RAPPORT DE PROJET MON2048

Elèves : **Ayman Zejli et Loic Magnan**
Projet encadré par : **Cristophe de Vaulx**

1. Résumé

Ce projet consiste à développer une version améliorée du jeu **2048** en **C++**, en utilisant **Qt** pour l'interface graphique et une **modélisation UML** pour concevoir l'architecture logicielle. Il vise à renforcer les compétences en **programmation orientée objet**, en optimisation algorithmique et en développement d'interfaces interactives.

Le jeu repose sur une grille 4x4, où le joueur doit déplacer et fusionner des tuiles de même valeur pour atteindre 2048, tout en conservant les règles classiques du jeu. Cependant, cette version apporte des améliorations et des bonus pour enrichir l'expérience.

Le développement a nécessité de relever plusieurs défis techniques, notamment la gestion efficace des entrées utilisateur, l'optimisation des mécanismes de fusion et de déplacement, ainsi que l'amélioration des animations et du rendu visuel pour garantir une interface fluide et agréable à utiliser. Un diagramme de Gantt a été mis en place afin de structurer les différentes phases du projet et assurer une progression efficace.

Ce projet aboutit à un jeu fonctionnel et optimisé, respectant fidèlement les mécaniques du 2048, tout en y ajoutant des éléments innovants pour une expérience plus dynamique et immersive.

Mots-clés : 2048, jeu, C++, Qt, UML, programmation orientée objet, interface graphique, optimisation, animations, interactions utilisateur, développement logiciel.

Table des matières

2. Introduction

Le **2048** est un jeu de réflexion connu pour son accessibilité et sa dimension stratégique. Dans le cadre du cursus des étudiants d'**IMDS**, ce projet avait pour objectif de développer une version personnalisée en **C++** avec la bibliothèque **Qt**, combinant ainsi **programmation orientée objet** et développement d'interfaces graphiques.

L'enjeu n'était pas seulement de reproduire fidèlement le jeu original, mais aussi d'approfondir les principes de conception logicielle à travers la modélisation UML et l'utilisation d'outils permettant une structuration rigoureuse du développement.

L'accent a été mis sur la création d'une application stable, performante et intuitive, tout en relevant plusieurs défis techniques, notamment l'optimisation des déplacements et fusions des tuiles, ainsi que l'amélioration de l'affichage pour une expérience fluide et agréable.

Ce document présente en détail les différentes étapes du projet, les défis rencontrés et les enseignements tirés de cette expérience.

3. Présentation du jeu

Le **2048** est un jeu de réflexion basé sur une grille **4x4**, où chaque case peut contenir une tuile numérotée, initialement de valeur **2 ou 4**. Le joueur doit déplacer ces tuiles dans **quatre directions** (haut, bas, gauche, droite), ce qui entraîne leur fusion lorsqu'elles portent la même valeur. Le jeu, simple au début, se complexifie de plus en plus, du fait du manque de place pour faire bouger les tuiles, et des erreurs de manipulation possibles, pouvant entraîner un blocage des tuiles et donc la fin du jeu à plus ou moins long terme, selon l'habileté du joueur. La partie est gagnée lorsqu'une tuile portant la valeur « 2048 » apparaît sur la grille, d'où le nom du jeu1,2.

Mon2048 est un jeu de logique addictif reposant sur la fusion de tuiles numérotées. Le joueur interagit avec une grille de 4x4 dans laquelle chaque mouvement produit :

- Un glissement des tuiles dans la direction choisie.
- Une fusion des tuiles adjacentes de même valeur, doublant ainsi leur somme.
- L'apparition aléatoire d'une nouvelle tuile avec une valeur de 2 ou 4.

L'esthétique simplifiée de Mon2048 le rend accessible à tous, tandis que ses mécaniques de jeu offrent une profondeur stratégique qui saura captiver les joueurs les plus exigeants.

