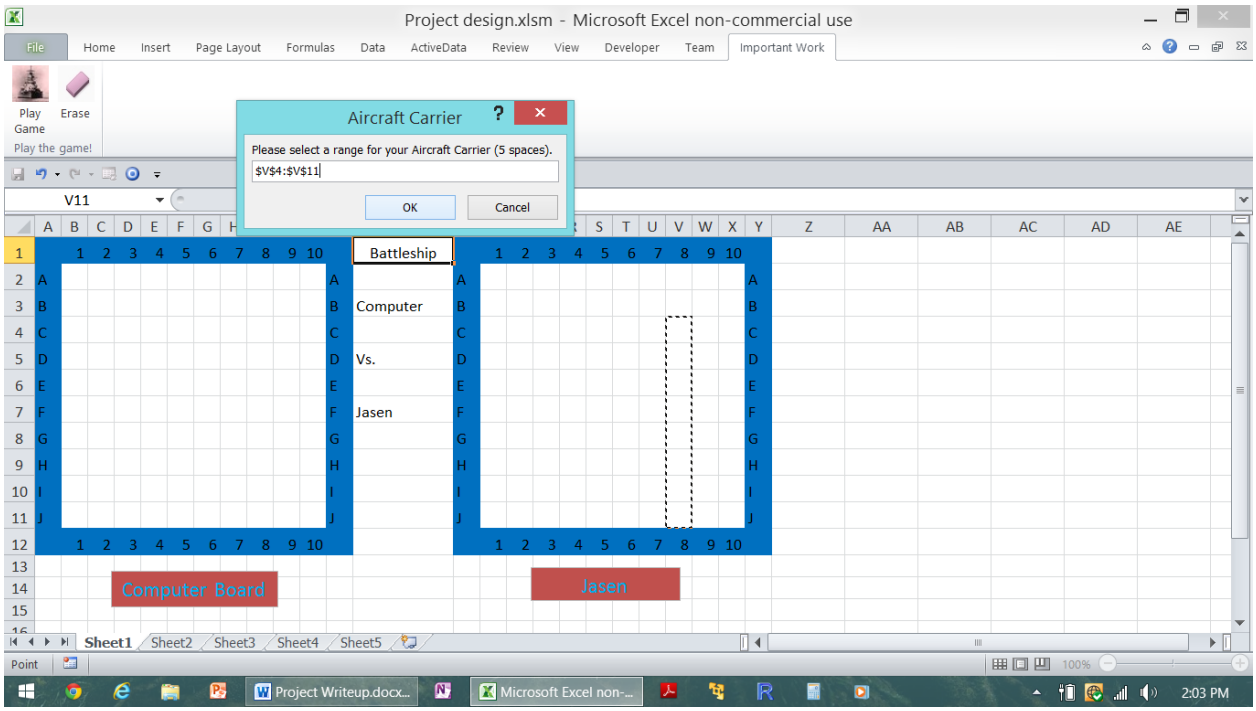
Jasen and hit okay. If you press cancel, no name will be recorded. Your name is then inserted into the game board. It will show up in the middle of the screen as well as a label showing you



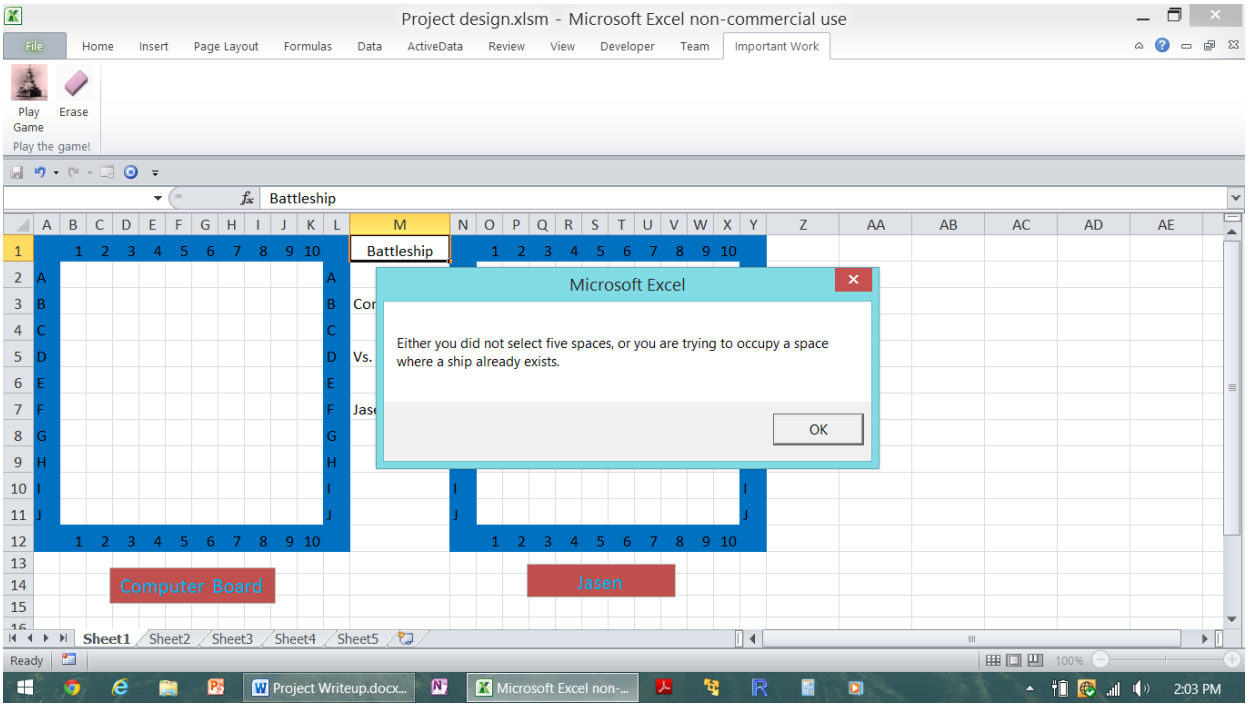
which board is yours, and which is the computers. Macros have been built to create the setup of the board. There are also two ten-by-ten arrays set up that help to hold values in the game board. I will go into this in a little more detail when I get to laying ships. As we continue, a user form pops up with three different options. The first option is an instructions button, the second



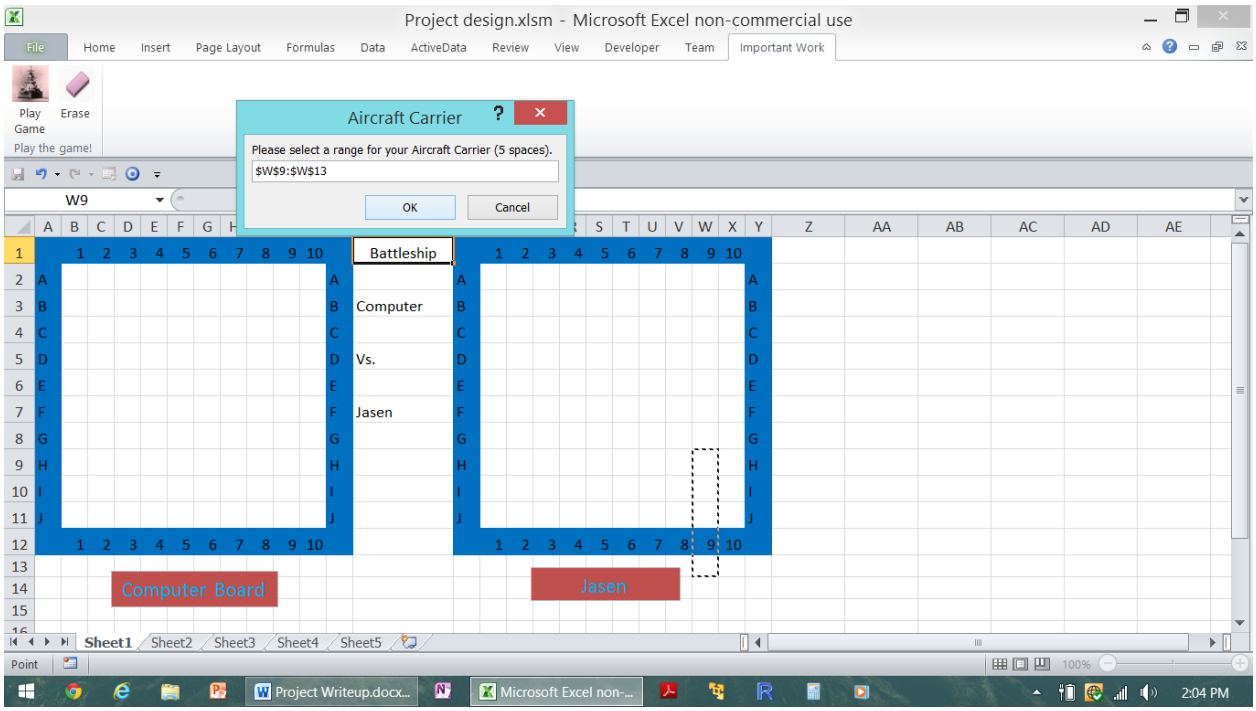
starts the game, and the third quits. When you press the Instructions button, a new user form will appear that has the instructions on how to play the game. Once you are done reading on how to play the game, you can click the Back to home button and it will take you to the first user form that popped up. From there we can begin our game by pressing the Play a game



