Projet AOOP : Bataille naval

Projet du semestre 4 MPCIE Informatique

# Sommaire :

### [1-Introduction](#_1-Introduction)

### **2-Conception du jeu**

**2.1-Les classes de bateau**

**2.2-Les classes de jeu**

**2.3-Les classes d’interfaces**

**2.4-Les fonctionnalités**

**2.5-Les petits plus**

### **3-Conclusion**

# **1-Introduction**

Le projet consiste à concevoir et développé un jeu de bataille naval, pour ceci j’ai utilisé le langage Java et j’ai utilisé le logiciel Eclipse pour développer.

Pour ce jeu il m’a été demandé :

-différent bateau de différentes tailles

- un placement automatique et aléatoire suivant certaines règles

-la création d’un adversaire suivant une Intelligence artificiel

-différentes bombes avec différents effets

-la possibilité de réinitialiser le jeu

-la possibilité de sauvegardé la partie actuelle

-et si possible de rajouter des fonctionnalités supplémentaires

Pour ce projet j’ai mis environ 3 semaines à le concevoir et le développé.

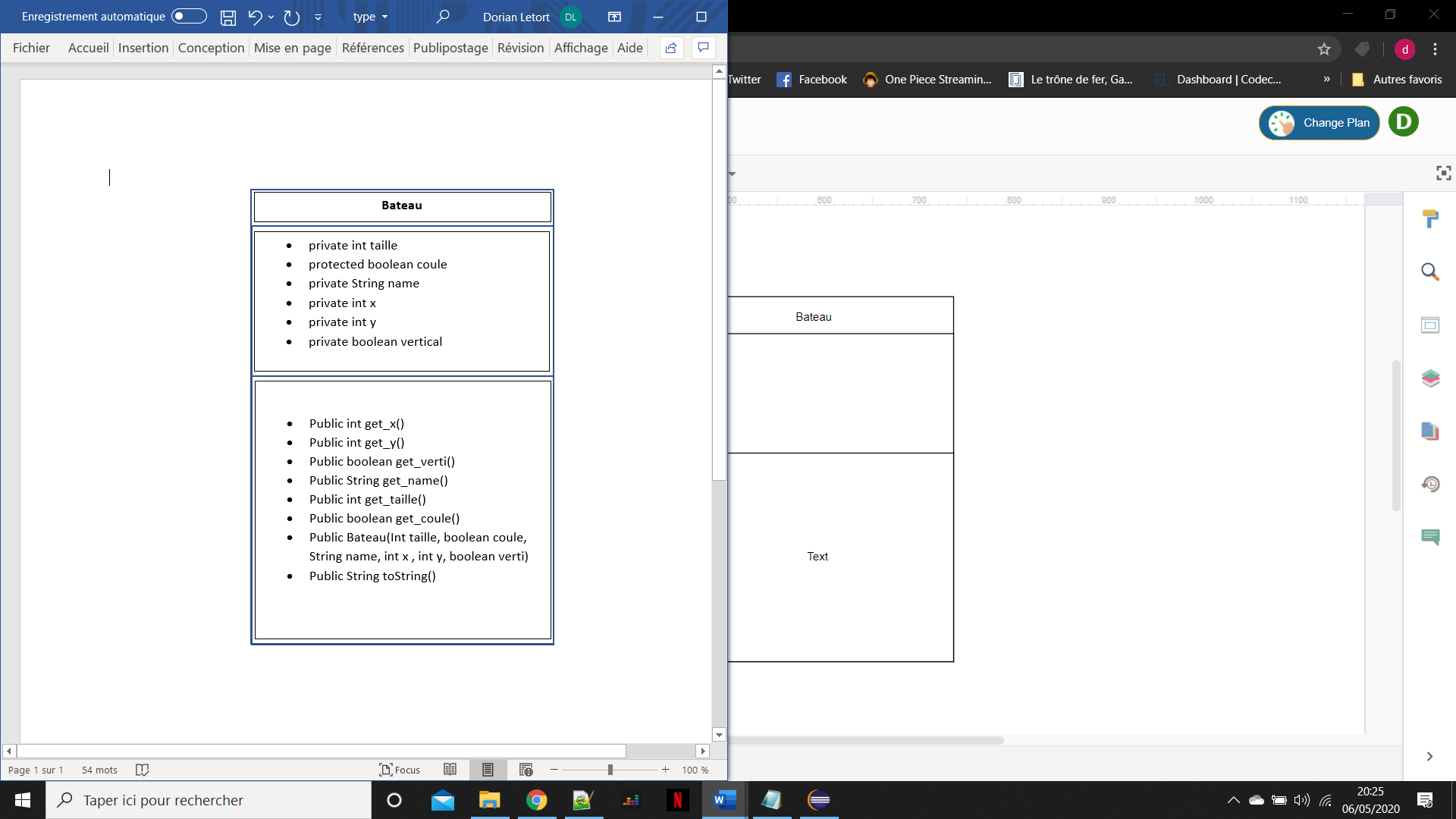
Passons maintenant à la conception et à son développement.

