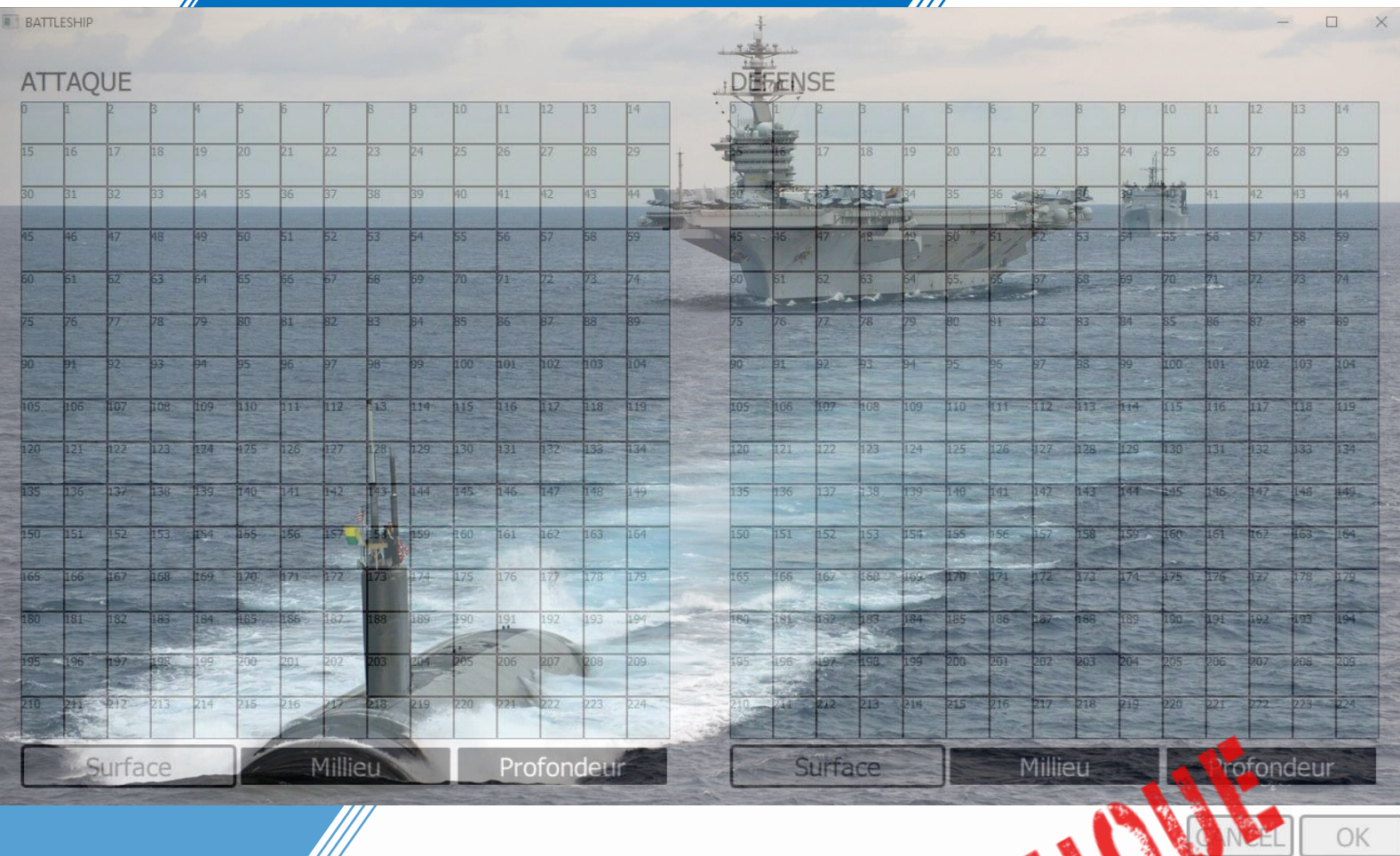


2020

Documentation : Partie Graphique

Projet POO Mega Bataille Navale





Sommaire

Sommaire	1
Documentation.....	2
Introduction.....	2
Difficultés.....	2
Notions apprises.....	3
Planification.....	3
Notice d'utilisation	4
Lancement.....	4
Choix du mode.....	5
Placement des bateaux	6
Jouer	10
Conclusion	11