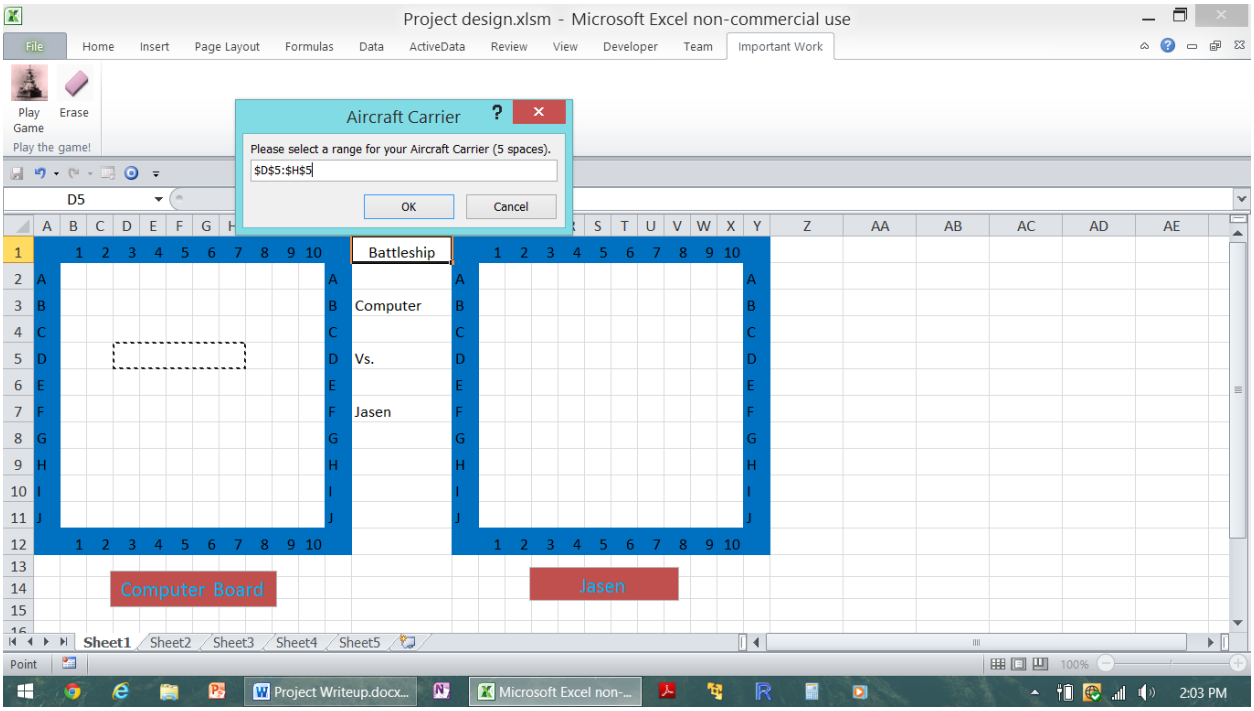
button. The first thing that pops up is a prompt asking you where you want to lay your Aircraft Carrier. Now here is where I need to go into a little detail on what is happening in the background. The ten-by-ten array mentioned earlier is initially set to contain 3's all across the board. When a ship is laid on the board, the spaces that it occupies are changed to a different number. On the opponent board, all ships are turned to a 4. On the players board (Jasen's board) an aircraft carrier is a 4, battleship a 5, cruiser a 6, submarine a 7, and mini-cruiser an 8. These different numbers are meant to help the computer know when it has sunk one of the player's ships. For example the computer will know that if it hits a five, it will need to hit four more 5's in order to sink the ship and continue on. I will continue to explain how the number system works when we get to shooting. These numbers can't be seen on the board, they are just there in the background. A few fail safes have been included here to help the player not cheat. The first one, as shown above, is that the player can't make the ship larger or shorter than it has to be. If you try to make your ship too large, or short, or overlap another ship, a warning box will show up forbidding you from continuing.



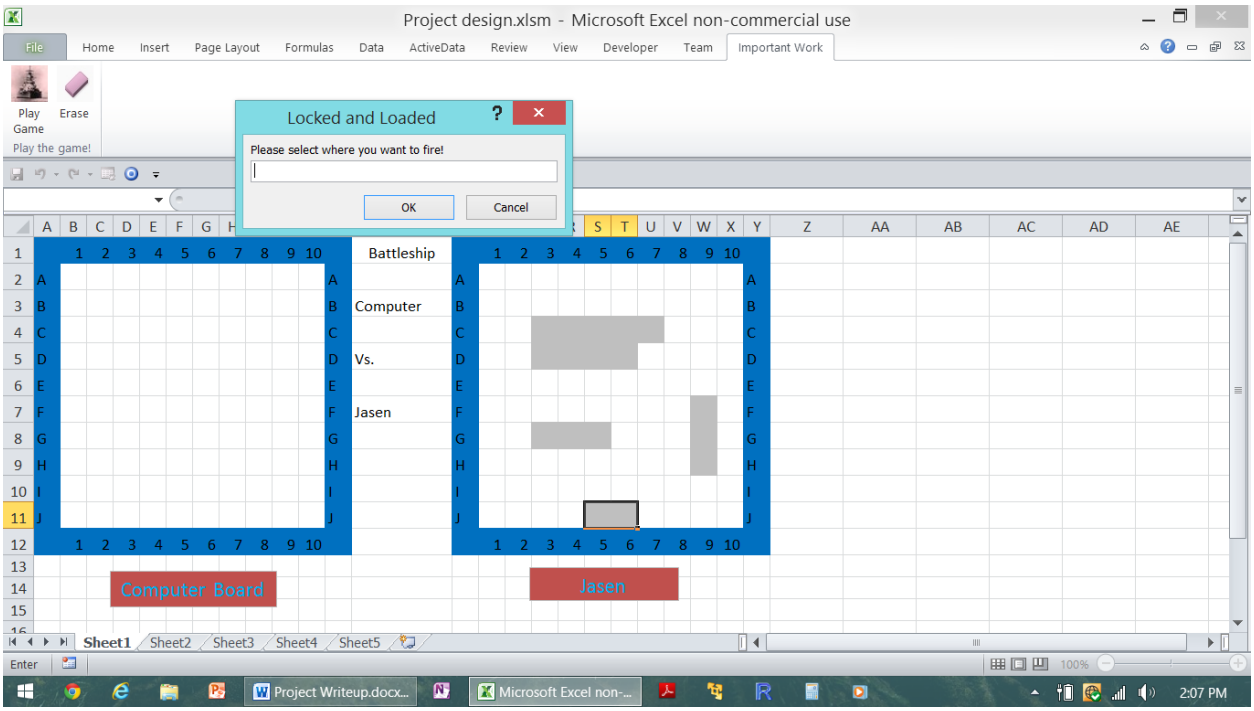
Another couple fail safes that I put are first, the player can't have part of his ship in the board and part of it out of the board.



The other fail safe I put in makes sure that the player doesn't place his ship on the opponent's board.

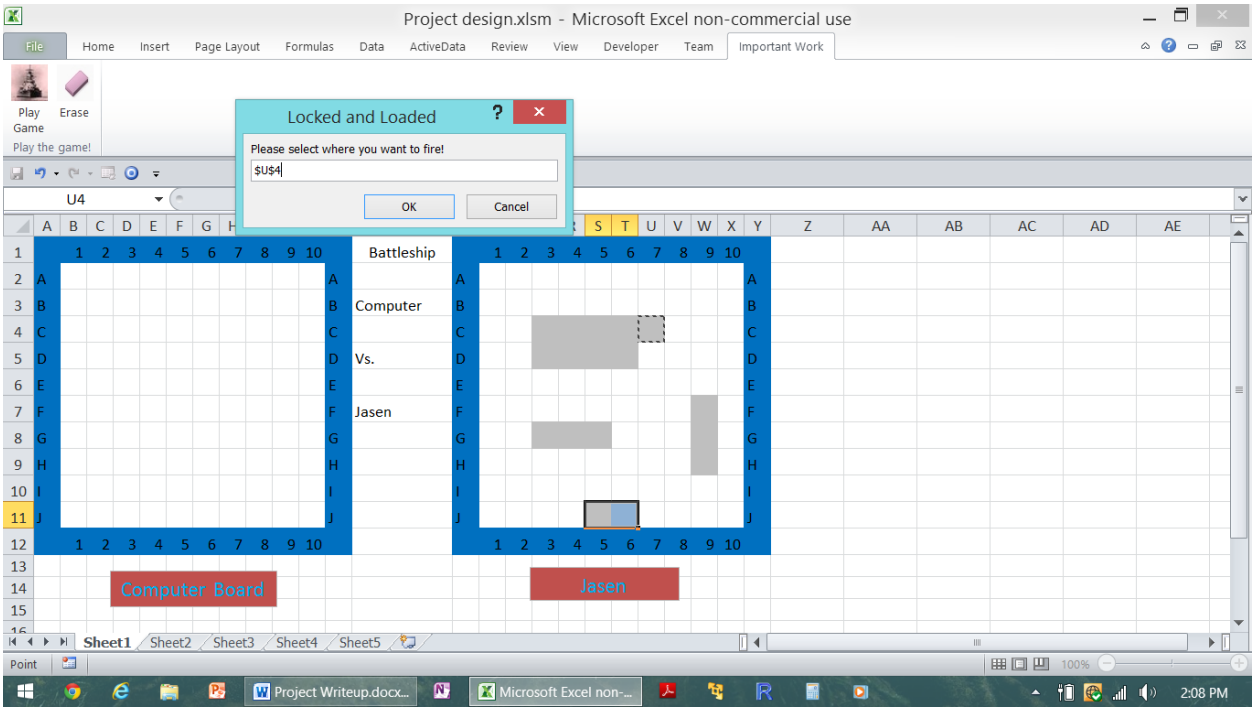


These same restrictions are applied to the battleship, cruiser, submarine, and mini-cruiser. If you try any cheating nonsense then you will not be allowed to continue. Also, once you have started the game, you can't cancel laying a ship. During this time, a random opponent board is set. The opponent's board may have connecting ships, but it won't allow ships to overlap. A random board will be created for each game. After all ships are laid down, it is time to begin the