# **2-Conception du jeu**

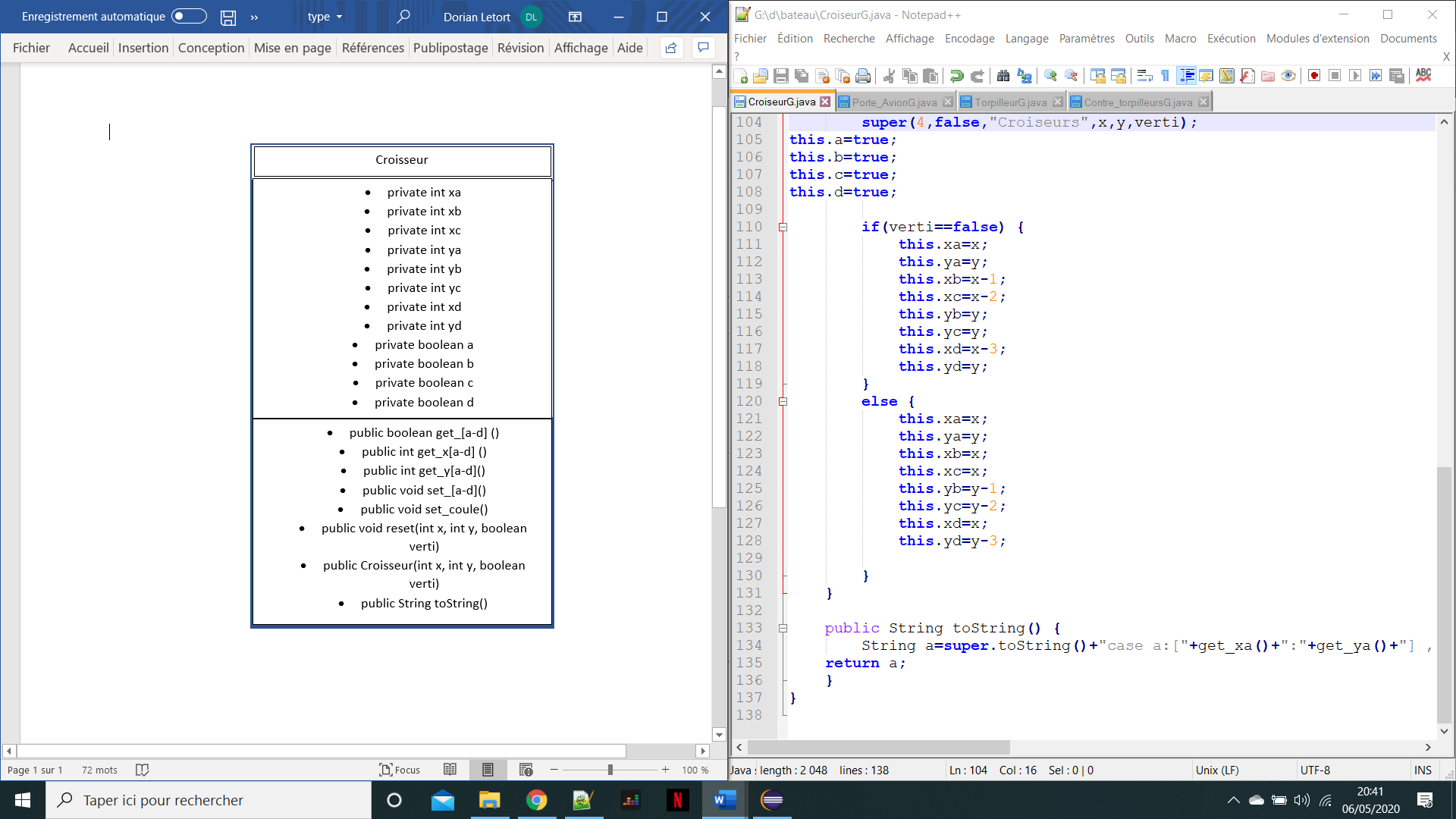
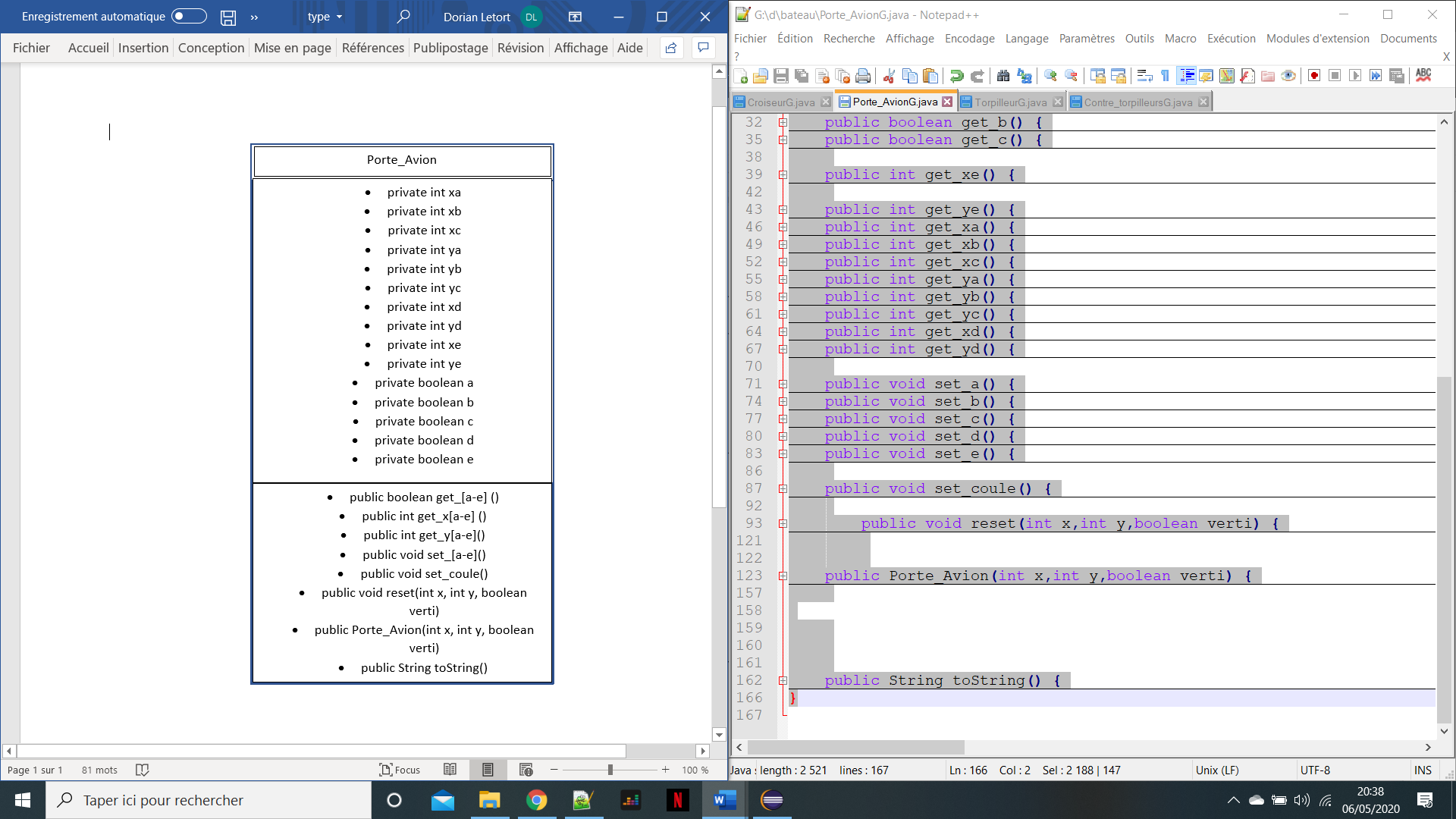
## 

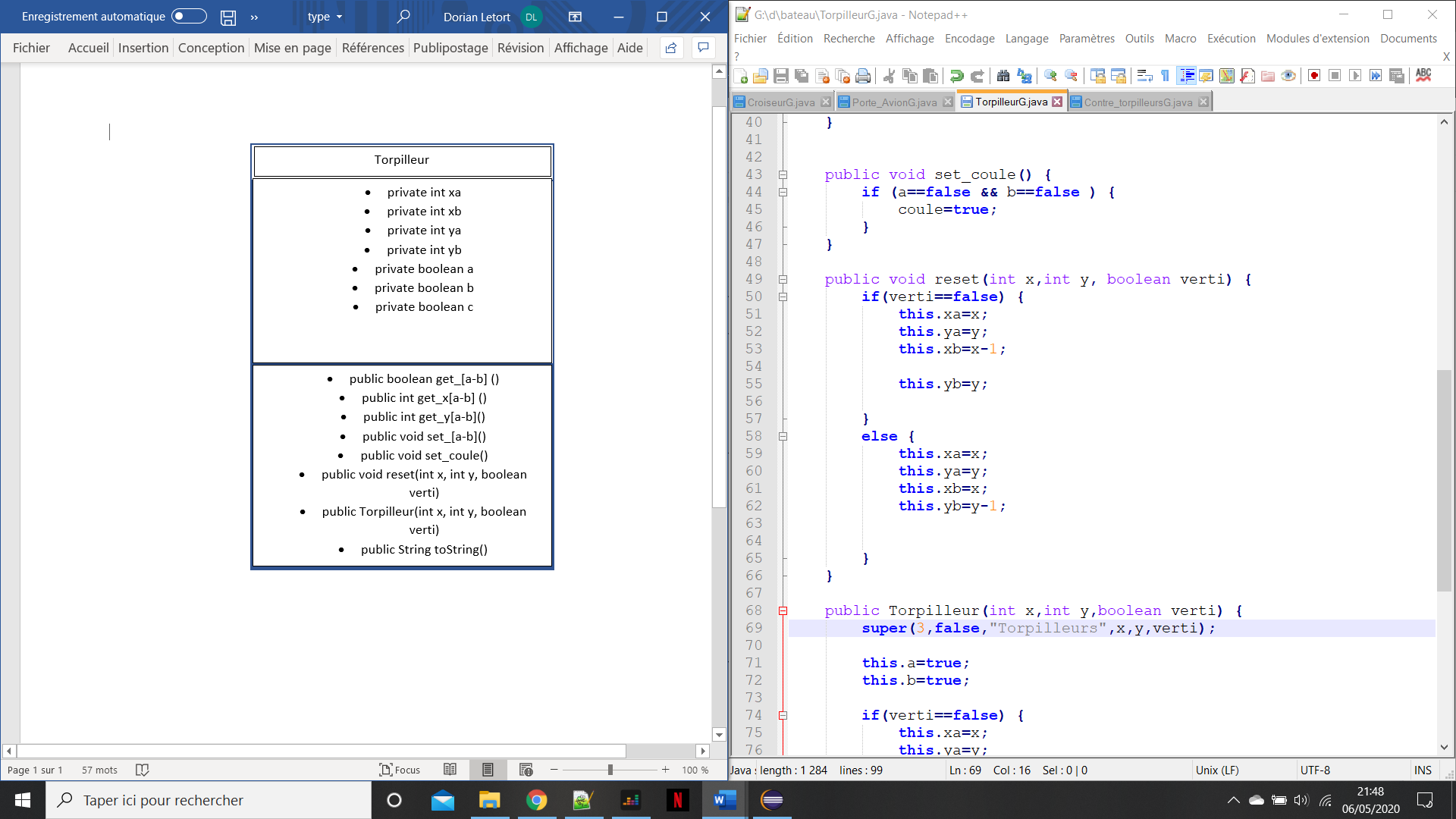
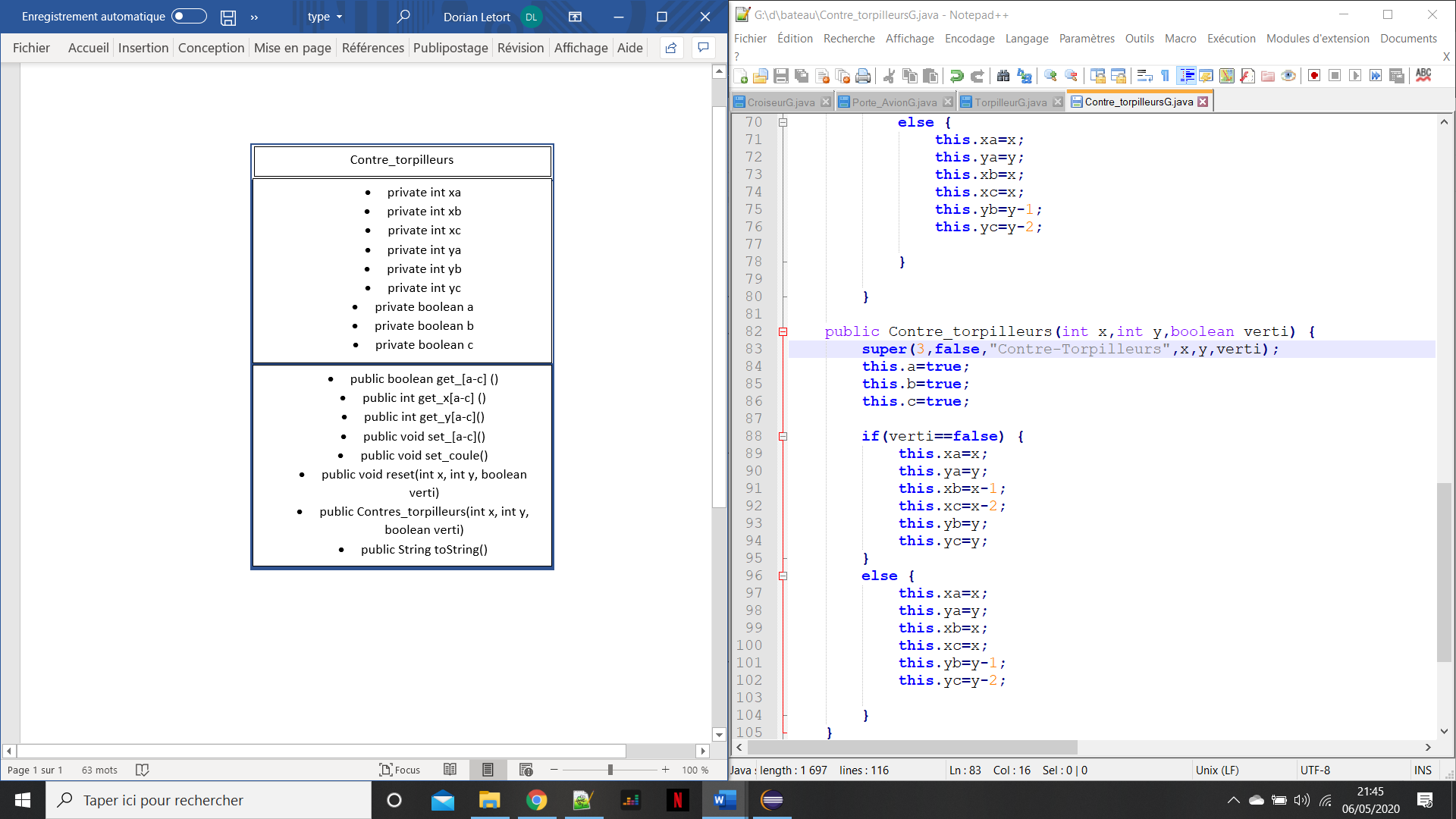
## **2.1-Les classes**