Table des illustrations

Figure 1 - Chronologie réelle de la partie graphique	4
Figure 2 - Mode de connexion : Héberger.....	5
Figure 3 - Mode de connexion : En ligne.....	6
Figure 4 - Placement des bateaux	6
Figure 5 - Menu déroulant du choix Vertical/Horizontal	7
Figure 6 - Menu déroulant du choix des bateaux	7
Figure 7 - Représentation des trois profondeurs.....	7
Figure 8 - Flow chart du placement des bateaux.....	8
Figure 9 - Identification des cases	9
Figure 10 - Page de jeu avec attaque et défense	10



Documentation

Auteur : Massicot Jean-Gabriel

Depôt Git : <https://github.com/massicotjgab/mega-bataille-navale>

Introduction

La conception de la partie graphique de ce projet m'a beaucoup appris en langages orientés objets. En effet je n'avais que des bases peu solides dans ce type de langages, n'ayant jamais pratiqué réellement. Ce projet m'a donc permis de me familiariser avec les classes, les méthodes, les objets et les notions d'héritages.

J'ai décidé de me lancer un défi lors de ce projet qui a été de concevoir la partie graphique en QML à l'aide de Qt, lancé avec un script Python minimaliste. Le défi était que je ne connaissais pas le QML. Depuis ayant pris quelques cours de java, j'aurai sans doute décidé de choisir le Java pour programmer cette partie, mais j'ai tout de même réussi à faire une partie graphique plutôt réussie, bien que non-fonctionnelle.

Par ailleurs j'ai beaucoup aimé gérer le dépôt GIT. Cela m'a beaucoup appris sur les techniques qui permettent d'avoir un dépôt propre.

Difficultés

Une des difficultés que j'ai eu est que j'ai passé beaucoup de temps à apprendre et tester des choses de mon côté. Quand j'ai été prêt pour commencer à intégrer, mes camarades avaient bien avancés. Nous avons donc commencé à intégrer vers la fin du projet, et des problèmes se sont posés sur la méthode de communication entre les parties. Le fonctionnement graphique m'a obligé à faire changer des éléments dans la partie GameEngine. L'idéal aurait été d'intégrer au fur et à mesure, mais cela s'avère compliqué quand on apprend le langage qu'on utilise au fur et à mesure. L'impact a été important au niveau du travail du groupe.

Le travail de mon collègue Thomas Coutant sur la partie réseau a pris également du temps, et l'intégration Réseau-GameEngine a eu lieu également vers la fin du projet. Finalement je n'ai même pas pu intégrer la gestion réseau alors que cette intégration paraît assez légère à posteriori.

L'apprentissage de la gestion du dépôt GIT a pris également du temps sur les heures de projet prévues initialement. J'ai beaucoup aimé gérer le dépôt, cela fait partie intégrante du projet et me paraît indispensable, mais le temps de projet en lui-même en a été réduit.

Par ailleurs j'ai eu beaucoup de mal dès le début à avoir un environnement qui fonctionne. Cela a été très frustrant et j'ai perdu beaucoup de temps pour commencer à avoir un premier résultat.



En termes de planification par rapport à ma partie cela s'est traduit par un flou artistique total, et une course contre la montre pour essayer de rattraper ce temps perdu.

Nous n'avons pas tenu les délais des semaines de travail prévues, et par la suite il a été compliqué de s'organiser pour trouver des moments de travaux communs. Enfin est arrivé le confinement qui nous a encore compliqué les choses.