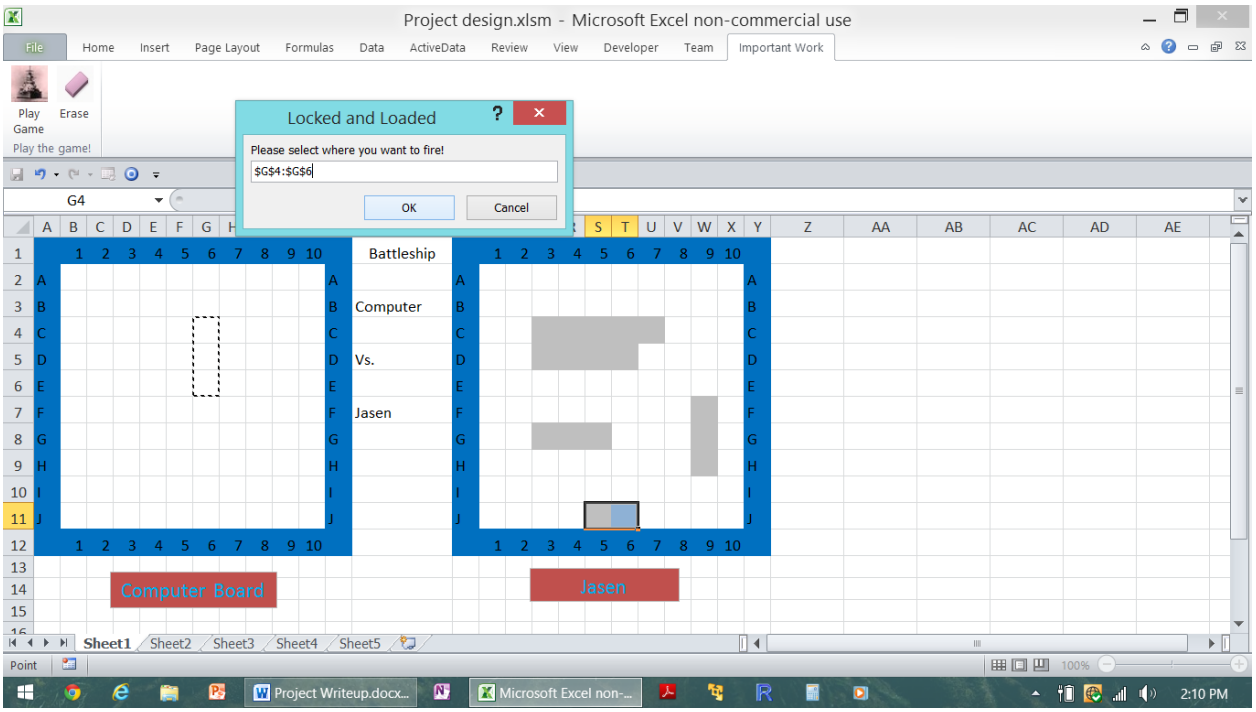


warfare!

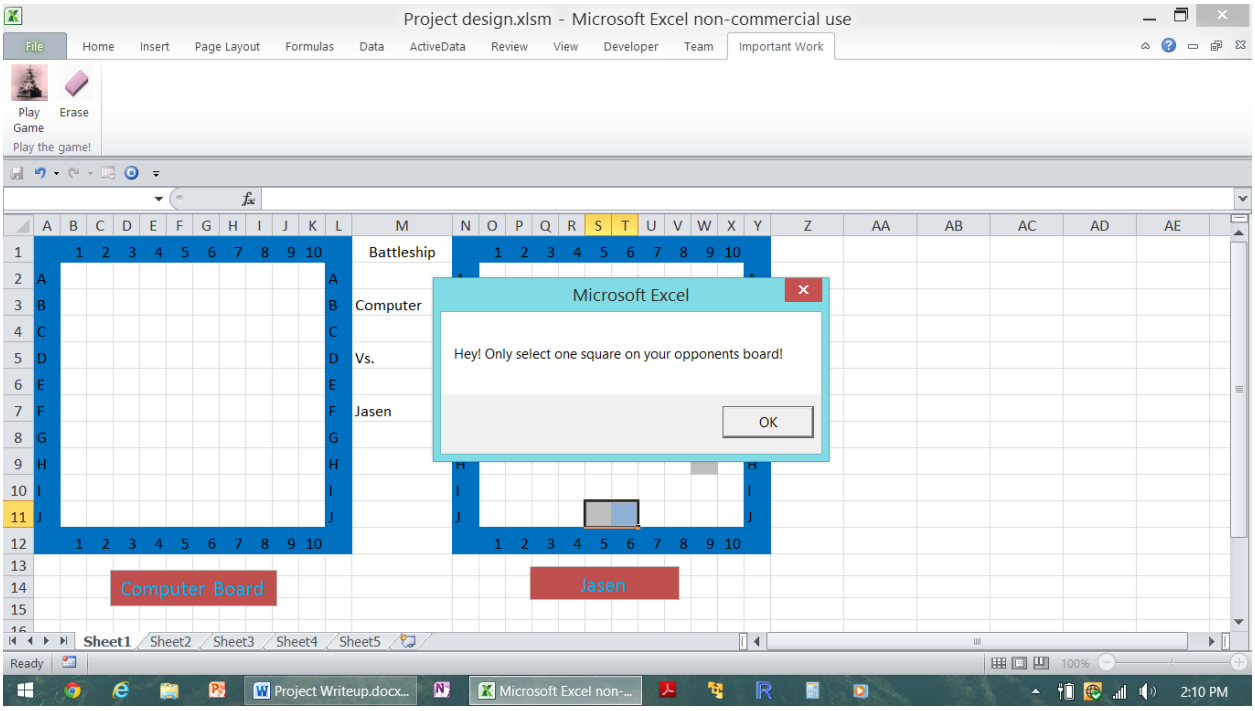
Once all ships are laid down, a prompt comes up asking for where you should fire. Once again there are some measures taken that help prevent the user from cheating. First off, the player can't shoot on his own board. I don't know why a player would want to try this, but he can't.



The second prevention helps stop the player shooting more than one square at a time. If a



player tries, a prompt will appear telling the player to cut it out. The user will not be able to



continue until they properly select a place to fire.

As I mentioned before, there is an underlying number system. If a square is selected one of a few different things happens. First the program checks to see what number is placed there. If we just shot onto the opponent's board, the options will either be a 4 or a 3. If the number is a 4, there is a ship in that square and the program then changes the square to the color red, and decreases the number of that square to a 1. The 1 helps the computer know if it has already shot at that location or not. The 1's also help the computer know when the game is over. There will be a total number of 17 ones. A brief function helps to count all the 1's on the board and determine if the game is over or not. If the player picked a shot and the number was 3, that would be a miss and the square would be changed to blue. The number of the square is then turned to a 2 so that once again, the computer knows not to shoot there again.

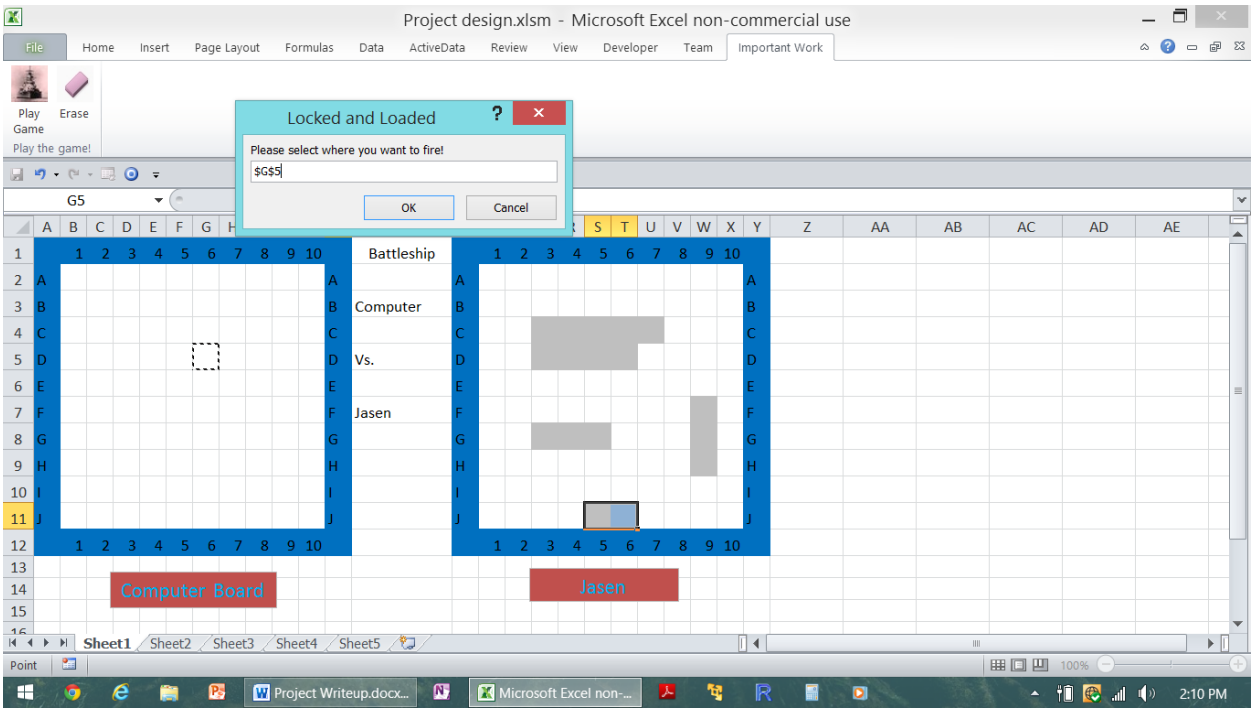
When the opponent takes a shot it becomes a lot more complicated. The program will first randomly pick a spot to shoot. The program will then check to see what number is occupying the square. Like the opponent's board, if the number is a three, it is then decreased to a 2, and the square changes blue. If the square is a 4-8, the program runs through a series of tests. If this is the first time hitting a square containing that number, a Boolean variable tells the computer that it can't randomly select a spot from the whole board on the next turn, but that it needs to randomly choose an adjoining spot to the last shot. If the number has been selected before, a variable is used to help the computer know which direction the ship is pointer. In each