**Commençons par les classes de bateau pour cela j’ai créé une première classe Bateau qui est une classe dont héritera toutes les classes des différents bateaux .**



**Ensuite passons aux classes de chaque Bateau :**





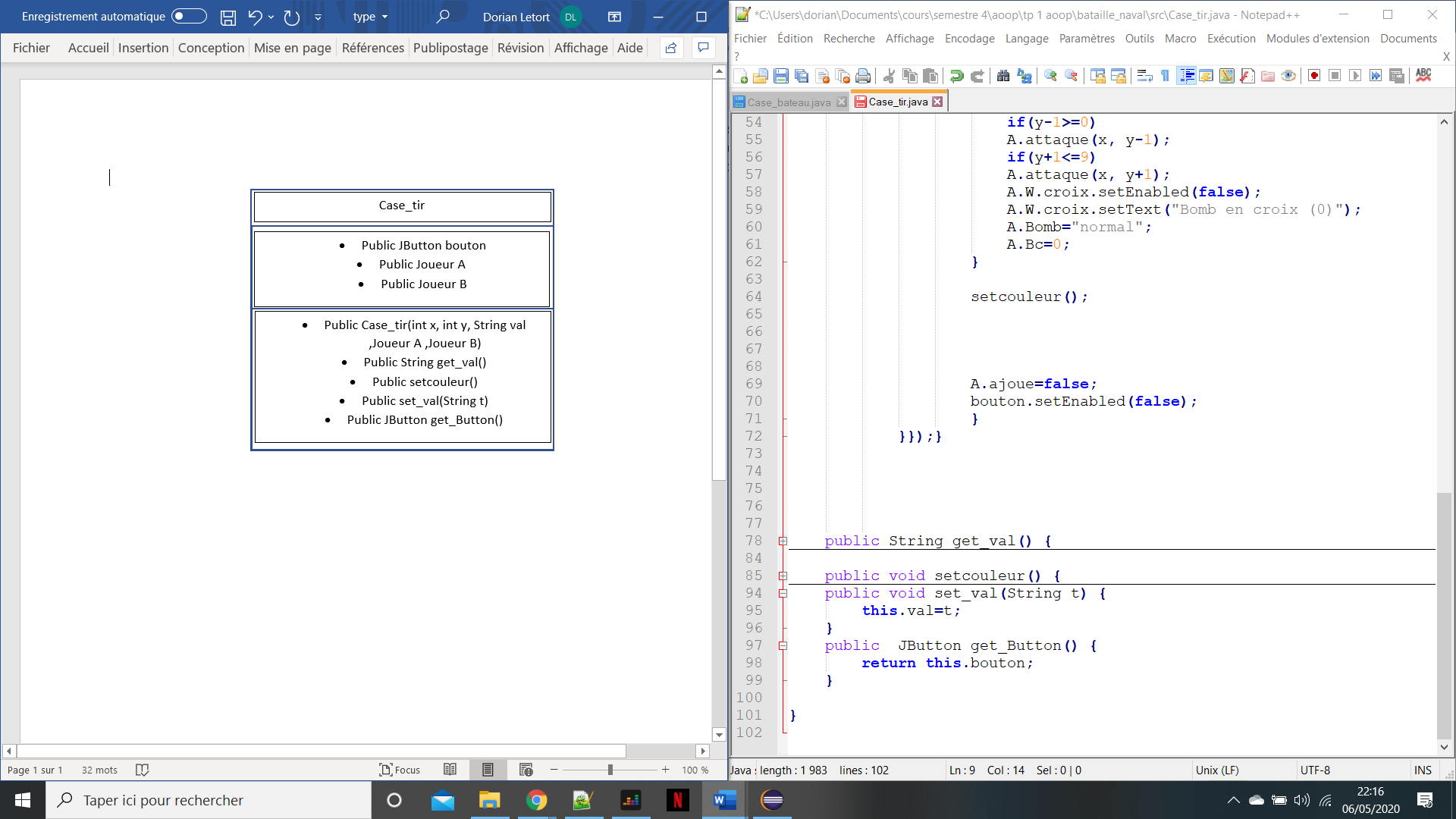
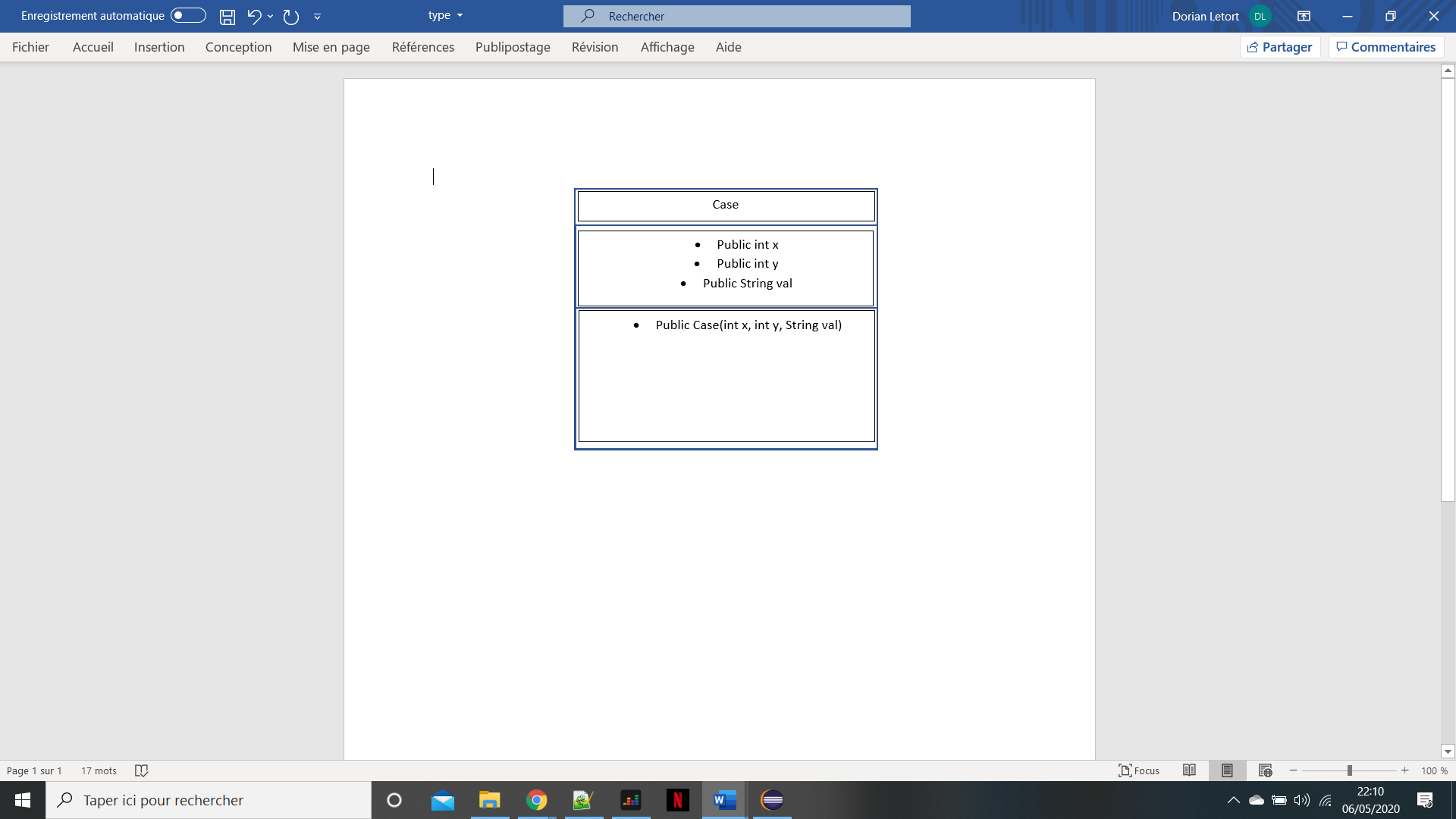
**\*pour raison de place et de répétition les méthodes avec [ a-e] correspond à la méthode répétée avec chaque caractère allant de a à e.**