Pour résumer les problèmes rencontrés sont :

- La découverte d'un nouveau langage associé à une mauvaise connaissance de la POO
- La mise en place de l'environnement
- L'organisation
- La communication
- La planification
- Le fait de trop considérer les parties comme indépendantes
- L'apprentissage parallèle de l'administration d'un dépôt GIT

Notions apprises

Les notions que j'ai apprises sont multiples. Je les ai déjà citées pour la plupart, mais je vais les résumer ici.

Tout d'abord, j'ai appris les notions d'orienté-objet. J'avais des difficultés sur ces langages que je n'avais jamais réellement utilisé, et j'apprends mieux par la pratique. Même si nous avions déjà eu des cours sur le sujet, je n'avais pas bien acquis ces notions. Maintenant je comprends bien mieux le système de classes, les méthodes, les objets, et la notion d'héritage, et j'ai appris à les utiliser.

J'ai appris le QML que je ne connaissais pas du tout, et cela m'a permis de me conforter dans l'idée que j'aime bien les langages haut niveau et les langages descriptifs, notamment parce que le rendu visuel/fonctionnel est souvent immédiat et facilement visible. Cela correspond bien à mon caractère créatif et à mon mécanisme de pensée basé sur le visuel.

J'ai appris que l'organisation d'un projet de ce genre est assez complexe, qu'il faut vraiment bien se connaître, bien connaître son groupe et avoir une vision commune. La découverte et l'apprentissage doivent être planifiés. L'intégration doit se faire au fur et à mesure, se discute et se réfléchit aussi régulièrement. Même si les parties semblent bien délimitées en termes de fonctionnalités, elles n'en restent pas moins un projet unique.

J'ai appris à gérer un dépôt git et son historique, à manager les pull-request et les merges, et à maintenir un dépôt git propre. Je suis assez fier du nôtre, même s'il n'est pas parfait suite à quelques fausses manipulations. Un « `git log --all --oneline --graph --decorate` » pourra le prouver.

Planification

La planification de ma partie n'existe pas. Comme expliqué précédemment, quand j'ai commencé le projet, j'ai perdu l'équivalent de 4 jours de travail pour la mise en place de

l'environnement et pour commencer à avoir un premier résultat très basique visible. Cela a été très frustrant.

Par la suite j'ai tâtonné pour apprendre le langage et j'ai perdu à nouveau du temps de développement. Dès que je voulais faire quelque chose de plus il fallait que j'apprenne des notions supplémentaires et des éléments de langage. Mon efficacité n'a été que de 5 jours environ, c'est-à-dire la moitié du temps imparti à l'origine, ce qui explique en partie l'état d'avancement du projet et ma désynchronisation avec mes camarades.

Finalement mes tâches peuvent se représenter grossièrement de cette façon :

CHRONOLOGIE PARTIE GRAPHIQUE					
Taches	Semaine 1	Semaine 2	Semaine 3	Semaine 4	Semaine 5
Mise en place de l'environnement					
Création premier quadrillage					
Gestion des multi-couches					
Gestion ATTAQUE/DEFENSE					
Fenêtre de placement des bateaux					
Première fenêtre gestion de connexion					
Début de positionnement statique des bateaux					
Début d'intégration et de positionnement des bateaux à l'aide du GameEngine					

Figure 1 - Chronologie réelle de la partie graphique

Notice d'utilisation

Lancement

En termes d'utilisation, je vais présenter chaque fenêtre, et expliquer comment elles sont censés être liées au reste du projet.