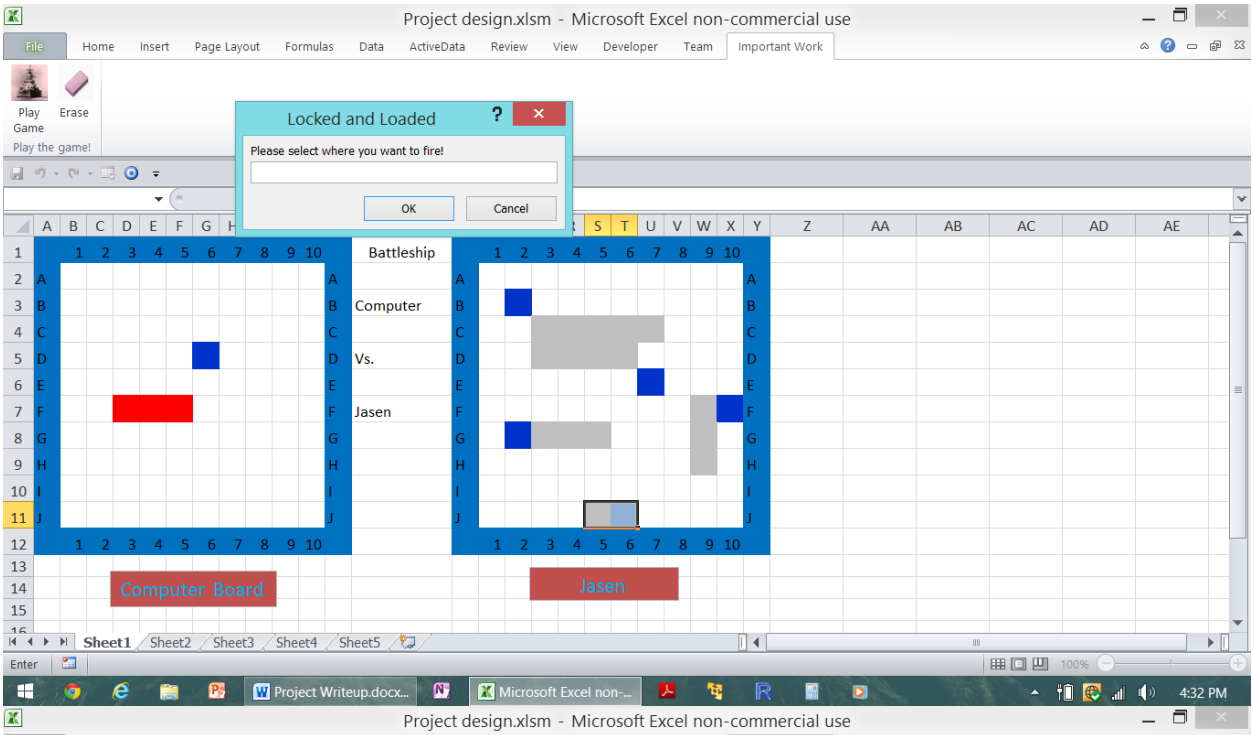
subsequent shot, the computer will continue moving in the direction that it found the second hit. If the computer finds a miss after the second hit, and the ship is not sunk, the computer will then go back to its first hit and switch directions. It will then continue in the opposite direction until the ship is sunk. Once the ship is sunk, the computer will start randomly shooting until it gets a hit again.

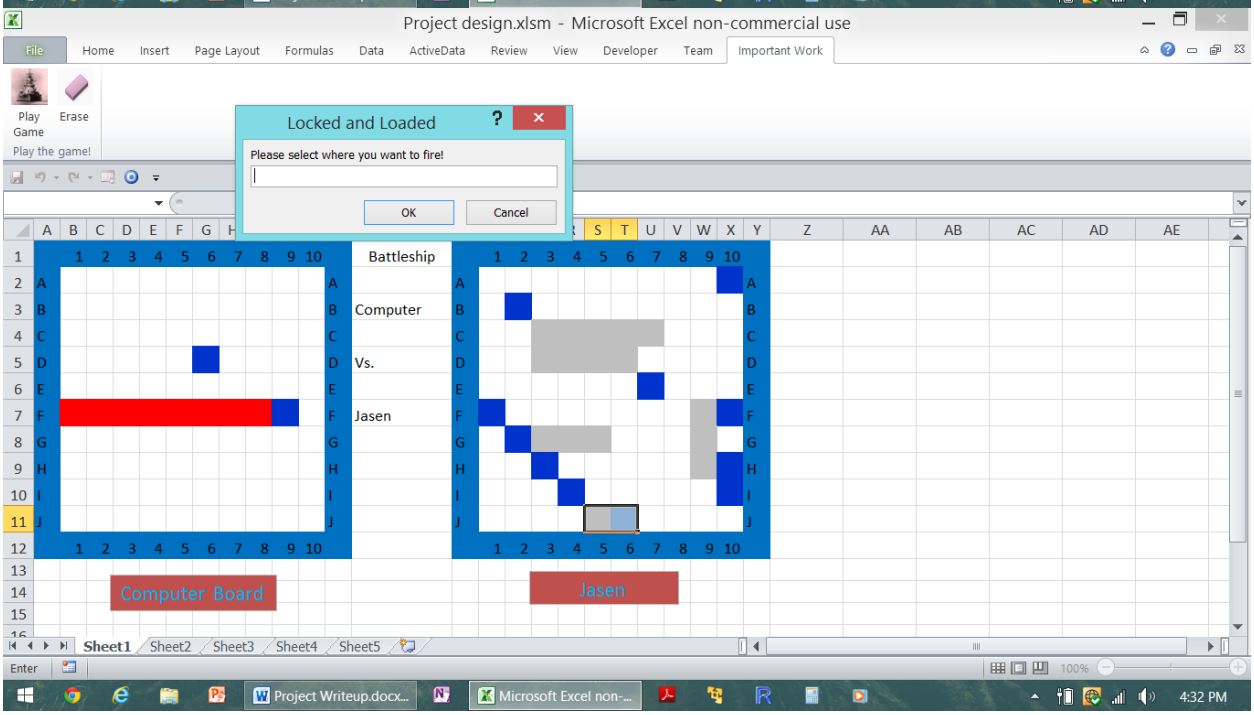
There are two scenarios that posed quite a difficult problem with the computer logic. First was if the player puts a ship on the edge their board. This was a problem because sometimes the computer would guess a shot that was outside the board, and thus outside the array created for the board. This obstacle was eventually overcome by introducing a bunch of logic that was meant to help prevent the computer from selecting outside its range.

