Battleship

Object Oriented Programming Project

Marc-Antoine Danner • IG3

BATTLESHIP®
THE CLASSIC NAVAL COMBAT GAME
LE JEU CLASSIQUE DE COMBAT NAVAL
EL CLÁSICO JUEGO DE COMBATE NAVAL
AGE • EDAD 7+
2 PLAYERS • JOUEURS • JUGADORES
ADULT ASSEMBLY REQUIRED • DOIT ÊTRE ASSEMBLÉ PAR UN ADULTE • REQUIERE SER ARMADO POR UN ADULTO
CA3264
Hasbro Gaming

Introduction

This project was introduced to the class as a way to learn OOP (Object Oriented Programming) with Java. It's divided in two big parts, the first one is creating a battleship game as we use to play as a board game and the second one is creating an AI (Artificial Intelligence) which would play for us. Additionally we were asked to create a CSV file with the results of our AI games as a proof of their performances. We were asked to use Java language as an OOP language which is used world wildly. We were given only few informations about some classes to put in Java and that's all. In this document I will try to explain how I have done this project. I am not sure that when you will read this, the project will be ended and I don't think it will because we all know we can always improve our programs. But with this document you will be able to understand how I decided to work and how I imagined my program. It will show you how my project could and should work.

Instructions

It's a very simple game to launch, all you have to do is open a terminal on your laptop, go to the Battleship.jar directory and type the following command "java -jar BattleShip.jar", assuming you have java installed on your machine ;)

Architecture

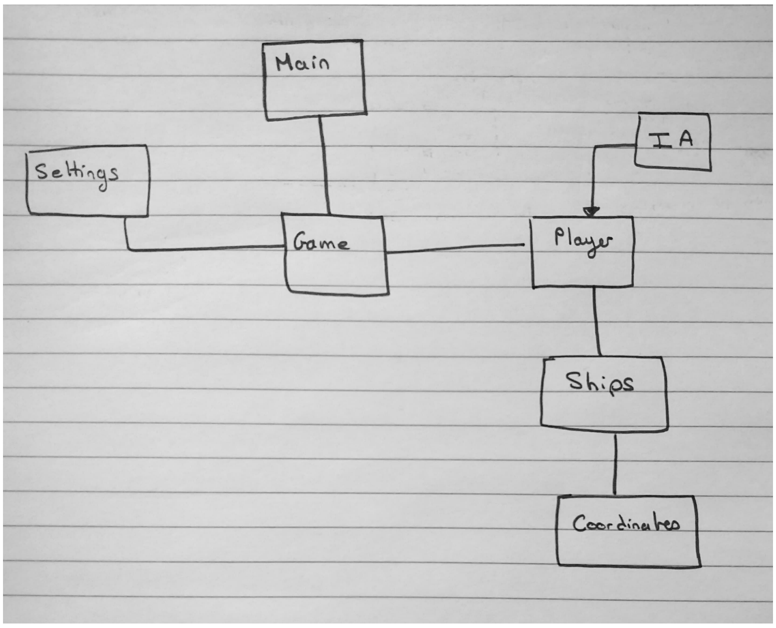
My BattleShip Architecture is quite simple, I think everyone knows the game well and it's not difficult to determine the classes. I have 9 classes in two packages as I was asked to do. In the first Package you can find anything you need to play to Battleship and in the second one you find anything you need to verify the IA's. Moreover, the Battleship rules or quite simple as well and translate them to algorithmic language was not so hard. Here is the game how it works. The most difficult part is to translate it to java when you don't know this language. Here is the way I see the game playing:

- Game start
    - We create two entities, two player for this game

- Each player places its ships one by one (Player1 then Player2)
    - Everytime we check that the ships are on the Grid and don't overlap another ship
- (We choose randomly which player start to play)
- First Player begin
    - The player guess and gives a position (ex "E6")
        - We check if it's a valid position (on the grid)
        - We check if it's a new shot
            - If not we inform the player that he lost a shot
        - We check if there is a ship at this position
            - If not, the current player change
            - If the ship has lives remaining
                - If it is the player last ship, the ship is sinked (we check if it is his last ship or not, the end of the game or not)
                - If it is not, the ship loose a live and we change the current player
        - We check if the opponent player still have ships in his fleet before we continue. If not the game ends

For me the most difficult part is not the reasoning but the translation to Java because clearly I had no idea of how I can display a grid for the human player or how I compare strings and int. Moreover, where I had a lot of difficulties was how to place my function.If I have to check if a ship is on a grid: should I do it on the grid class or on the ship class ? And finally there is always the public/private, static, final, void, this, super etc which where difficult because even if I have read the definitions and other informations about it, thanks to Google, it's still too abstract to me.

