

Mini-projet Orienté Objet

Bataille navale

Objectif de ce mini-projet	2
Déroulement et règles du jeu	2
Le menu d'accueil.....	2
Caractéristiques du jeu.....	2
Caractéristique des navires	2
Les actions d'un navire par tour de jeu et par joueur	3
Visualisation des 2 grilles.....	4
Victoire, sauvegarde et chargement d'une partie.....	5
Planning et équipes	5
Les grandes étapes à respecter	5
1- Analyse et conception générale du diagramme de classes.	5
2- Analyse et conception détaillée	5
3- Développement dans un langage objet (codage, tests).....	5
Versionning de votre projet : GIT + Bitbucket.....	6
Consignes pour la soutenance	6

Objectif de ce mini-projet

L'objectif de votre projet est de réaliser un jeu de bataille navale dans un langage orienté objet, au choix en C++ ou Java, en mode console. **Vous devrez permettre à deux joueurs humains de s'affronter sur votre ordinateur.**

Déroulement et règles du jeu

Le menu d'accueil

Une fois lancé, le jeu proposera un menu classique permettant de réaliser les actions suivantes :

1. Jouer une partie
2. Charger une partie
3. Aide (doit expliquer clairement les règles du jeu et les touches du clavier à utiliser lors d'une partie)
4. Quitter

Caractéristiques du jeu

Le joueur et l'ordinateur disposent chacun de deux grilles de 15*15 cases :

- Une grille n°1 pour positionner et visualiser ses navires
- Une grille n°2 pour visualiser les dégâts causés à l'adversaire

Chaque joueur possède une flotte de 10 navires :

- 1 cuirassé
- 2 croiseurs
- 3 destroyers
- 4 sous-marins

Caractéristique des navires

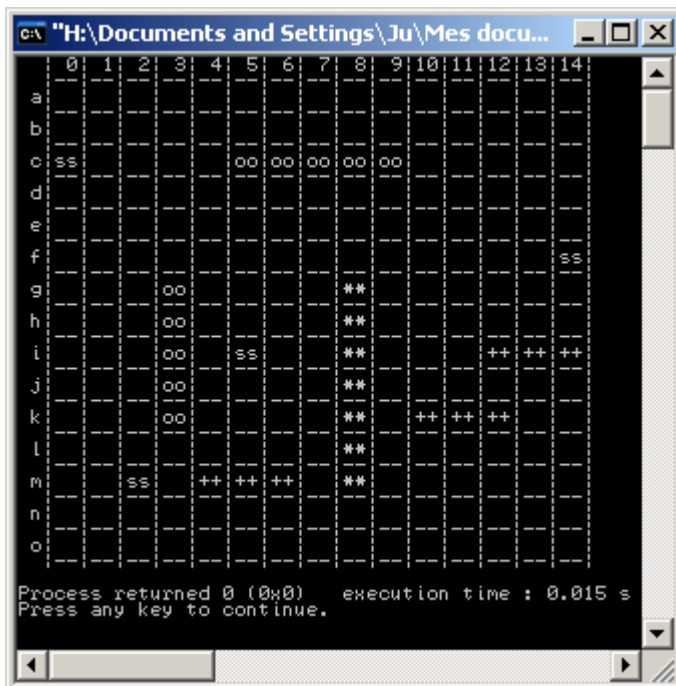
- Chaque type de navire est de taille différente :

type	taille
cuirassé	7 cases
croiseur	5 cases
destroyer	3 cases
sous-marin	1 case

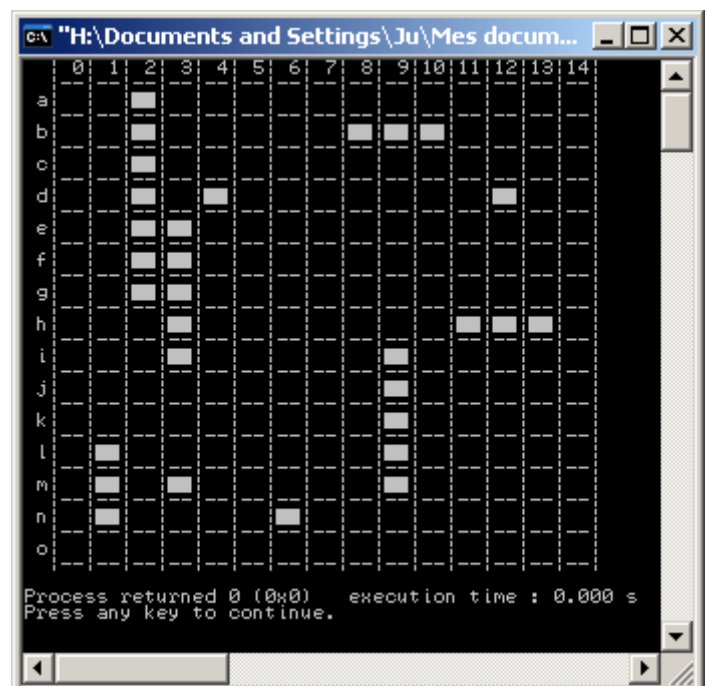
- Le positionnement des navires :

En début de jeu les navires sont positionnés aléatoirement et bien répartis dans la grille N° 1 de chaque joueur (voir ci-dessous des exemples). **C'est le joueur 1 qui commence la partie.** Les navires peuvent être positionnés verticalement ou horizontalement sur la grille. Mais attention, deux navires ne peuvent occuper la même case.