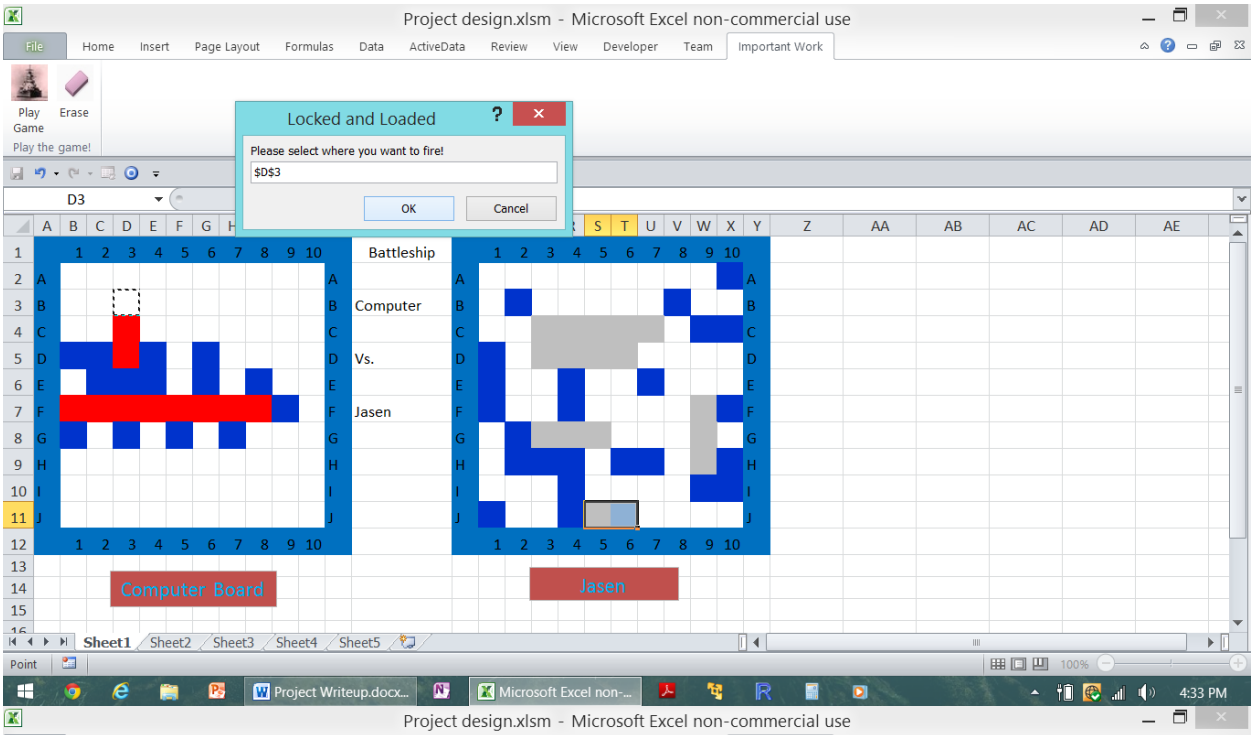
The second, and most complicated problem, was what to do if multiple ships were connected. The computer would need to know that it sunk a ship, and also know that it was another ship it needs to sink. I created a series of Booleans that helped to keep track of whether or not the ship had already been hit. Each of the players' ships is managed by their own set of global variables. These variables keep track of ship location, number of times hit, whether or not the ship has already been hit, and many others. If another ship is hit before the current ship is sunk, a Boolean variable tells the program to come back as soon as the new ship is sunk. This was a very hard problem to figure out, but it seems to work for the most part now.

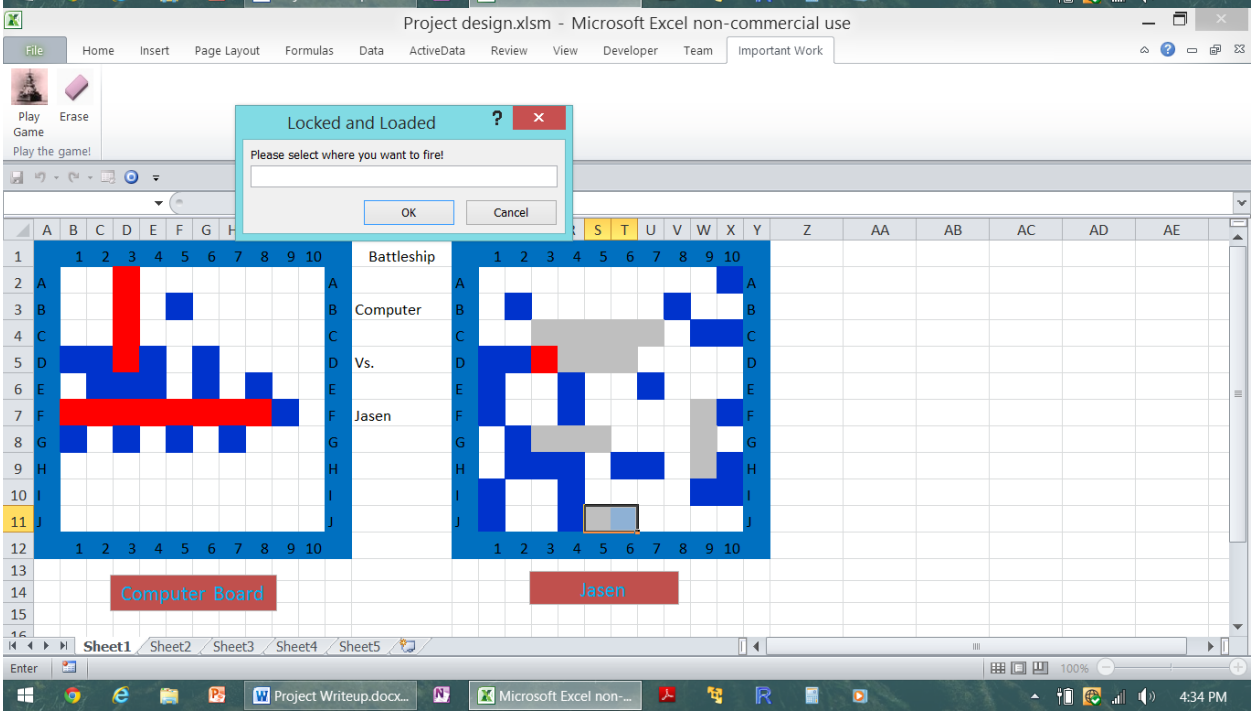
The next few pictures are a progression of a game I played and won. Commentary won't follow every picture, but once the game is over I will resume.