**Pour chaque classe on remarque quelle possède un couple de int pour la position et un boolean pour chaque case du bateau cela dans le but de déterminer si ce bateau est coulé ou non .**

**Pour chaque class il y a une méthode reset cette méthode permet de set les positions des cases selon le x et le y de base et si le bateau est vertical ou horizontal.**

## **2.2-Les classes de jeu**

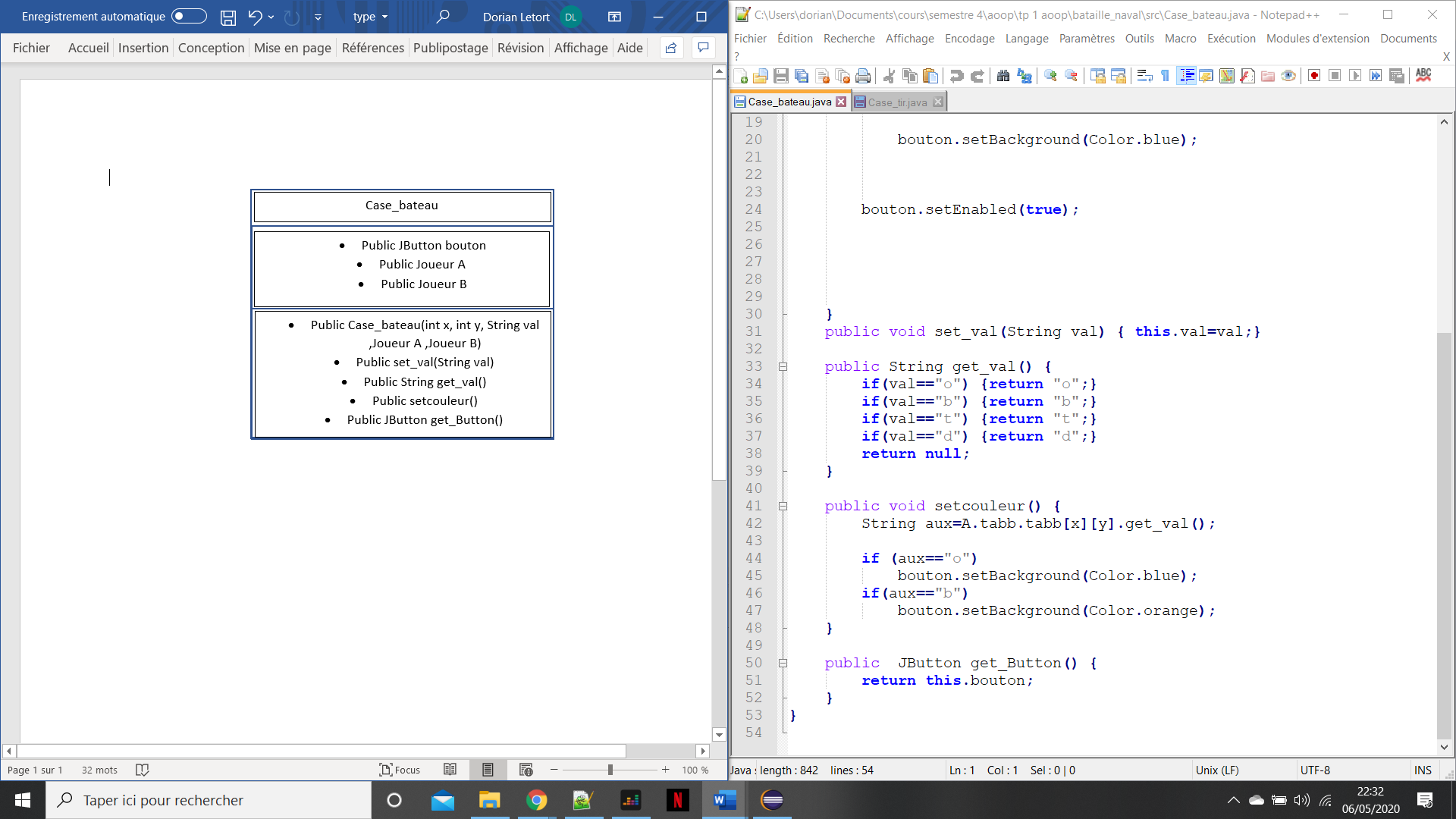
**Pour crée le jeu j’ai dû crée des classes pour les cases ,l’une pour celle qui tir et une autre pour celle qui contient nos bateaux, c’est pourquoi j’ai créé une classe Case\_tir et Case\_bateau qui herite toutes les deux de la classe Case :**