4. Diagramme de Gantt

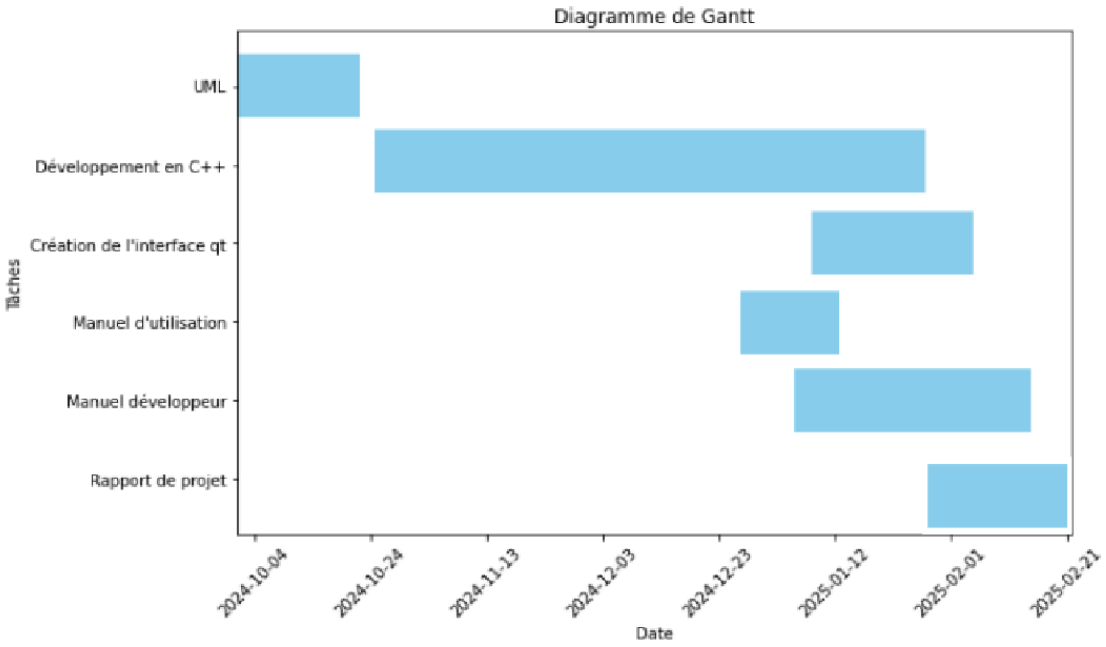

Figure 1 - diagramme de Gantt

Analyse des tâches :

- **UML** : Débute début octobre 2024 et se termine vers la mi-octobre.
- **Développement en C++** : Démarre juste après UML et s'étend jusqu'à début décembre 2024.
- **Création de l'interface Qt** : Commence vers début décembre et dure jusqu'à la mi-décembre.
- **Manuel d'utilisation** : Se déroule sur une courte période en janvier 2025.
- **Manuel développeur** : Débute après le manuel d'utilisation et s'étale sur une période plus longue.
- **Rapport de projet** : La dernière tâche du projet, qui commence en février 2025 et s'achève vers la fin du mois.

L'ordonnancement des tâches montre une progression logique du développement du projet (**cf. figure 1**), en commençant par la conception (UML) avant d'entrer dans le développement (C++ et interface Qt), puis en terminant avec la documentation et le rapport.

5. Présentation de votre travail

Ce projet a été mené de manière rigoureuse afin d'assurer un développement structuré et cohérent. Dès les premières étapes, une **modélisation UML** a été réalisée pour définir l'architecture logicielle et les interactions entre les différentes classes du jeu. Cette phase préparatoire a facilité la transition vers l'implémentation en **C++**, en posant des bases solides pour la gestion des règles et de la logique du **2048**.

La première étape du développement a consisté à programmer le **jeu en console**, en mettant en place les déplacements des tuiles, la fusion des valeurs identiques, ainsi que les conditions de fin de partie. Une fois cette mécanique fonctionnelle, l'accent a été mis sur la conception d'une **interface utilisateur avec Qt**, visant à rendre l'expérience de jeu fluide, agréable et ergonomique.

5.1. Défis rencontrés

Au fil du développement, plusieurs obstacles techniques ont dû être surmontés. La gestion des mouvements et des fusions de tuiles s'est révélée plus complexe que prévu, nécessitant plusieurs ajustements pour éviter les erreurs de doublons et garantir un comportement conforme aux règles du jeu. Après plusieurs tests et corrections, un algorithme optimisé a été mis en place pour assurer une exécution correcte des actions du joueur.

L'intégration de Qt a également représenté un défi majeur, notamment en ce qui concerne la gestion des événements clavier et le rendu graphique de la grille. L'affichage dynamique des tuiles et l'actualisation en temps réel ont demandé des ajustements pour garantir une réactivité optimale du jeu. En particulier, la **gestion des pixels** pour chaque tuile a nécessité une réflexion sur l'espace disponible à l'écran et sur la manière de disposer les éléments afin d'éviter des conflits visuels ou des problèmes de dimensionnement. Trouver l'équilibre entre la taille des tuiles, l'espace entre elles et la lisibilité a été un défi technique important, d'autant plus que l'application devait être responsive et s'adapter à différentes résolutions d'écran. Le placement et le redimensionnement des éléments ont ainsi été minutieusement ajustés pour garantir une expérience utilisateur fluide et intuitive.