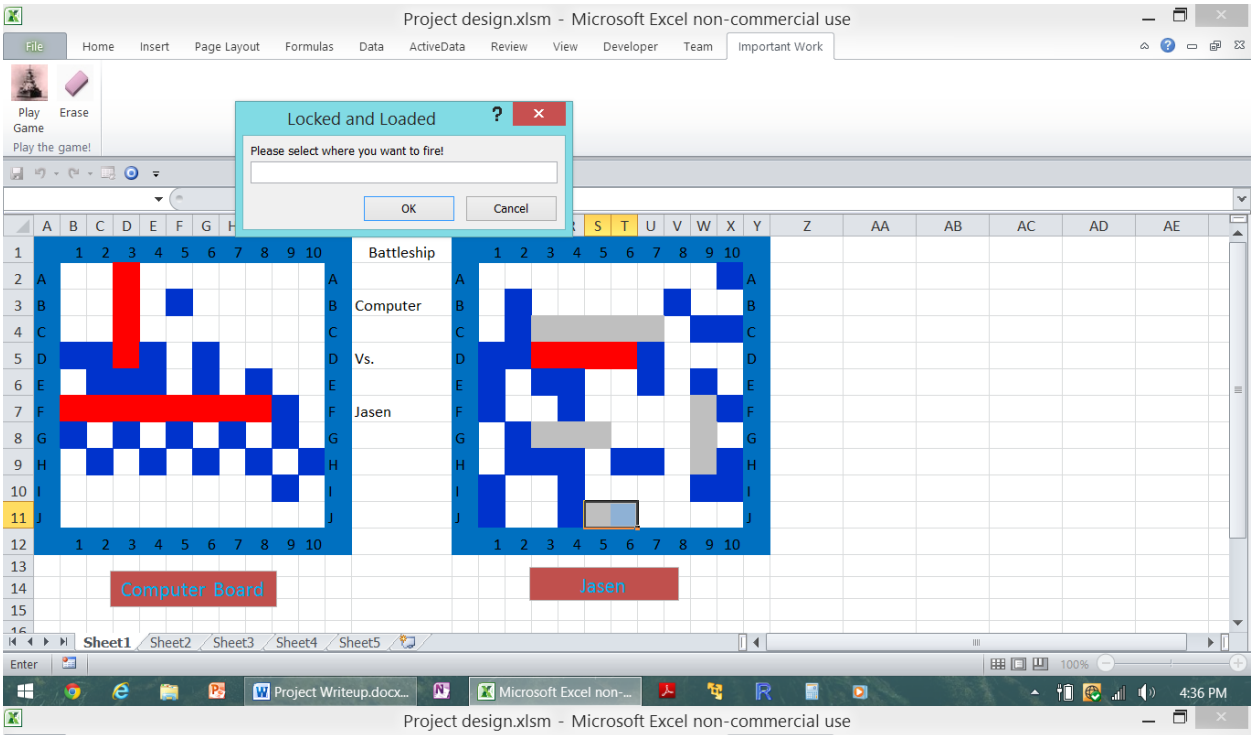


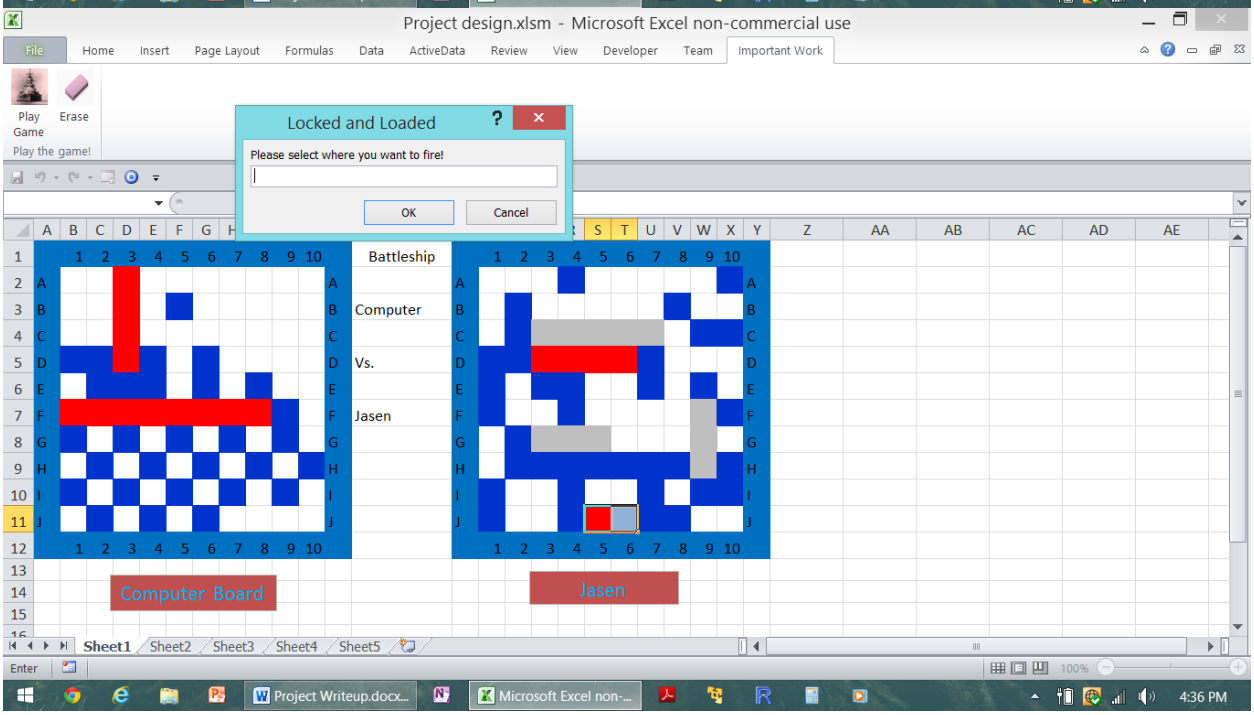

Project design.xlsm - Microsoft Excel non-commercial use
Play Game
Erase
Play the game!
Locked and Loaded
Please select where you want to fire!
OK
Cancel
Battleship
Computer
Vs.
Jasen
Computer Board
Jasen
Sheet1
Sheet2
Sheet3
Sheet4
Sheet5
Project Writeup.docx...
Microsoft Excel non-...
4:32 PM

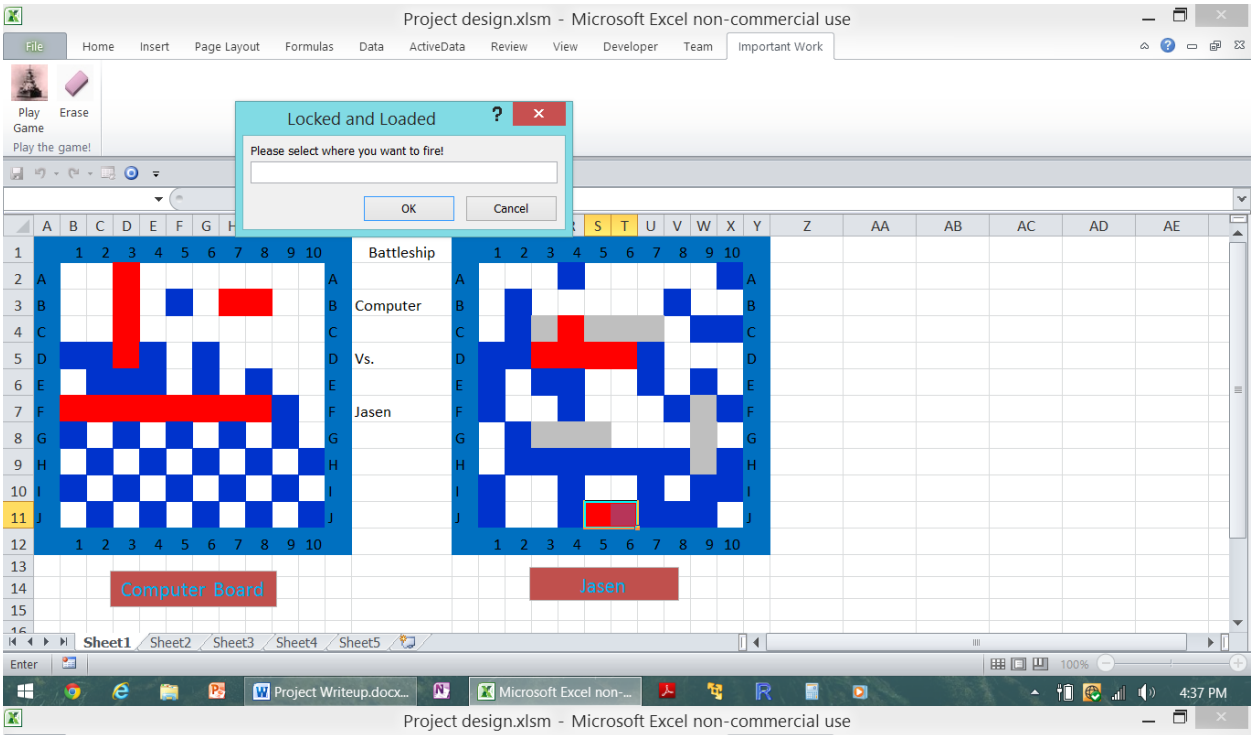

Project design.xlsm - Microsoft Excel non-commercial use
Play Game
Erase
Play the game!
Locked and Loaded
Please select where you want to fire!
OK
Cancel
Battleship
Computer
Vs.
Jasen
Computer Board
Jasen
Sheet1
Sheet2
Sheet3
Sheet4
Sheet5
Project Writeup.docx...
Microsoft Excel non-...
4:32 PM

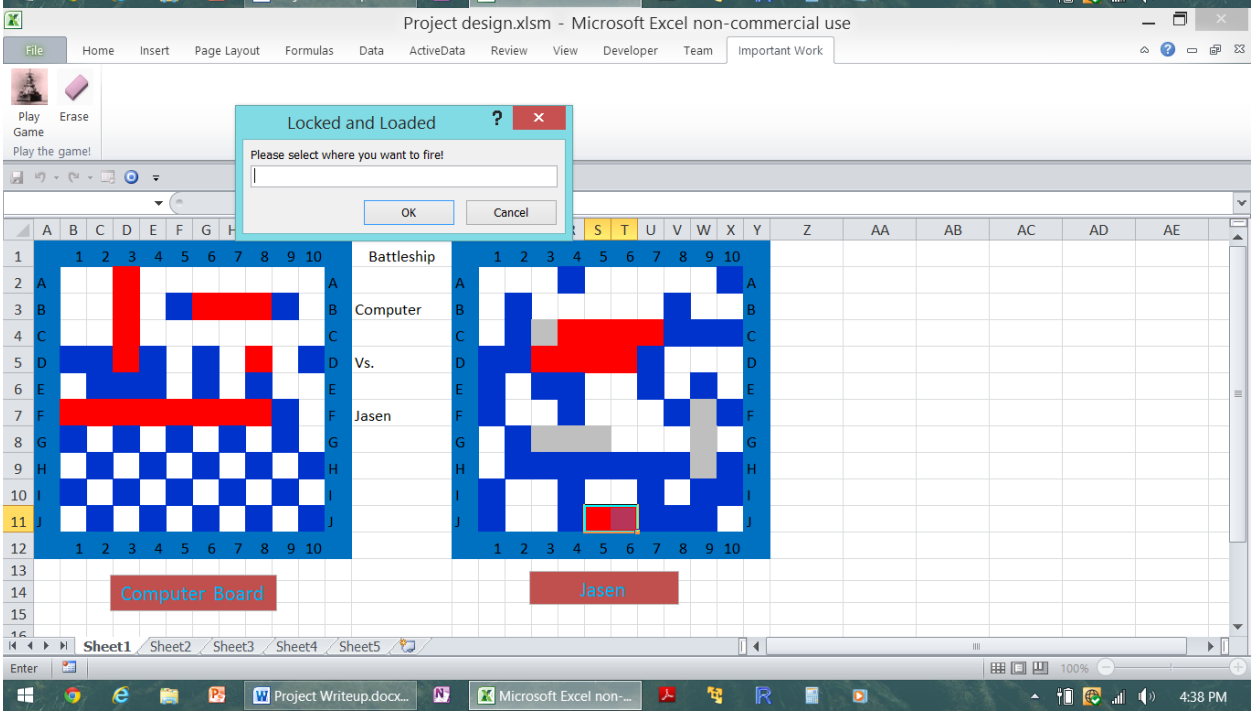

Project design.xlsm - Microsoft Excel non-commercial use
Play Game
Erase
Play the game!
Locked and Loaded
Please select where you want to fire!
D3
OK
Cancel
Battleship
Computer
Vs.
Jasen
Computer Board
Jasen
Sheet1
Sheet2
Sheet3
Sheet4
Sheet5
Point
4:33 PM