Positionnement des navires sur la grille n°1 du joueur humain



Positionnement des navires sur la grille n°1 de l'ordinateur



- La puissance de tir (nombre de cases touchées autour du point d'impact) dépend du type de navires :

type	Puissance de tir
cuirassé	9 cases
croiseur	4 cases
destroyer	1 case
sous-marin	1 case

Les actions d'un navire par tour de jeu et par joueur

Chaque joueur joue à tour de rôle. Bien entendu, un joueur ne doit pas voir les grilles de son adversaire : seules les grilles du joueur en cours sont affichées.

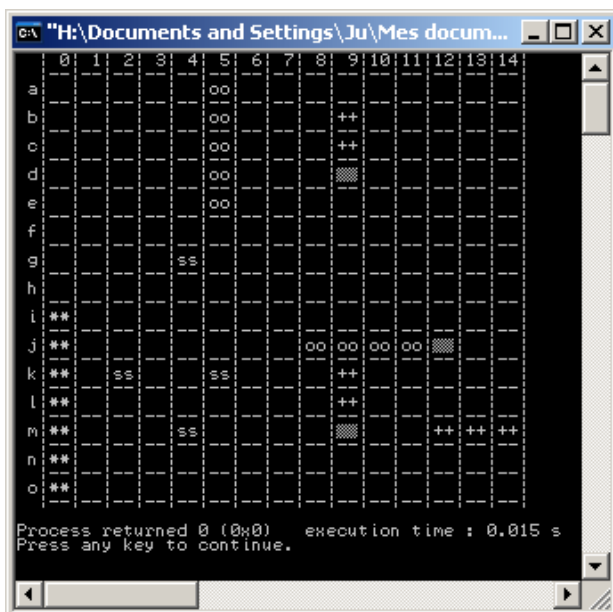
A chaque tour de jeu, les 2 joueurs peuvent choisir l'une des 3 actions suivantes sur un seul navire de leur choix de la grille n° 1, en choisissant les coordonnées de l'une des cases du navire :

- 1) Déplacer le navire d'une seule case. Un navire ne peut pas se déplacer en diagonale. Il ne peut se déplacer que dans sa direction horizontale ou verticale, et d'une seule case à la fois, et le point cardinal de déplacement (est ou ouest s'il est horizontal, nord ou sud s'il est vertical). Evidemment, en cas d'obstacle (case de déplacement occupée par un autre navire ou le bord de la grille), un navire ne pourra pas effectuer son déplacement et devra tenter une nouvelle action.
- 2) Tourner toutes les cases d'un navire d'un angle de 90 degrés (d'horizontal à vertical ou le contraire), vers l'un des 4 points cardinaux (sud ou nord s'il est horizontal, est ou ouest s'il vertical), sauf en cas d'obstacle dans la zone où le navire souhaite tourner. Dans ce cas d'échec, le navire devra tenter une nouvelle action. Les cuirassés sont plus difficiles à manœuvrer : tourner leur prend 2 tours.
- 3) Tirer. Le joueur tire en choisissant les coordonnées d'une case valide de l'adversaire (grille n° 2). Chaque destroyer n'est muni que d'une seule fusée éclairante. Le premier tir d'un destroyer dévoile un carré de 4*4 cases dans la grille adverse. Mais attention, les navires adverses de ce carré ne seront visibles que lors du tour du jeu (quelques secondes). Pour couler un navire, il faut avoir touché toutes les cases qu'il occupe : un cuirassé est plus résistant qu'un simple destroyer, les sous-marins ne peuvent couler que des sous-marins. Dans ce cas, le navire adverse coulé disparaît et la case touchée s'affiche sur la grille n° 2.