Un autre défi majeur a été le choix de la **structure de données** appropriée. Afin de gérer efficacement la grille de jeu, qui doit pouvoir évoluer à chaque mouvement ou fusion de tuiles, un tableau de Qlabel[4][4] été choisie. Cette structure permet de représenter facilement les différentes tuiles et d'appliquer les transformations nécessaires lors des déplacements et fusions. Cependant, il a été important de veiller à ce que cette structure permette une manipulation efficace des données, notamment pour vérifier la validité des mouvements ou les fusions possibles sans introduire de redondance.

Malgré ces difficultés, les solutions apportées ont permis d'aboutir à une version fonctionnelle et fluide du **2048**, respectant les exigences du projet et offrant une expérience utilisateur agréable.

6. Bilan

Ce projet a permis de développer une version pleinement fonctionnelle du **2048**, respectant fidèlement les mécaniques du jeu original. Les objectifs principaux ont été atteints, notamment en renforçant notre maîtrise du **langage C++**, de la **programmation orientée objet**, ainsi que du **Framework Qt** pour l'interface graphique.

Bien que le jeu soit pleinement fonctionnel, des améliorations restent possibles, notamment en termes de design visuel et d'ajout d'animations plus fluides pour enrichir l'expérience utilisateur. Il serait également intéressant d'intégrer des **bonus** tels que des multiplicateurs de points **x2** ou **x4**, permettant aux joueurs de progresser plus rapidement et d'ajouter une dimension stratégique supplémentaire au jeu.

En ce qui concerne l'évolution du projet, toutes les fonctionnalités principales ont été implémentées avec succès. Cependant, certaines améliorations peuvent encore être apportées pour rendre le projet encore meilleur.

6.1. Fonctionnalités complémentaires

Dans le cadre de l'enrichissement du jeu, nous avons ajouté des multiplicateurs de valeurs 2, 4 et 1024. Ces nouveaux éléments ont pour but de rendre le gameplay plus dynamique et stratégique. L'introduction de ces multiplicateurs permet aux joueurs d'augmenter rapidement leurs scores en fusionnant les tuiles adéquates, tout en introduisant une nouvelle dimension de réflexion sur les mouvements à effectuer. Ce changement vise à diversifier les choix tactiques des joueurs et à dynamiser les parties, rendant le jeu plus captivant et offrant davantage de possibilités pour optimiser les déplacements.

6.2. Améliorations et ajouts envisagés

Bien que le jeu fonctionne correctement, quelques améliorations sont à envisager, telles que l'ajout de petites animations pour la fusion des tuiles, à l'image du jeu original, afin de rendre le processus plus dynamique et agréable visuellement. Des optimisations concernant la fluidité des déplacements, l'efficacité des calculs et la gestion des entrées utilisateur sont également envisageables pour affiner l'expérience de jeu. En termes de nouvelles fonctionnalités, un mode "Score Max" pourrait être intégré, incitant les joueurs à optimiser leurs déplacements et à maximiser leur score. De plus, il serait intéressant de **conserver les scores en mémoire** pour permettre aux joueurs de suivre leur progression et d'ajouter une dimension compétitive avec un **classement des meilleurs scores**.

Enfin, pour accroître la complexité du jeu et offrir un défi supplémentaire, la possibilité de **modifier la taille de la grille** pourrait être ajoutée, permettant de jouer sur des grilles de plus grandes dimensions et d'aller au-delà de la tuile **2048**, ouvrant ainsi de nouvelles perspectives de jeu.

6.3. Retour d'expérience sur le projet

Cette expérience a été particulièrement enrichissante, tant sur le plan technique que collaboratif. Personnellement, elle m'a permis de renforcer mes compétences en C++ et en POO, tout en découvrant Qt pour la gestion des interfaces graphiques. L'utilisation de la modélisation UML a également été un atout précieux dans l'organisation et la conception du projet.