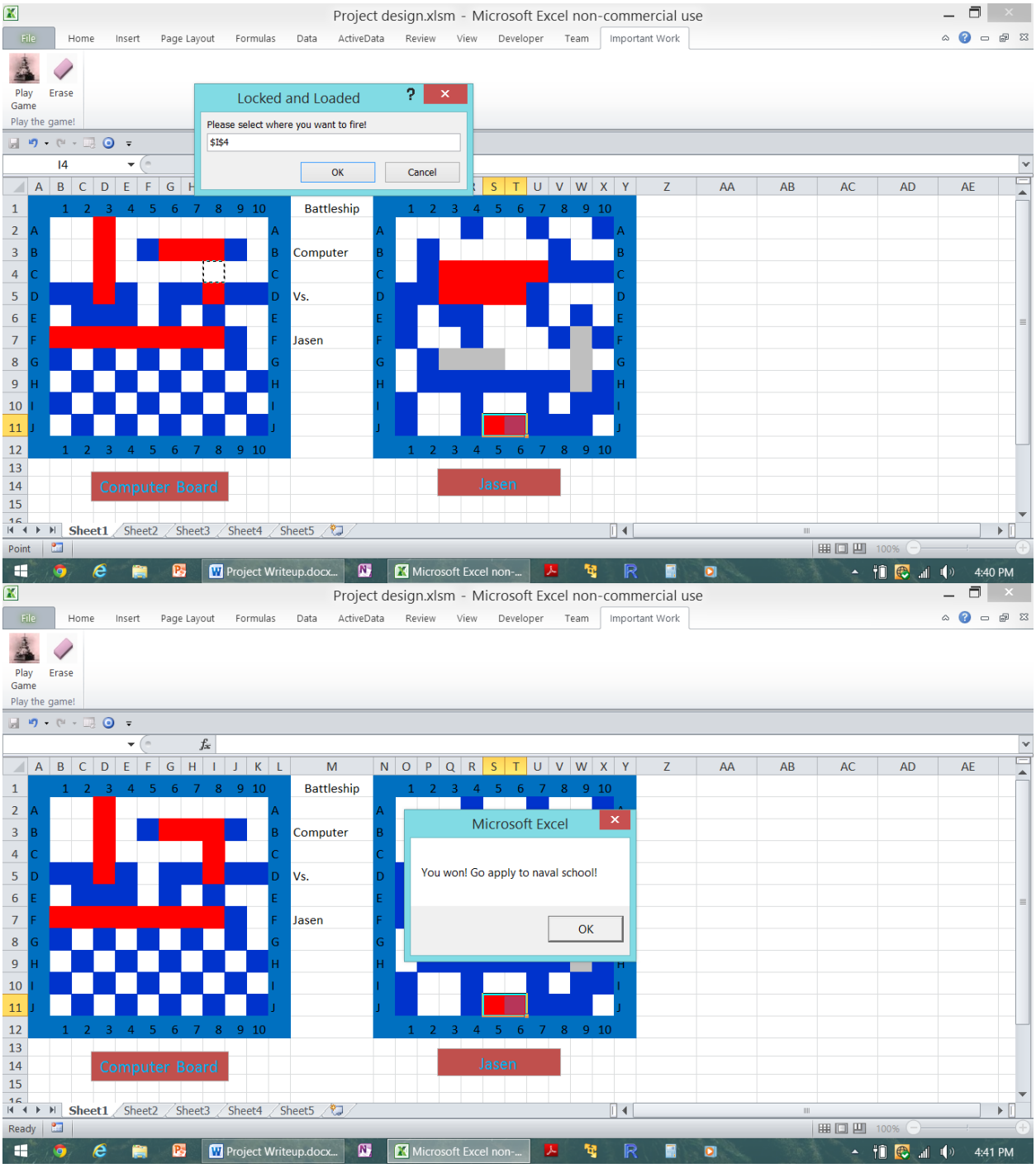

Project design.xlsm - Microsoft Excel non-commercial use
Play Game
Erase
Play the game!
Locked and Loaded
Please select where you want to fire!
OK
Cancel
Battleship
Computer
Vs.
Jasen
Computer Board
Jasen
Sheet1
Sheet2
Sheet3
Sheet4
Sheet5
Enter
4:34 PM


Project design.xlsm - Microsoft Excel non-commercial use
Play Game
Erase
Play the game!
Locked and Loaded
Please select where you want to fire!
OK
Cancel
Battleship
Computer
Vs.
Jasen
Computer Board
Jasen
Sheet1
Sheet2
Sheet3
Sheet4
Sheet5
Enter
4:36 PM

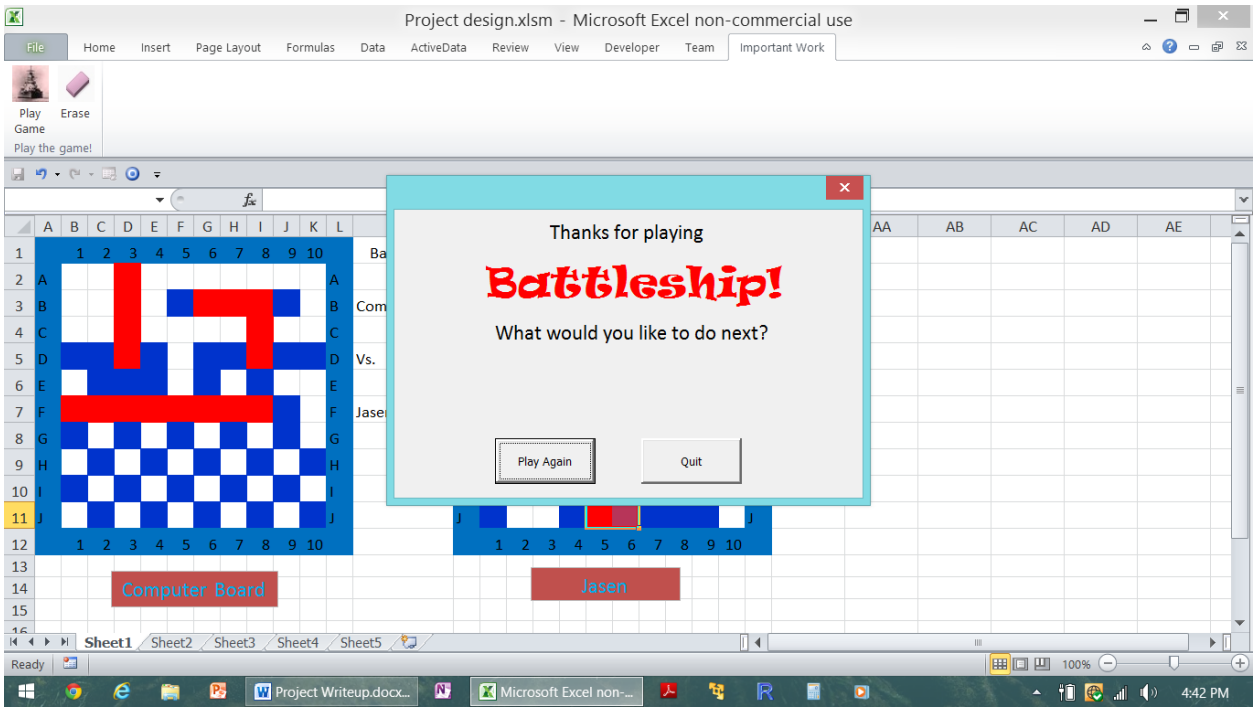

Project design.xlsm - Microsoft Excel non-commercial use
Play Game
Erase
Play the game!
Locked and Loaded
Please select where you want to fire!
OK
Cancel
Battleship
Computer
Vs.
Jasen
Computer Board
Jasen
Sheet1
Sheet2
Sheet3
Sheet4
Sheet5
Enter
4:36 PM


Project design.xlsm - Microsoft Excel non-commercial use
Play Game
Erase
Play the game!
Locked and Loaded
Please select where you want to fire!
OK
Cancel
Battleship
Computer
Vs.
Jasen
Computer Board
Jasen
Sheet1
Sheet2
Sheet3
Sheet4
Sheet5
Project Writeup.docx...
Microsoft Excel non-...
4:37 PM

