# PA5 - Project report

I created the Battleship board game. The board is 10x10 squares. The program randomly positions the ships for both the player and the AI. There are 5 ships on the board. One 5 cell ship, one 4 cell ship, two 3 cell ships, and one 2 cell ship. The player plays against an AI that has two levels of difficulty. In easy mode, the AI shoots randomly at new coordinates. In hard mode, when the AI hits a ship, it will shoot around those coordinates to find the next hit.

Each item that they fulfilled from their project proposal:

Make an entirely new program with similar methods to some previous assignment.

- This is an entirely new program.
- 60 points

Make the program take simple decisions (if then else)

- The easy mode fulfils this item.
    - I used a rand to get new coordinates to hit and check if the coordinates have already been hit. If they have already been hit a new random coordinate, would be created.
- 20 points

Make the program take complex decisions (search)

- The hard mode fulfils this item.
    - The AI has 2 search modes, hunt mode, and target mode. Hunt mode is just like easy mode, which is randomly shooting to find ships. Target mode starts when hunt mode hits a target. The initial coordinates are added to a queue and the AI will shoot around that coordinate. When it hits again the new coordinates are added to the back of the queue. When the AI has shot at all the coordinates next to the first hit, it will then remove it from the queue and use the next coordinates in the queue. When the queue is empty the AI will go back to hunt mode to find the next target.
- 40 points

Each significant implementation item that students added but isn't in the original proposal:

Make the program take simple decisions (if then else)

- The gameboard fulfils this item.
    - The program prints out the gameboard with grey X for when you miss, red X when you hit and blue X when the whole ship has been destroyed. It also prints out green S for the cells that your ship cells are in.
- 20 points

Any specific problems students ran into

- Had some problem with marking a ship destroyed when all the cells that the ship occupied were destroyed.

Any interesting solutions students found or came up with and implemented into their program.

- To solve the problem of knowing if the ship is destroyed. I created pointers to both the ship class and ship cell class in the cell class. The ship class has a vector of its ship cells. So, when a cell with a ship cell in it is shoot. It goes into the ship class and checks if all the ship's ship cells have been shot. If so, the ship has been destroyed.

I believe the project is worth 100 points.