Concernant le travail en groupe, une communication fluide et une répartition claire des tâches ont été essentielles à la réussite du projet. Chaque membre a pu mettre en valeur ses compétences spécifiques et contribuer activement à l'avancement du développement. Sur le plan organisationnel, une planification rigoureuse a permis de respecter les délais et d'assurer un suivi efficace des tâches. La gestion du temps a également été un facteur clé : bien que le délai alloué ait été suffisant pour implémenter les fonctionnalités de base, un peu plus de temps aurait permis de perfectionner les animations et d'optimiser les performances du jeu, rajouter des améliorations et rendre le jeu plus immersif et original.

De plus, le fait de devoir gérer ce projet tout en poursuivant les cours a offert une excellente opportunité pour améliorer nos compétences organisationnelles. Ce défi nous a forcés à établir des priorités, à gérer plusieurs tâches simultanément et à nous assurer que nous respections à la fois les exigences académiques et les objectifs du projet. Cela a renforcé notre capacité à nous organiser de manière plus efficace et à travailler sous pression.

6.4. Compétences développées

Ce projet a permis de renforcer plusieurs compétences cruciales. La maîtrise du **C++** et l'utilisation de **Qt** pour le développement d'interfaces graphiques ont constitué des bases solides pour notre travail. La découverte de Qt nous a permis de mieux comprendre les concepts liés à la création d'interfaces interactives, offrant une grande liberté pour personnaliser et améliorer l'expérience utilisateur. L'analyse et la conception à l'aide d'UML ont facilité l'organisation du projet, en assurant une structuration claire et efficace des différentes étapes de développement.

En outre, la gestion des interfaces graphiques avec Qt, ce qui nous a permis d'approfondir nos connaissances sur l'affichage dans une fenêtre graphique, les interactions avec l'utilisateur. Ce projet a également été une excellente occasion de travailler sur un **gros projet** en développant un jeu complet, ce qui nous a permis d'aborder des aspects techniques variés, tout en renforçant nos compétences en **programmation orientée objet** et en **optimisation des performances**. La diversité des tâches et l'ampleur du projet ont été des facteurs très enrichissants, contribuant à notre évolution en tant que développeurs.

7. Conclusion

Ce projet nous a permis de développer des compétences solides en programmation et en conception d'interfaces graphiques, tout en approfondissant nos connaissances en modélisation UML. Travailler ensemble nous a permis d'appliquer des concepts théoriques dans un contexte pratique et de créer une application réelle, ce qui a renforcé notre maîtrise de la programmation orientée objet. L'utilisation de C++ et de Qt nous a permis de bien appréhender les aspects techniques du développement d'un jeu, en particulier la gestion des événements, des animations et de l'affichage graphique.

Bien que nous ayons rencontré quelques défis techniques, tels que la gestion des fusions et déplacements des tuiles ou l'optimisation de l'affichage graphique, nous avons su les surmonter. Cela nous a permis de parvenir à une version fonctionnelle du jeu, offrant une expérience agréable et fluide pour l'utilisateur.

Ce projet a été une excellente occasion de renforcer nos compétences en développement logiciel et de créer un jeu interactif et complet, tout en prenant conscience des défis à la gestion d'un projet de cette durée. Travailler sur ce jeu a également été un excellent moyen de perfectionner nos capacités organisationnelles, car nous avons dû structurer nos tâches sur plusieurs semaines, respecter des délais et ajuster nos priorités en fonction des imprévus.

Enfin, cette expérience a renforcé nos capacités à collaborer efficacement au sein d'un groupe, à échanger des idées et à résoudre ensemble les problèmes techniques. Elle a aussi mis en évidence l'importance de la communication et de la planification pour mener à bien un projet complexe. Ce projet est désormais un excellent exemple de notre capacité à gérer un développement logiciel complet, et il constitue un atout précieux à inclure dans notre CV. En résumé, ce projet n'a pas seulement été une opportunité d'appliquer nos connaissances techniques, mais aussi de développer des compétences précieuses en gestion de projet et en travail d'équipe, qui sont directement transférables dans le milieu professionnel.

8. Bibliographie

[1] CodeBlocks : https://www.codeblocks.org/

[2] Qt : https://www.qt.io/

[3] Documentation Qt Creator : https://doc.qt.io/qtcreator/creator-how-to-install.html

[4] Qt Widgets : https://doc.qt.io/qt-6/qtwidgets-index.html

[5] Qt Widgets C++ Classes : https://doc.qt.io/qt-6/qtwidgets-module.html

[6] Qlabel : https://doc.qt.io/qt-6/qlabel.html