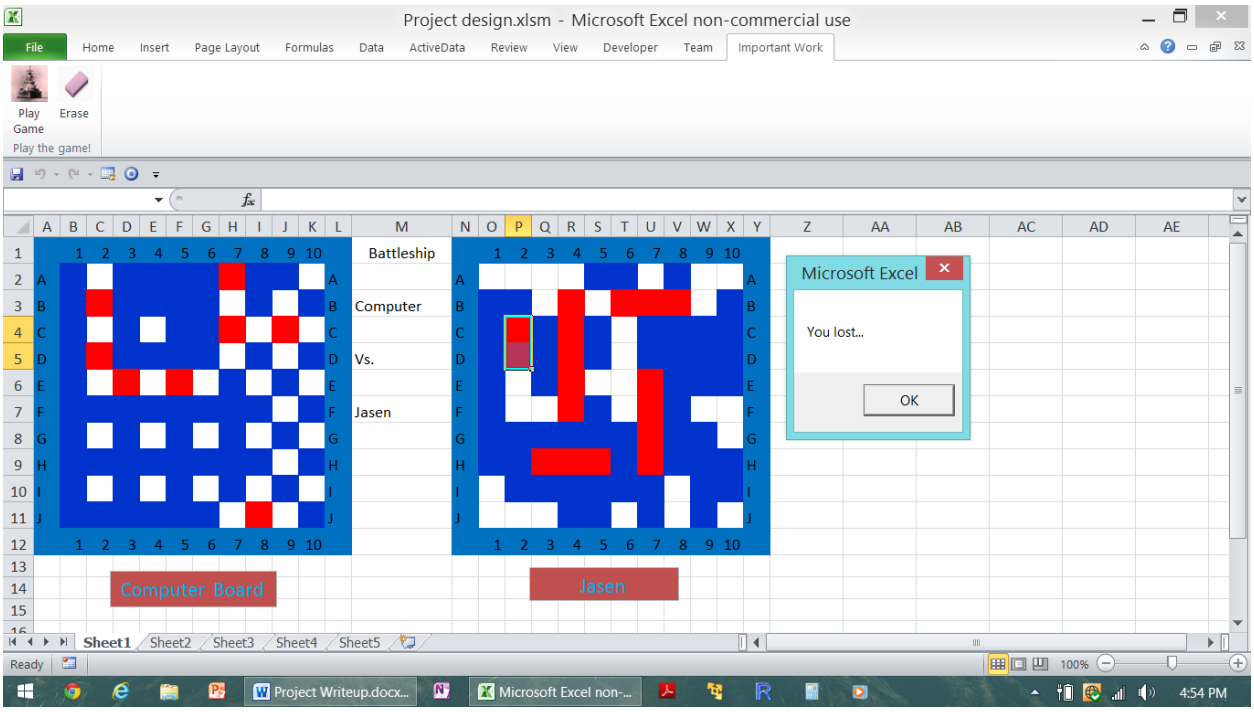

Project design.xlsm - Microsoft Excel non-commercial use
Play Game
Erase
Play the game!
Locked and Loaded
Please select where you want to fire!
OK
Cancel
Battleship
Computer
Vs.
Jasen
Computer Board
Jasen
Sheet1
Sheet2
Sheet3
Sheet4
Sheet5
Project Writeup.docx...
Microsoft Excel non-...
4:38 PM



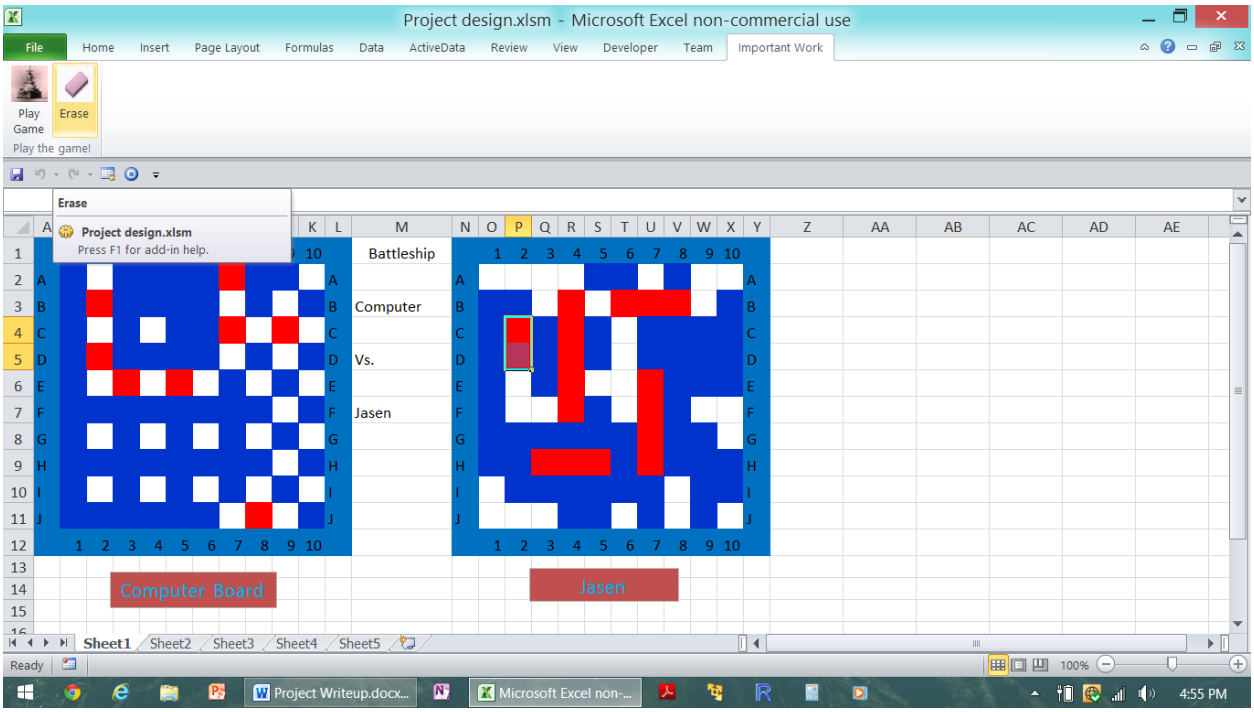
As you can see, I just won. When you win, a box pops up congratulating you. After you hit ok, a user form shows up asking if the player if they want to quit or play another game. If you hit quit, the game is over, and the user form goes away. If you hit the Play again button, the whole process begins again.



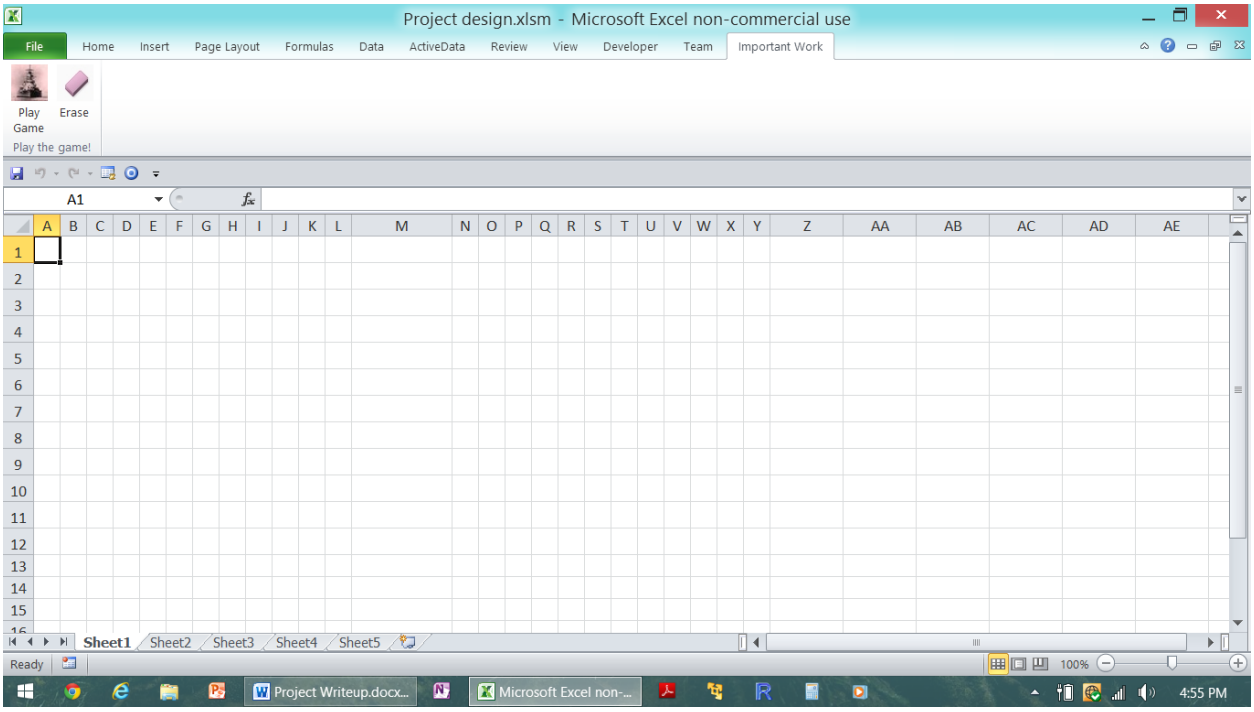
I won in the example, but if you lose you get a box that says “You lost…” and then the play again user form pops up. Just like the following.



This is pretty much how the function works. It is built using a lot of sub procedures and functions. There is one more thing, the Erase button. After the game ends, the board and labels are still sitting around, just like the picture below.



If you hit the erase button, everything is erased, just like the following.



This is in essence, my project. It was challenging and very time consuming, but I felt like I learned a lot from it.

Some of the things I learned were an intensive amount of logic, user forms, debugging, global variables, and many other useful things. I needed to know how each piece of the puzzle fit together to help the function run smoothly. I did run into a lot of problems when dealing with the computer coming back to finish off a ship that it had already hit. Even still, the logic doesn't always work. I would say that the program works about 90% of the time, but every once in a while, I hit a glitch. I believe I can find the solution, I just ran out of time. It ended up taking me a very long time to make sure that the program ran smoothly. I didn't end up asking either the professor or TA for help. Looking back I probably should have so that I would have had more time. I just really wanted to figure it out all on my own. I did a lot of my own debugging to help find the solution to my problem. I also used a lot of google research and used the textbook a lot to help me. I feel like it should work just fine, but there must be a little problem or two in the code. So if you run the program, and run into a problem, you can just start it again and most the time it works just fine.

As I mentioned before, I wanted to do this project all on my own. Thus I did not have any assistance while I worked on this project. I guess the only assistance I did have was some reminders on how to do certain things from google and the textbook. Usually it was just syntax questions on different formulas. I really had a fun time doing this project. I really liked the lessons I learned and the ending result. I actually think I will use this program. I think if I had the time to improve it, I would make sure there are no bugs, have different difficulty levels, and make it two player on the same computer, and two player through the internet. I will probably add these things just for fun someday, but for now this version will have to suffice. I hope you enjoyed this project and are able to have a little fun with it as well.