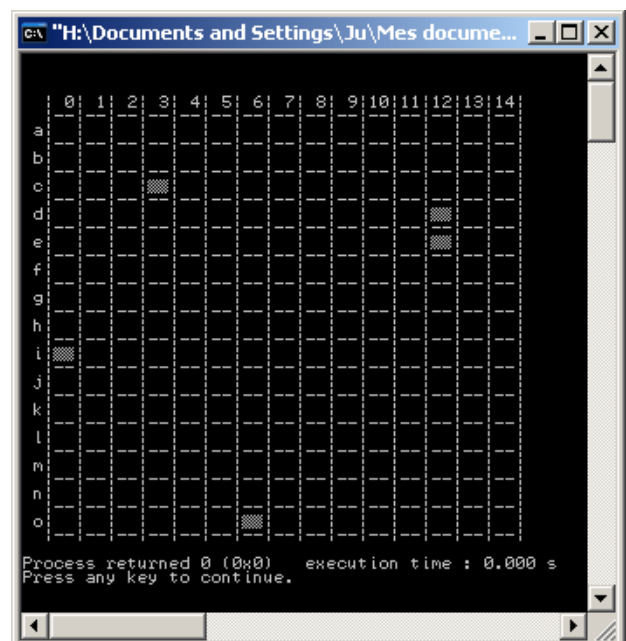
A chaque tour de jeu, il faudra visualiser les 2 grilles du joueur en cours avec les navires restant et coulés.

Visualisation des 2 grilles

Les captures d'écran ci-dessus sont données à titre d'exemples. A vous d'imaginer comment visualiser au mieux vos navires et les dégâts subits ou causés à l'adversaire.



Exemple (grille n°1) : En d9 et m9, 2 destroyers ont été touchés. En j12, un croiseur a été touché.



Exemple (grille n°2) : En i0, c3, o6, d12 et e12, des navires adverses ont été touchés.

Victoire, sauvegarde et chargement d'une partie

Le premier qui a coulé toute la flotte de son adversaire a gagné la partie !

A chaque instant, un joueur peut quitter la partie en cours avec une touche du clavier. La partie est alors sauvegardée dans **un fichier binaire (objets)**. **La sauvegarde comprend les objets de la partie : grilles des 2 joueurs et tous les objets contenus dans ces grilles (navires restant et coulés des grilles).**

Pour charger une partie, il faut passer par le menu principale et choisir "Charger une partie" qui charge tous les objets.

Planning et équipes

Période de réalisation du projet : 2 semaines du 13 mars et 20 mars 2017 et soutenances semaine du 27 mars 2017.

Inscrivez votre équipe au plus tard le lundi 13 mars 2017 en cliquant sur le formulaire suivant : [Projet OO | Équipes](#). Respectez bien les consignes qui sont spécifiées dans ce formulaire.

Les grandes étapes à respecter

1- Analyse et conception générale du diagramme de classes.

Votre **diagramme de classes** doit mettre en relation les classes en y intégrant si possibles de **l'héritage et du polymorphisme**. Pour chaque classe, définir les attributs (en général **private**, ou **protected** en cas d'héritage), et les méthodes (inutile d'y mentionner, les constructeurs, les getters et les setters).

2- Analyse et conception détaillée

Pour chaque classe, lister les prototypes de toutes les méthodes requises en précisant ses paramètres d'entrée et de sortie.

Réaliser progressivement une maquette du jeu en testant au fur et à mesure du codage et en tenant compte des différents scénarios.

Entre autres critères de qualité, le programme final devra être très facilement adaptable par tout autre développeur (exemples : changement des valeurs d'initialisation, changement des caractères et couleurs d'affichage, ...).

3- Développement dans un langage objet (codage, tests).

Dans le langage objet de votre choix (Java ou C++), implémenter le jeu en respect du CDC et de votre analyse des 2 étapes précédentes : diagramme de classes, organisation modulaire multi-fichiers selon l'approche **Modèle-Vue-Contrôleur**, si possible héritage et polymorphisme, commenter vos méthodes en précisant les paramètres d'entrée/sortie et commenter aussi l'ACD de ces méthodes et du programme principal.

Versionning de votre projet : GIT + Bitbucket

Cette année, vous pourrez utiliser l'outil de versionning qui vous sera enseigné durant l'intersemestre.

Vous utiliserez GIT avec Bitbucket pour gérer votre projet. Une formation vous sera dispensée pour apprendre à utiliser ces outils.

Lors de la soutenance, vous devrez montrer les différentes phases de développement de votre projet (les "révisions" du projet). Vous devez aussi être en mesure de montrer "qui a fait quoi" dans le projet.

Grâce au versionning, vous n'aurez plus de problèmes du type :

- c'est mon camarade qui a tout le projet, je n'ai pas pu corriger les erreurs...
- mon disque dur est mort la veille de la soutenance...
- on m'a volé mon ordinateur le jour de la soutenance...
- j'ai renversé du liquide sur mon clavier...
- je ne comprends pas, mon code a été écrasé ? et je n'ai pas fait de backups...
- ...

Et donc plus d'excuses le jour de la soutenance :)

Consignes pour la soutenance

Le lien vers le fichier d'explication qui suit décrit le principe, le déroulement, les attentes, les critères d'évaluation lors de la soutenance : [Projet OO - Déroulement soutenances](#). **Lisez attentivement ce fichier.**