Tout d'abord, sous Windows, il faut se placer dans le répertoire du projet **mega-bataille-navale/** cliquer dans la barre où se trouve le chemin, taper « powershell » et valider avec la touche

« Entrée ». La console de commande Powershell s'ouvre, et il ne faut ensuite pas oublier de préparer l'environnement avec les commandes suivantes :

```
PS yourpath/mega-bataille-navale> & d:/GIT/mega-bataille-navale/venv/Scripts/activate.ps1
(venv) PS yourpath/mega-bataille-navale> $Env:PYTHONPATH="."
(venv) PS yourpath/mega-bataille-navale> python .\Src\Graphic\batleship.py
```

La page d'accueil suivante s'ouvre alors :

The screenshot shows a window titled 'BATTLESHIP' with the subtitle 'CHOIX DU MODE DE CONNEXION'. It contains two buttons: 'Héberger' (highlighted with a grey border) and 'En ligne' (black background). Below these buttons are two input fields: one labeled 'Nom' and another containing the IP address '255.255.255.255'. At the bottom right, there are two buttons: 'CANCEL' and 'OK'.

Figure 2 - Mode de connexion : Héberger

Choix du mode

Nous sommes sur la page de sélection du mode de jeu. Par défaut, nous sommes sur le mode hébergeur, c'est-à-dire que c'est nous qui accueillons la partie. Le champ « Nom » est utilisé pour le pseudo du joueur, tandis que l'adresse IP affichée doit être celle de l'ordinateur du joueur hébergeur.

Dans le cas où nous sommes hébergeur, c'est la partie Réseau qui récupère l'adresse IP active et l'envoie pour qu'elle soit affichée.

Dans le cas du mode « En ligne » (voir ci-dessous) les deux champs sont modifiables, et le champ « Adresse ip » doit être remplis avec l'adresse IP de l'hébergeur (l'adversaire donc).

Une fois le mode choisis et les champs remplis on appuie sur « OK ». A noter qu'actuellement le bouton « OK » est actif en permanence et que les champs ne sont pas vérifiés. Les boutons « OK » et « CANCEL » servant à naviguer entre les pages à des fins de débogage visuel. « OK » sert donc à avancer d'une page tandis que « CANCEL » sert à reculer d'une page.

CHOIX DU MODE DE CONNEXION

Héberger En ligne

Nom

Adresse ip

CANCEL OK

Figure 3 - Mode de connexion : En ligne

Placement des bateaux

Après la validation, la fenêtre suivante s'ouvre. Il s'agit de la page de placement des bateaux.

DEFENSE

Porte-Container 5x2

VERTICAL

Surface Milieu Profondeur

CANCEL OK

Figure 4 - Placement des bateaux

Il existe tout d'abord deux menus déroulants. Le premier sert à sélectionner le modèle de bateau souhaité, et le second à choisir sa position verticale ou horizontale.