**Petite explication sur la classe Case\_tir :**

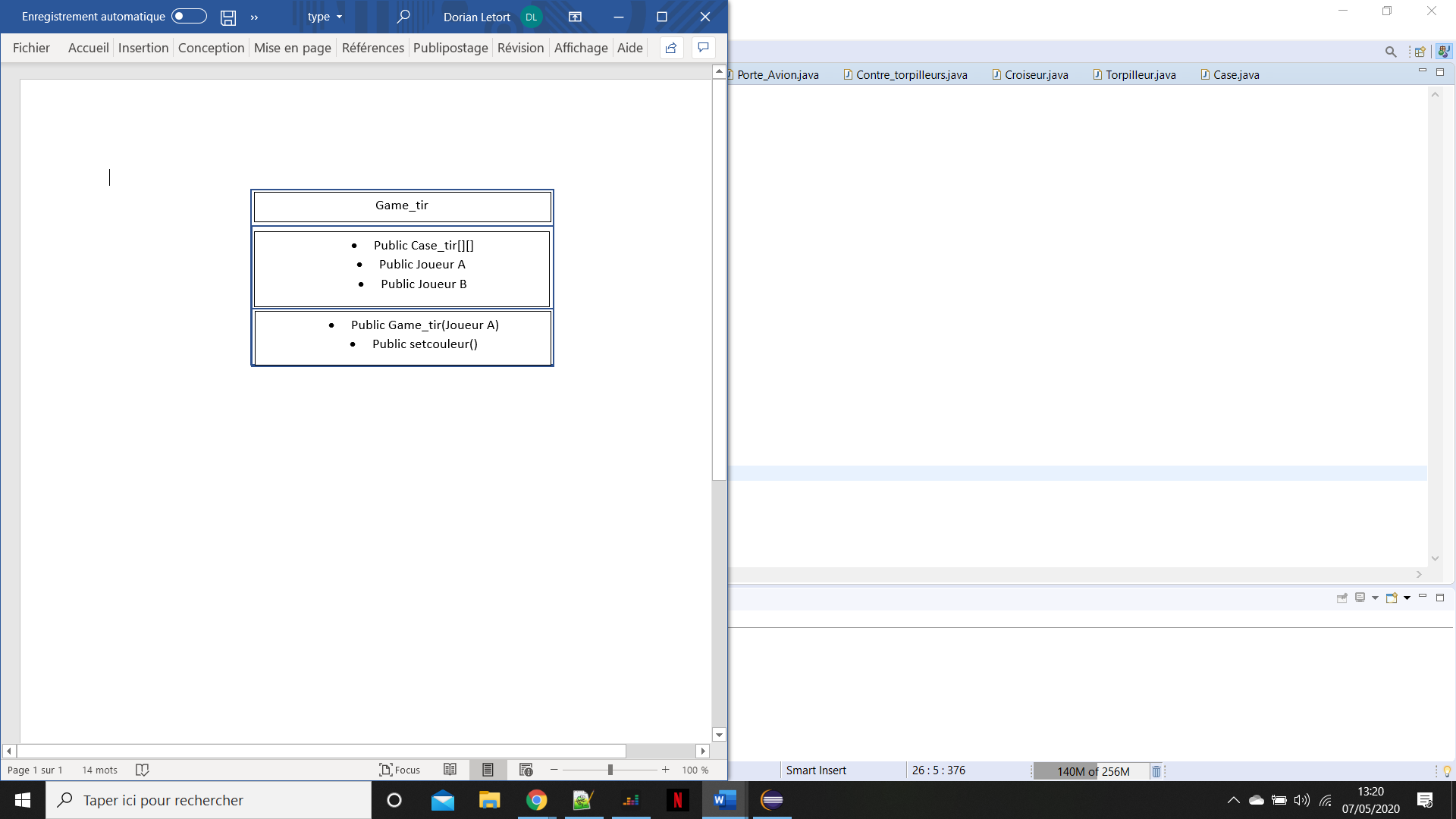
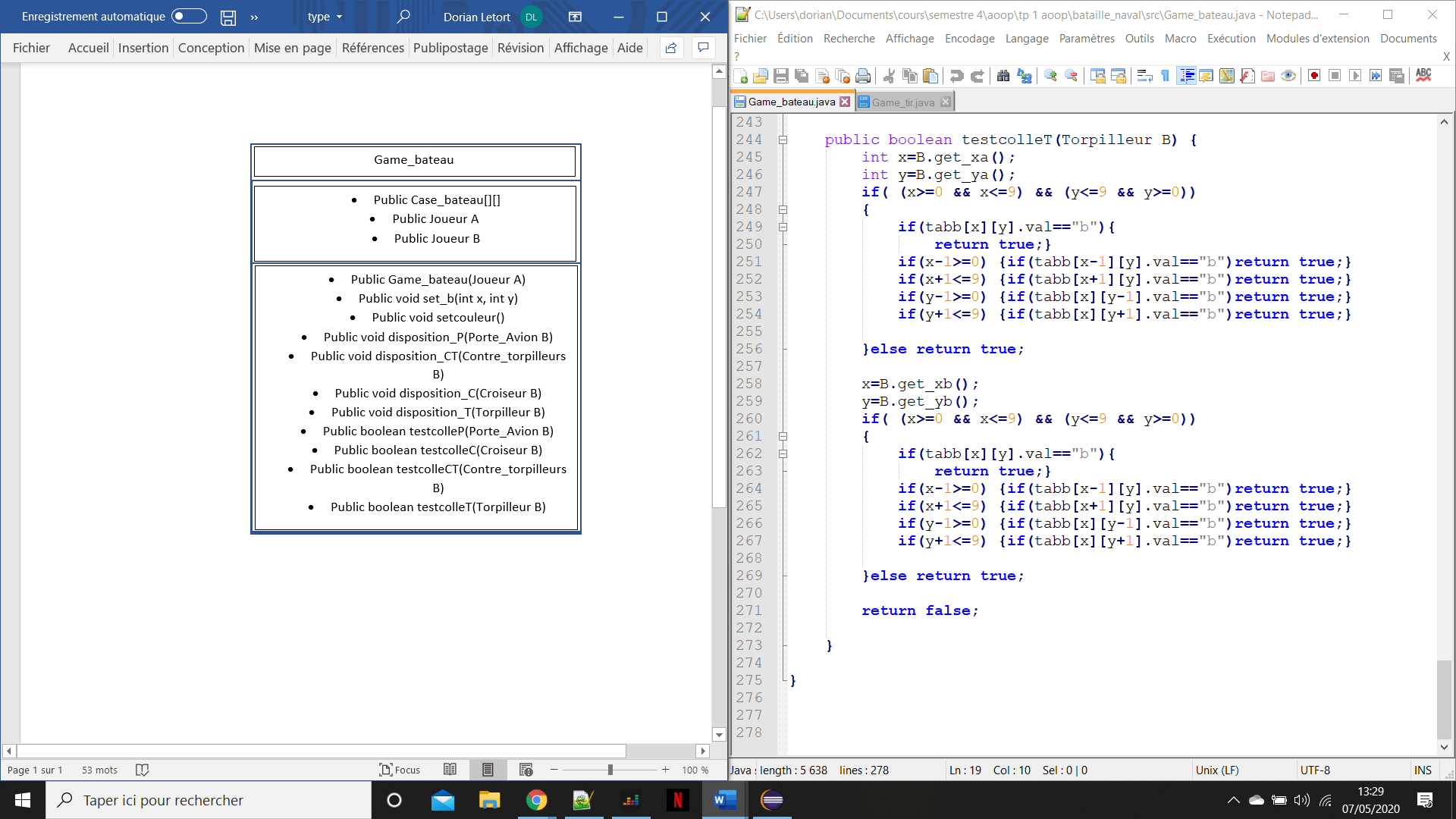
**Nous verrons ce qu’est l’objet Joueur .**

**La plupart des choses ce passes dans le constructeur où d’abord on effectue un super pour le constructeur de la classe Case , ensuite nous instancions le JButton et déclarons ça couleur en bleu , par la suite nous lui mettons un action listener qui fait une première vérification si c’est bien à lui de joué , ensuite nous cherchons quelle type de bombes doit être utilisé et selon la bombe utilise la méthode attaque de la classe Joueur , enfin il fait un appel a la méthode setcouleur qui selon le retour de attaque va changé la couleur de la case actuel puis va dire que le Joueur a joué et désactivé le bouton .**



**Pour la classe Case\_bateau c’est la même que la classe Case\_tir mais celle-ci dans sont constructeur désactive les boutons et met la couleur orange pour les bateaux avec la méthode setcouleur.**

**Pour incrémenter c’est Case j’ai les mis en tableau a deux dimensions dans la classe Game\_tir pour Case\_tir et dans la classe Game\_bateau pour Case\_bateau**

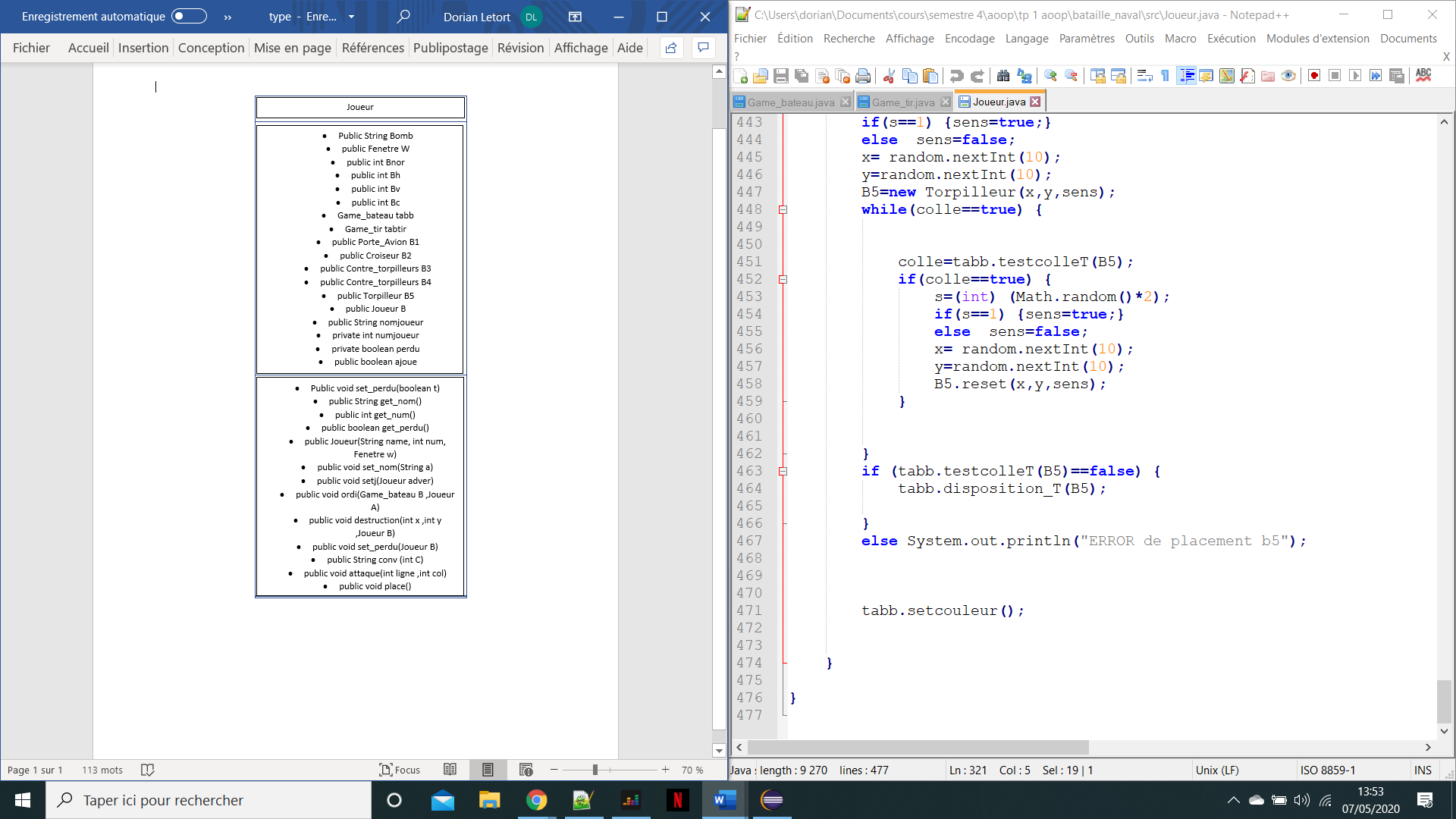
**Pour le Game\_tir il y a plusieurs choses à dire :**

