

2020

Jean-Gabriel Massicot

**Une image contenant trafic, lumière, arrêt, assis

Description générée automatiquement![Une image contenant horloge

Description générée automatiquement]()**Une image contenant eau, extérieur, navire, embarcation

Description générée automatiquement

Documentation : Partie Graphique

Projet POO Mega Bataille Navale

# Sommaire

[Sommaire 1](#_Toc36921191)

[Documentation 2](#_Toc36921192)

[Introduction 2](#_Toc36921193)

[Difficultés 2](#_Toc36921194)

[Ce que j’ai appris 3](#_Toc36921195)

[Planification 3](#_Toc36921196)

[Utilisation 4](#_Toc36921197)

# 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 aurai 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 parait 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 fonctionner. 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 résumé 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

## Ce que j’ai appris

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 a 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 connaitre, bien connaitre 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 |  |  |  |  |  |

## Utilisation

## 