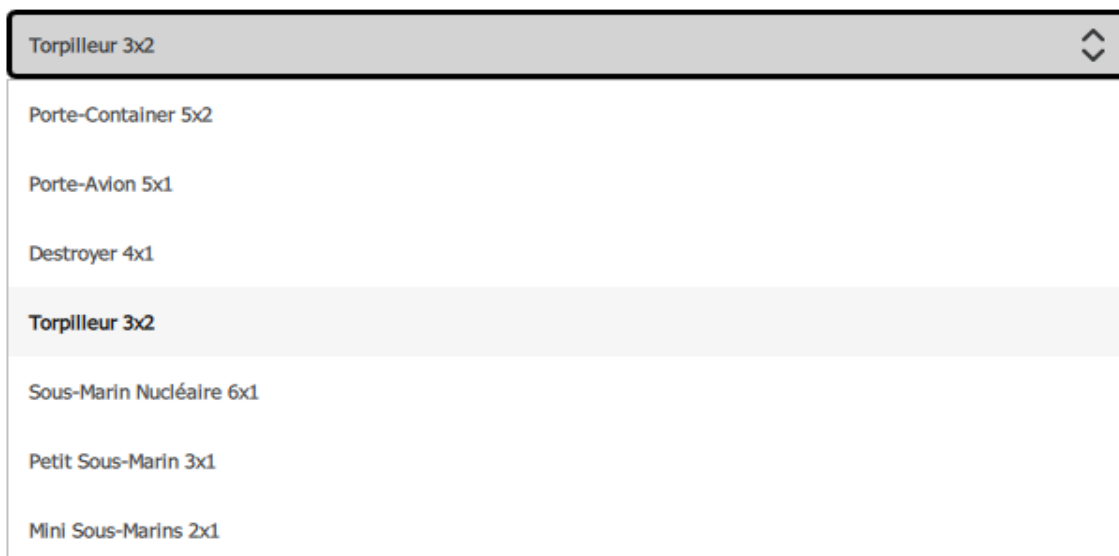


Figure 6 - Menu déroulant du choix des bateaux

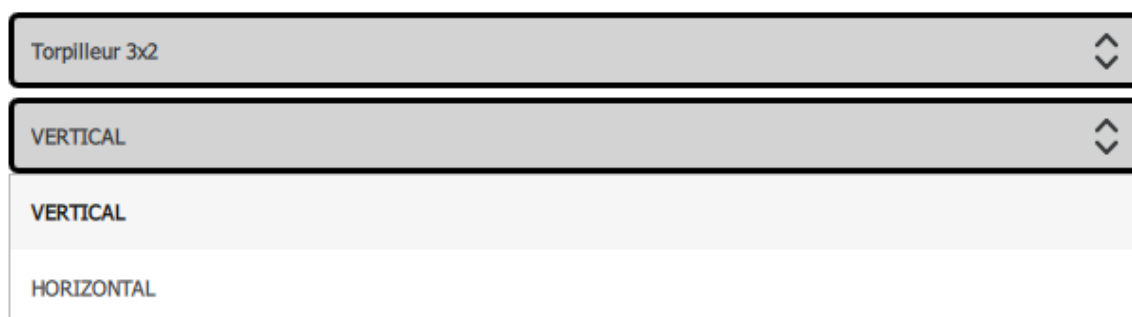


Figure 5 - Menu déroulant du choix Vertical/Horizontal

Le plateau est une grille de 15x15 sur trois couches. On peut naviguer entre les couches à l'aide des boutons « Surface », « Milieu » et « Profondeur ».

Pour placer un Navire, il suffit de le sélectionner dans la liste, de choisir sa position, et de choisir une case parmi les trois couches de la grille. Cette case est sensé représenter le coin supérieur gauche du bateau. Le placement n'est pas implémenté, mais lorsque la personne clique il était prévu qu'il se passe la chose suivante :