**Premièrement il existe des methodes pour disposer chaque type de bateau dans le tableau de Case\_bateau .**

**Secondement il existe des méthodes pour savoir pour chaque bateau si l’un des bateaux est collé à un autre bateau .**

**Et enfin une méthode set\_b qui permet de mettre la valeur « b » dans la case de coordonné x et y .**

**Passons à la dernière classe de cette partie , la classe Joueur :**



**Pour cette classe nous avons plusieurs int avec un B comme premier caractère (Bh,Bv,Bc) cela correspond au nombre de bombe restante de chaque type . Ensuite nous avons chaque bateau qui sera placé dans le Game\_bateau par la fonction place qui utilisera les méthodes vues dans la classe Game\_bateau .**

**Puis nous avons trois méthodes différentes mais qui rejoigne toutes :**

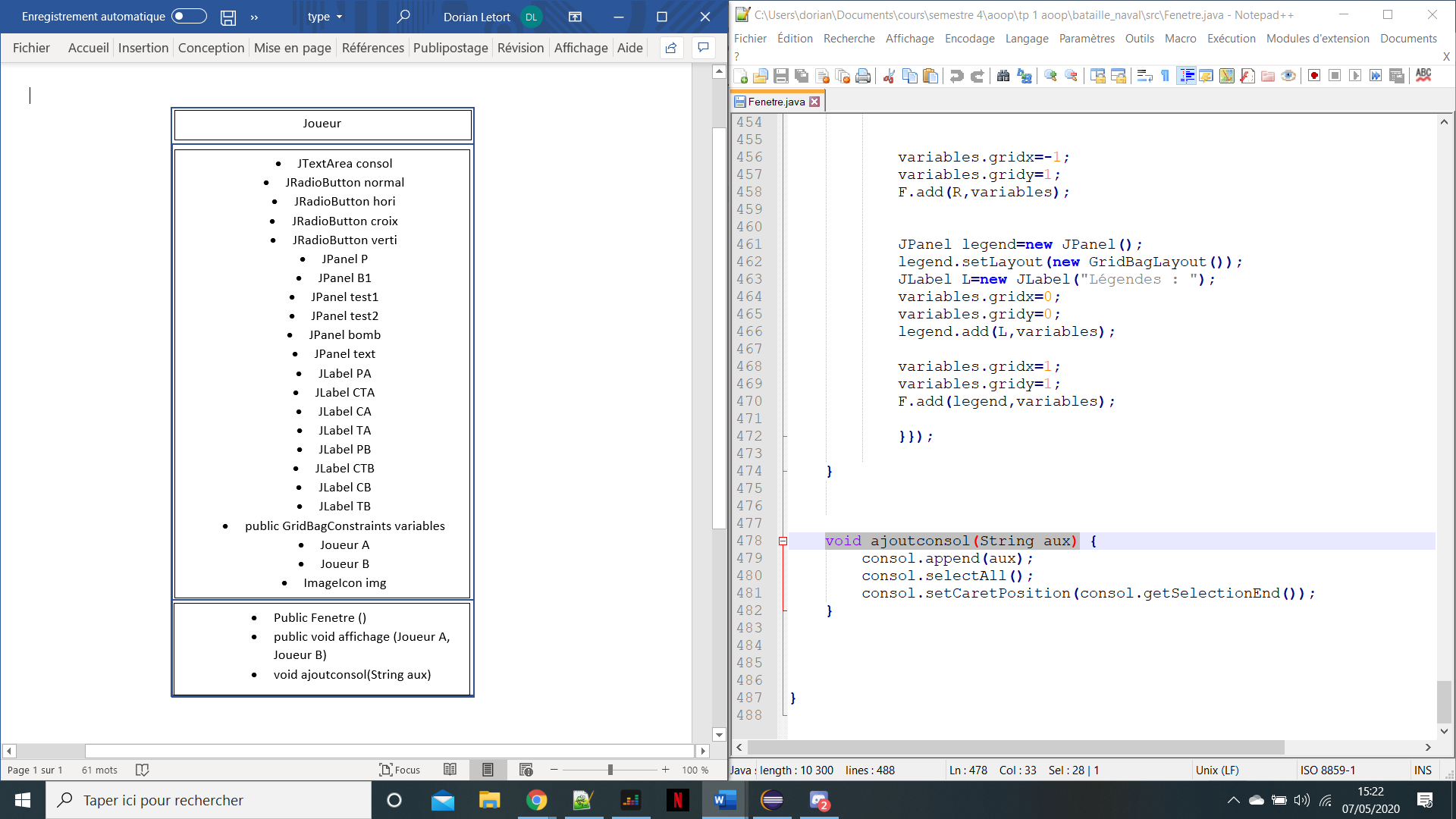
**-la méthode attaque qui est une méthode qui permet d’attaquer et selon la val du Game tir adverse modifie la Case\_bateau adverse et la Case\_tir de l’attaquant et si celle-ci attaque un bateau appelle la méthodes destruction . Cette méthode attaque est utiliser dans le action.Listener du bouton de Case\_tir .**

**-La méthode destruction , cette méthode détruit la partie du bateau attaqué .**

**-Et enfin la méthode ordi , qui est la méthode de l’intelligence artificielle qui permet d’avoir un adversaire , celui-ci va d’abord déterminer quelle bombe utilisé puis attaqué en appelant la méthode attaque et donc destruction .**

