1. Sommaire
2. Partie analyse et conception du logiciel
3. Les cas d’utilisation

Ce diagramme de cas d’utilisation a pour but de présenter les différentes fonctionnalités disponibles pour l’utilisateur lors de l’utilisation du logiciel « 2048 ».

Ces fonctionnalités sont celles que nous avons imaginées afin d’offrir le jeu le plus complet et agréable possible. Cependant, par manque de temps, nous n’avons pas pu toutes les implémenter.

Nous allons donc présenter le fonctionnement de ce logiciel ainsi que les différents cas d’utilisation pouvant se présenter à l’utilisateur, dans le cas où nous aurions implémenté toutes les fonctionnalités présentes dans ce diagramme.

Une image contenant texte, diagramme, croquis, dessin

Description générée automatiquement

Diagramme de cas d’utilisation 2048

Lors du lancement du logiciel, l’utilisateur atterrit sur une page d’accueil, via laquelle il peut configurer la partie et/ou s’authentifier.

La configuration permet de choisir la taille de la grille (le nombre de lignes, qui est égal au nombre de colonnes) et un objectif à atteindre pour gagner la partie (obtenir le bloc 2048, 4096, etc.).

De plus, l’utilisateur peut s’authentifier, c’est-à-dire se créer des identifiants (nom d’utilisateur et mot de passe) ou se connecter avec ses identifiants déjà existants.

Dans le cas où l’utilisateur s’authentifie, alors on propose à l’utilisateur la possibilité de reprendre une partie parmi l’ensemble des parties qu’il a sauvegardées.

Après que l’utilisateur a créer une nouvelle partie ou repris une de ses sauvegardes, il commence alors à jouer au 2048.

Il peut alors jouer (déplacer les blocs), sauvegarder sa partie (en lui assignant un nom de sauvegarde), consulter l’aide du jeu, ou encore réinitialiser la partie (remettre la grille à zéro).

De plus, l’utilisateur peut quitter la partie et revenir à l’écran d’accueil. S’il n’a pas sauvegardé sa partie, alors elle sera perdue et il ne pourra pas la reprendre.

L’ensemble de ces fonctionnalités permettent d’avoir un jeu dynamique et respectant les principes de la plupart des jeux existants (authentification, sauvegarde, configuration du jeu, etc.).

1. Le diagramme de classes

Pour coder ce logiciel, nous avons utilisé le langage C++, et en particulier l’API Qt Creator.

Le C++ offre la possibilité d’utiliser la programmation orientée objet, qui implique donc l’implémentation de classes, qui permettent de modéliser diverses entités du logiciel/jeu.

Afin de présenter ces différentes classes et d’avoir une première approche de leur implémentation, nous avons illustrer graphique leur structure via ce diagramme de classe.

Une image contenant texte, capture d’écran, diagramme, Police

Le contenu généré par l’IA peut être incorrect.

Diagramme de classe 2048

1. La classe Coordonnees

Cette classe, utilisée par la plupart des autres classes de ce logiciel, permet de modéliser les coordonnées X et Y relativement aux coordonnées de la fenêtre principale « MainWindow » implémentée via Qt Creator.

Ces coordonnées sont exprimées en pixel, et le coin en haut à gauche de la fenêtre principale correspondant aux coordonnées (0, 0).

1. La classe MainWindow

Cette classe permet d’instancier la fenêtre principale affichée à l’écran lors du lancement du programme.

Elle est implémentée via l’API Qt, et dispose donc de caractéristiques propres à Qt, telles que des méthodes pour définir sont affichage, capter des évènements du clavier, etc.

C’est dans cette fenêtre que sont affichés tous les éléments du code, comme la grille du jeu et le score.

On attribue à chaque fenêtre principale une largeur et hauteur, qui permet de définir sa taille lorsqu’elle est affichée à l’écran de l’ordinateur. Si sa taille dépasse les dimensions de l’écran, alors elle ne sera pas affichée dans son entièreté.

1. La classe Bloc et ses classes dérivées BlocNumerote et BlocNul

La classe bloc est une classe abstraite permettant de modéliser les cases du 2048.

Cette classe mère permet de définir les attributs qui sont communs aux différents types de bloc (aux différentes classes héritées).

Par exemple, chaque bloc possède une couleur, des coordonnées, une fonction pour dessiner le bloc à l’écran, etc.

La classe héritée BlocNumerote correspond aux cases du jeu possédant une valeur (une puissance de 2), tandis que la classe CaseVide modélise les cases de la grille dans lesquelles il n’y a pas de blocs numérotés.

Les cases vides sont automatiquement affichées en gris, à l’inverse des cases numérotées dont la couleur dépend de leur valeur.

1. La classe Grille

C’est la classe la plus conséquente du projet en termes de code.

Elle représente la grille du jeu, c’est-à-dire un carré composé de cases numérotées ou vides.

Chaque grille possède un nombre de lignes, qui est égal à son nombre de colonnes car la grille est un carré.

De plus, afin d’afficher la grille à l’écran, on lui attribue des coordonnées et une taille, en pixel, qui sont calculés en fonction de la taille de la fenêtre principale.

En termes de code, la grille est modélisée par une matrice carrée de pointeurs vers des blocs (qu’ils soient numérotés ou vides).

La matrice permet une représentation intuitive de la grille (car elle peut être vue comme un carré composé de blocs), tandis que les pointeurs vers des blocs permettent une gestion dynamique de la mémoire.

Enfin, la grille implémente toutes les fonctions permettant le déroulement de la partie, comme le déplacement des blocs, la création de nouvelles cases numérotées, la fusion de deux blocs, l’affichage de la grille du 2048, etc.

1. La classe Score

Enfin, la classe Score permet, comme son nom l’indique, de modéliser le score de la partie en cours.

Ce score est mis à jour via la classe Grille, et augmente à chaque fois que deux blocs numérotés sont fusionnés.

1. Le diagramme de séquences
2. Partie programmation du logiciel
3. L’architecture du logiciel

Afin de versionner et d’avoir une sauvegarde Cloud de notre projet, nous avons décidé de créer un répertoire distant « repository » GitHub.

Ce repository nommé « 2048-project », contient à sa racine plusieurs sous-dossiers ainsi que des fichiers propres à Git comme le « .gitignore », « README », etc.

Le projet Qt Creator est localisé dans le sous-dossier « src », qui contient donc la partie code du repository.

Pour l’ensemble des classes mères et héritées que nous avons implémentées, nous avons créé un couple (fichier d’entête « .h », fichier source « .cpp »).

De plus, ce dossier « src » contient le programme « main.cpp », permettant de lancer le logiciel.

Afin d’ouvrir le projet dans Qt Creator, il faut ouvrir le fichier « 2048-project.pro », permettant de construire le projet.

La prise en main de ce logiciel est expliquée via le fichier « README » inclus à la racine du projet.

1. Les points importants du code

s

1. Présentation des écrans