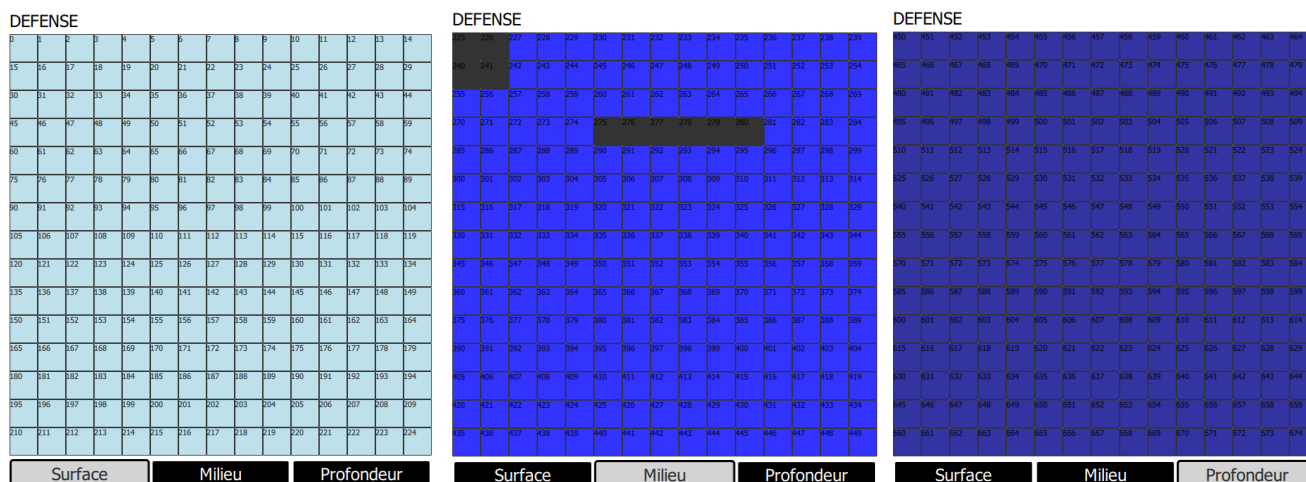


Figure 7 - Représentation des trois profondeurs

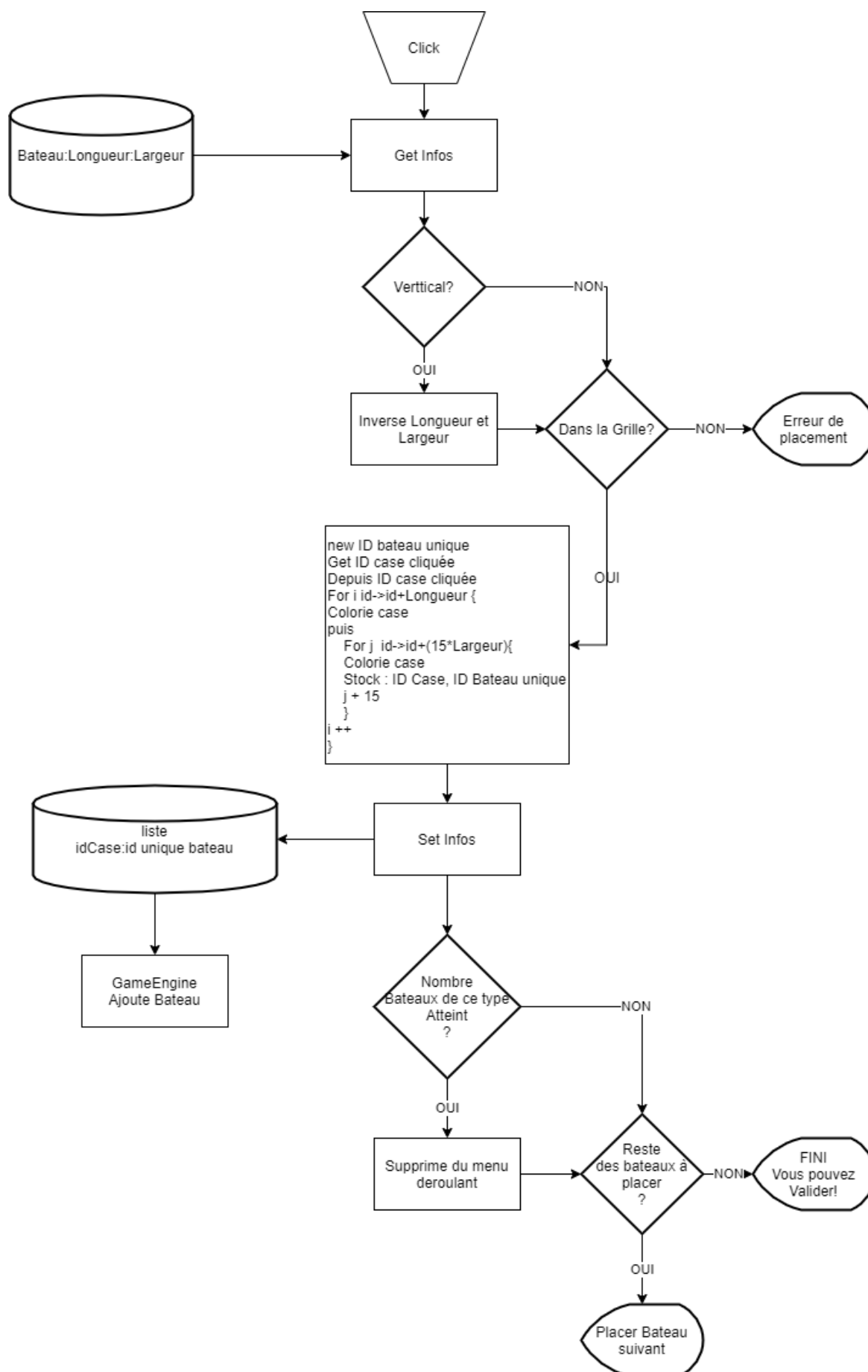


Figure 8 - Flow chart du placement des bateaux

Le placement des bateaux utilisera une base de données, soit un fichier de configuration dans lequel sera répertorié chaque bateau ainsi que ses dimensions à l'horizontale : la longueur pour l'axe des x et la largeur sur l'axe des y. La sélection du mode vertical inversera les valeurs avant le placement. Le placement utilisera ensuite le GameEngine pour créer la grille avec les bateaux disposés.

On peut remarquer sur chacune des grilles que les cases sont numérotées, il s'agit de l'ID de la case, présent pour débogage visuel, mais destiné à être caché par la suite. La première grille possède les ID de 0 à 224, la seconde, celle de la couche « Milieu » possède les ID 225 à 449 et la dernière couche de 450 à 674.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
75	76	77	78	79	80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99	100	101	102	103	104
105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134
135	136	137	138	139	140	141	142	143	144	145	146	147	148	149
150	151	152	153	154	155	156	157	158	159	160	161	162	163	164
165	166	167	168	169	170	171	172	173	174	175	176	177	178	179
180	181	182	183	184	185	186	187	188	189	190	191	192	193	194
195	196	197	198	199	200	201	202	203	204	205	206	207	208	209
210	211	212	213	214	215	216	217	218	219	220	221	222	223	224

Figure 9 - Identification des cases

On peut aussi remarquer des rectangles foncés sur la couche du milieu et qui sont un début de placement de bateaux à l'aide du GameEngine. Les méthodes créées par Samuel m'ont donc permis de placer deux bateaux, mais ceux-ci ne dépendent pas de la méthode de placement des bateaux non encore finalisée.

Une fois tous les bateaux disposés, le bouton « OK » devient cliquable et on peut commencer à jouer.

Jouer

On obtient alors la page suivante :