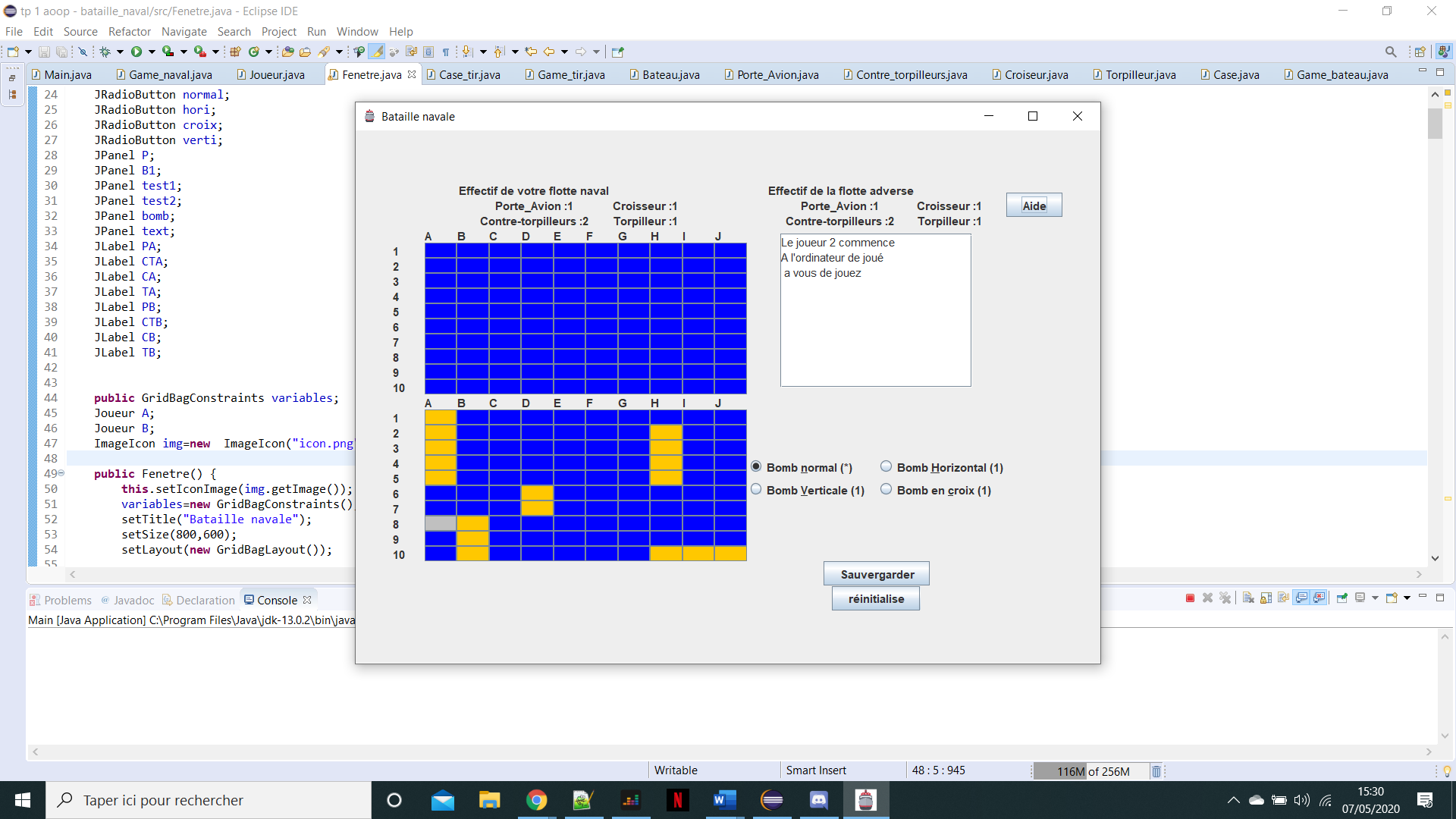
## **2.3-Les classes d’interfaces**

**Pour crée une interface interactive j’ai créé une classe qui est une extension de JFrame pour que celle-ci soit de ce type .**



**Quelque explication :**

**J’ai fait le choix de mettre la plupart des JLabel et JPanel pour pouvoir les modifier avec les autres classes et j’ai mis leurs constructeurs dans la méthode affichage, cette méthode permet de mettre en place tous les éléments pour obtenir ce rendu graphique :**



## **2.4-Les fonctionnalités**

**Cette interface est composée de deux tableau de JButton l’un correspondant au Game\_tir et l’autre au Game\_bateau**

**Pour les boutons de Game\_tir ceux-ci ne peuvent être activé seulement quand c’est au joueur de joué ,cela veut dire qui est impossible de jouer lorsque c’est au tour de l’ordinateur .**

**J’y ai ajouté au-dessus des JLabel indiquant l’état de la flotte adverse et de notre flotte**

**Ensuite il y a un petit bouton aide qui ouvre une JFrame contenant des aides pour jouer au jeu .**

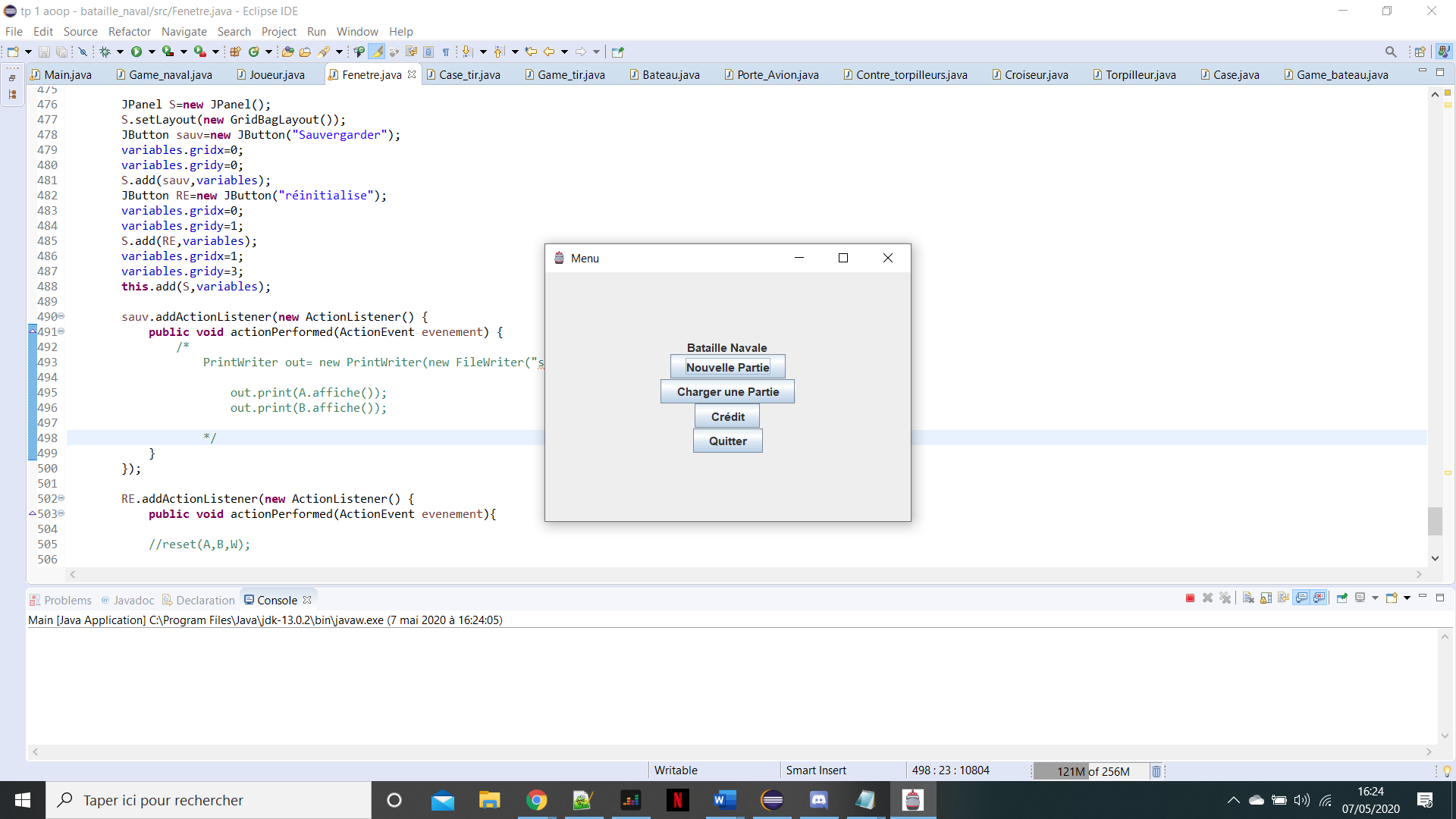
**Puis j’ai installé une console qui permet de voir les actions adverses et les notre , en ajoutant un scroll cela permet au joueur de regardé les actions passées et une actualisation a chaque coup effectué met la console en bas et donc affiche les dernières actions .**

**Des JRadioButton ont été ajoute pour pouvoir sélectionne le type de bombes a utilisé ceux -ci possède un JLabel dynamique qui indique le nombre de bombes restantes. Quand celle-ci passe à 0 le bouton est désactivé et passe automatiquement à la bombe normale ,il est donc impossible d’utilisé plusieurs fois les bombes spéciales.**

**Et enfin deux boutons, l’un pour sauvegardés qui enregistre les donné actuel du dans un fichier txt et l’autre qui réinitialise or par manque de temps et par différent problème dont j’ai repéré la cause (dû au contenu du JPanel text qui contient la console des actions dans la classe Fenetre celle-ci fait beugé toute la fenêtre et ne permet pas de recommencé une partie sans que je comprenne pourquoi ) .**

## **2.5-Les petits plus**

**Pour améliorer l’expérience de jeu j’ai créé dans mon Main une JFrame qui sert de menu d’accueil qui ressemble à cela :**