Talking about Pros and Cons, I admit I dont't know what to say, I know my program is not perfect, with this exceptions for example but for the architecture it's the way I have seen it so I don't have enough "recoil" to see what I should change for now… But I'm sure that with the next OOP classes I will see all the mistakes I have made and how foolish I could have been.



Here you can see the UML graph I made for my game, it changed a lot of time and could change another one time if needed.

Artificial Intelligences

Even if I didn't end well my project with AI, I was really curious about it because it's something really new for us. Dealing with AI might be something we will do in the future and learning it now is a good way to understand step by step how it works. So I decided to make some researches about it and it's really interesting all the articles you can find on the internet and how many ways there are to think about an AI.

Before I go further I will quote all the articles and sources links I have read:

- https://www.csuohio.edu/sites/default/files/82B-2016.pdf
- http://www.datagenetics.com/blog/december32011/
- https://github.com/GrahamBlanshard/AI-Battleship/blob/master/prograham/battleship/player/AIPlayer.java

The First One is only dealing with probability density but it is still interesting and gives you a method on how to use it efficiently.

The Second One is more interesting and more attractive to read, it shows a lot of graphs and game boards so you can see how the algorithms have evolved to go to a median of 97 shots to win (in 100 millions simulations) to a 42 median with optimised AI. It shows different methods and the utilisation of mathematics and statistic to resolve it. It is something I would love to use in my program in the future.

The Last one is a Java example of AI for battleship. I have tried to understand it but again my lack of Java knowledge wasn't great and I spent more time on forum to try to understand it than the code himself.

Dealing with my AI, I choose, due to a lack of time, to do the "basics" things. The first AI is only shooting randomly on the map and that is all. It might be considered as very simple but when I heard we had to do three AI, I chose the easiest way… The second one is not sophisticated as well, I only reused the first one and added a memory so it doesn't shoot twice at the same coordinates. And for the third one I wanted to improve the second one as well by adding neighbours coordinates so when you hit a ship, it doesn't go far away until the ship is sunk. But unfortunately I was not able to make it work on time… I obviously haven't managed my schedule and the task I had to do, and I am a little bit upset about myself because I really wanted to make it right but I hope to work on it on my own in the future.

Java

As I said, we were asked to work with Java. I have to say that I was a bit scared about this new language and this new paradigm of programming because after that course I have had the opportunity to work with it. Moreover we were given this project without a lot of lessons and we were told to learn by failing (I may be exaggerating but that is the way I felt) so I was really lost at the beginning. But thanks to Google I have been able to find a lot of solutions to issues I faced.

POST MORTEM

Facilities

The easiest part for me was the beginning, when you have to think with just your brain, a pen and a paper. It was the most interesting part for me. Also I loved to learn about OOP and this new paradigm of programmation. It was really nice to learn vague concepts in English on forums and then have a real explanation in classes with you. I know there are a lot of things to learn about it and I am excited to know more about it.

Issues

As you can guess, the most difficult part was java. It is something totally new for me and even if it's popular and used by a lot of people, It is not a language I love. I know it's old and it has proven it's efficiency but still…

The other issue was the time… I clearly managed my time badly and I'm really unhappy about it. We had a lot of work to do and instead of putting deadline to one project, finish it and take another project I sadly chose to do them all together which wasn't clearly a good idea when I see I didn't finished any of them the way I wanted to…

Also about the code I had multiple issues :

I didn't have enough time to use the exceptions we talked about in classes, the notion was still not acquired and I preferred not to use them instead of using them badly.

Also I still have issues to launch the game well… issues I already had before and that I had fixed but due to git problems I had to redo what I had done.

It was not something totally bad because it allowed me to have a new vision with all I could see with my classemates (which I thank a lot because they helped me a lot for conception and especially for java with the structure, the list and the conditions). I tried to minimise the cloning but it was hard when you are stucked on something for hours and your friend just near you have the solution !

What I learned

From this project I learned a lot of things. Firstly, the initiation to OOP which pushed me to learn more about it and in different languages. Secondly it gave me the opportunity to learn more about java and as you said to me "C'est en forgeant que l'on devient forgeron" (which we can translate by "Practice makes perfect" by the way). And finally it made me want to do more programmation… It could sound "brown-noser" (sorry for the vulgarity) but before this project we always had boring programmation projects with multiples rules… But with this project I learned to make research on my own, and even if I had hard time just to compare strings for example, I learned to like it. I can clearly see the change of the classes in engineering school and I love the idea of having similar projects in the future !