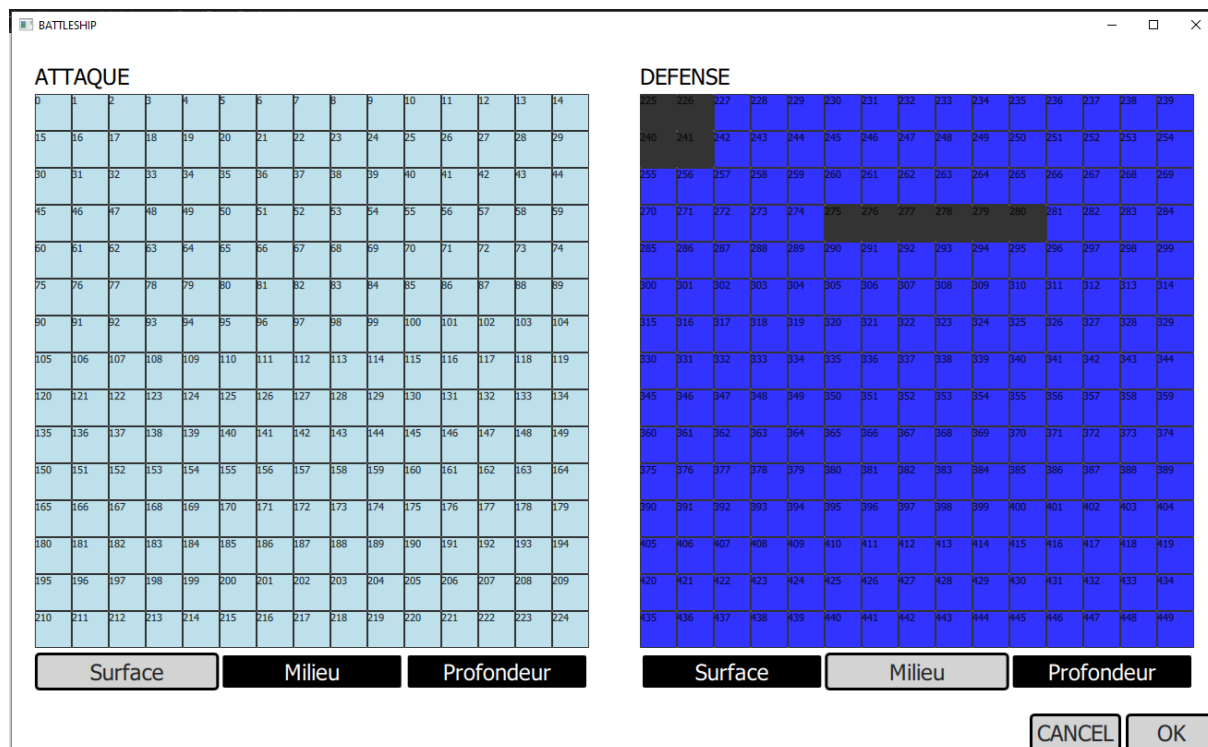


Figure 10 - Page de jeu avec attaque et défense

A gauche la grille d'attaque. Quand vient notre tour elle devient cliquable. Il suffit alors de cliquer sur la case voulue de la grille de surface. Un message préviendra alors le joueur du résultat de son tir et de quelle case de quelle couche est touchée. Les cases présenteront alors un signe distinctif.

Il était prévu une croix dans les cases « à l'eau » (pour chaque couche traversée, donc un tir sans touche positionne 3 croix : une à chaque couche) et une coloration rouge pour les cases touchées (et le tir s'arrête au bateau touché). La navigation entre les couches servira à contrôler l'état des lignes adversaires. Un bateau coulé sera coloré en noir.

Les tirs sont envoyés au GameEngine et au réseau. Le GameEngine renvoie sur chaque action la liste mise-à-jour de l'ensemble des cases.

Le fonctionnement de la grille de défense est sensiblement la même mais est strictement passive (non cliquable) et il est prévu que les bateaux non coulés soient vus plus ou moins en transparence à travers les couches de façon à donner une idée au premier coup d'œil au joueur. A l'origine les bateaux sont gris clair, touchés ils sont rouges, coulés, ils sont noirs.

Une possibilité aurait été d'avoir un champ en dessous des grilles montrant les bateaux coulés alignés pour chaque camp.



Conclusion

La partie graphique dans sa partie affichage a pu être terminée. Cependant dans la partie fonctionnelle de nombreux éléments restent à implémenter. Le temps perdu par manque de compétence, de connaissance et d'organisation, s'il avait été mis à profit pour développer le projet, aurai pu permettre sa finalisation. La partie placement n'est pas finalisée, la partie jeu n'est pas implémentée, et l'intégration en est à peine à son ébauche.

J'estime à environ 5 jours de travail supplémentaire le temps qu'il nous aurai fallu pour finir ce projet.

Cela dit, le principe d'un projet d'étude est surtout d'apprendre et de ce côté-là, ce projet a été très intéressant et très efficace. J'ai beaucoup appris, et cela faisait longtemps que je n'étais pas venu en cours avec plaisir. Or, en dehors des premiers jours assez frustrant, ce fut le cas tout au long du projet.

Je remercie donc M. Vincent Poulailleau pour ce projet, son suivi et ses explications, notamment sur GIT. Je remercie également mes camarades Samuel Huet, Thomas Coutant et Vincent Villedieux. Même si ca n'a pas toujours été facile de se coordonner, il est plutôt facile de travailler ensemble, et nous avons passé de bons moments.