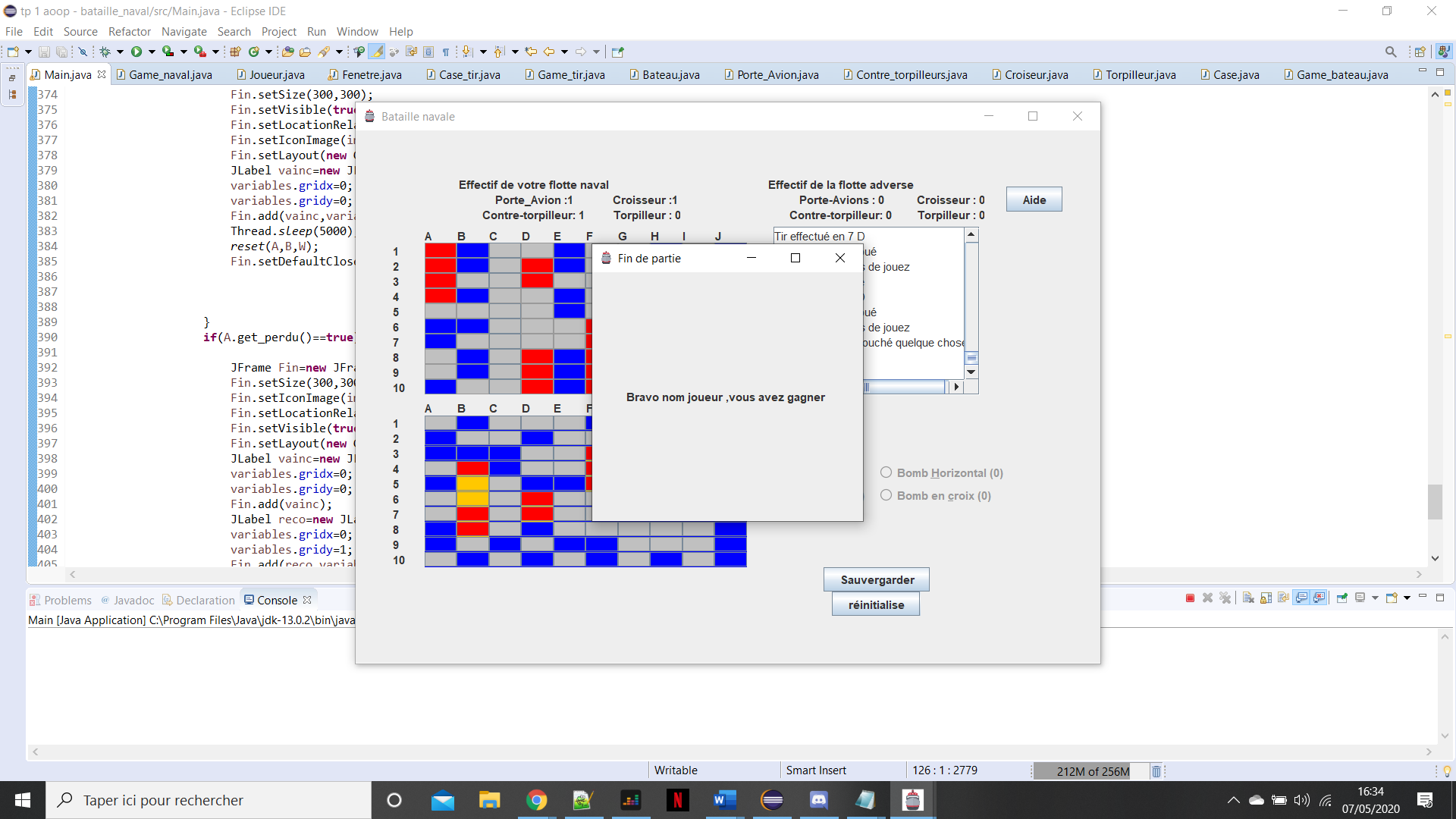
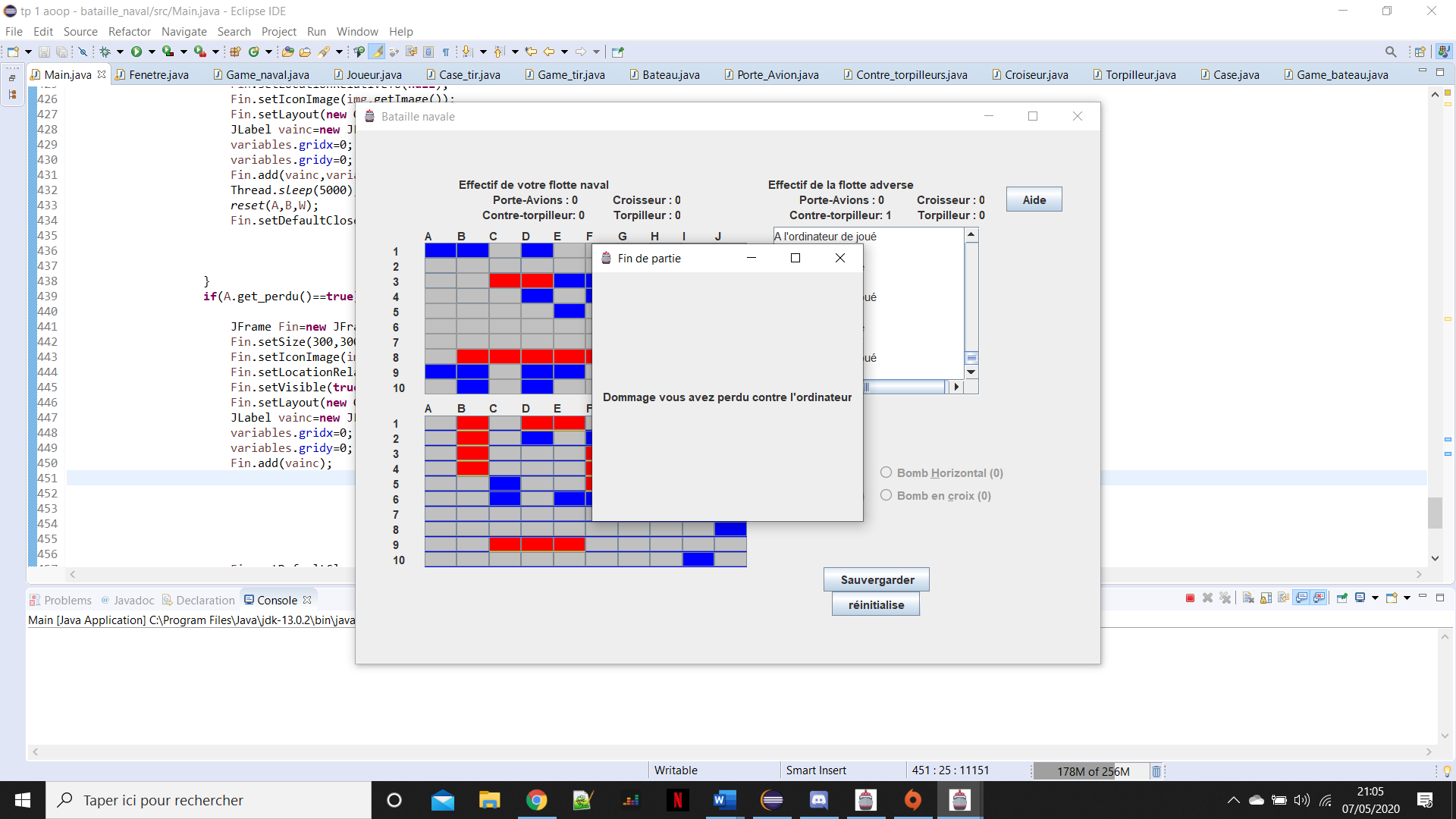
**Celui-ci permet de faire 4 choses :**

**-créé une nouvelle partie (cela ouvre une nouvelle JFrame), qui demande alors le nom du joueur et une petite option qui permet d’enlever les bombes spéciales et de joué seulement avec les bombes normales**

**-Chargé une partie sauvegardée (par manque de temps et empilement de différent problèmes dû a ça programmation cette fonctionnalité n’est pas active et est enlevé du code )**

**-Une petite page de Crédit pour voir qui a effectué de jeu et dans quelle circonstance**

**-un bouton pour quitter qui simplement ferme la fenêtre**

**Lorsqu’une des deux personnes à gagner l’une de c’est des deux fenêtres s’ouvre :**  

**J’ai inséré une petite Icon pour chaque fenêtre pour un meilleur visuel**

# **3-Conclusion**

**Pour conclure je pense que ce projet fut très enrichissant cela mon permis de voir comment se débrouiller seul ,organiser sont temps de travaille , avoir un enseignement autonome , savoir faire des choix de direction de projet .**

**Malgré le temps donné ,les échéances sont là et il faut rendre le projet mais je pense qu’il y a encore beaucoup a amélioré dans mon code même si celui-ci marche .**

**Voici quelques exemples de ce que j’aurai fait avec plus de temps pour ce projet :**

**-régler les problèmes autour de la sauvegarde et chargement de partie**

**-régler le problème autour du reset de partie , je laurais rajouter aussi en fin de partie avec des JButton en demandant si le joueur voudrait recommencer .**

**-j’aurai aimé rajouter se document dans les crédits , c’est-à-dire crée un bouton qui ouvrirai ce document avec le logiciel par défaut , principalement avec la fonction Desktop.getDesktop().open(File file).**

**-Amélioré l’IA en le scriptant aurai été un objectif et grâce a cela j’aurais pu mette différent niveau de difficultés .**

**-L’implémentation de Sprite dans les JButton de Case\_tir et Case\_bateau aurait donné un aspect plus esthétique agréable en plus de cela rajouté c’est Sprite dans les JLabel d’état de la flotte.**

**-Et enfin rajouté un système de score basé sur le nombre de bateau qui ont été détruit , le nombre de bombe utilisé et sur un temps .**