

Démineur

Dans le cadre du cours :

Programmation 3

Travail effectué par :

Marc-André Larocque Blais
Jean-François Duchesne

Travail remis à :

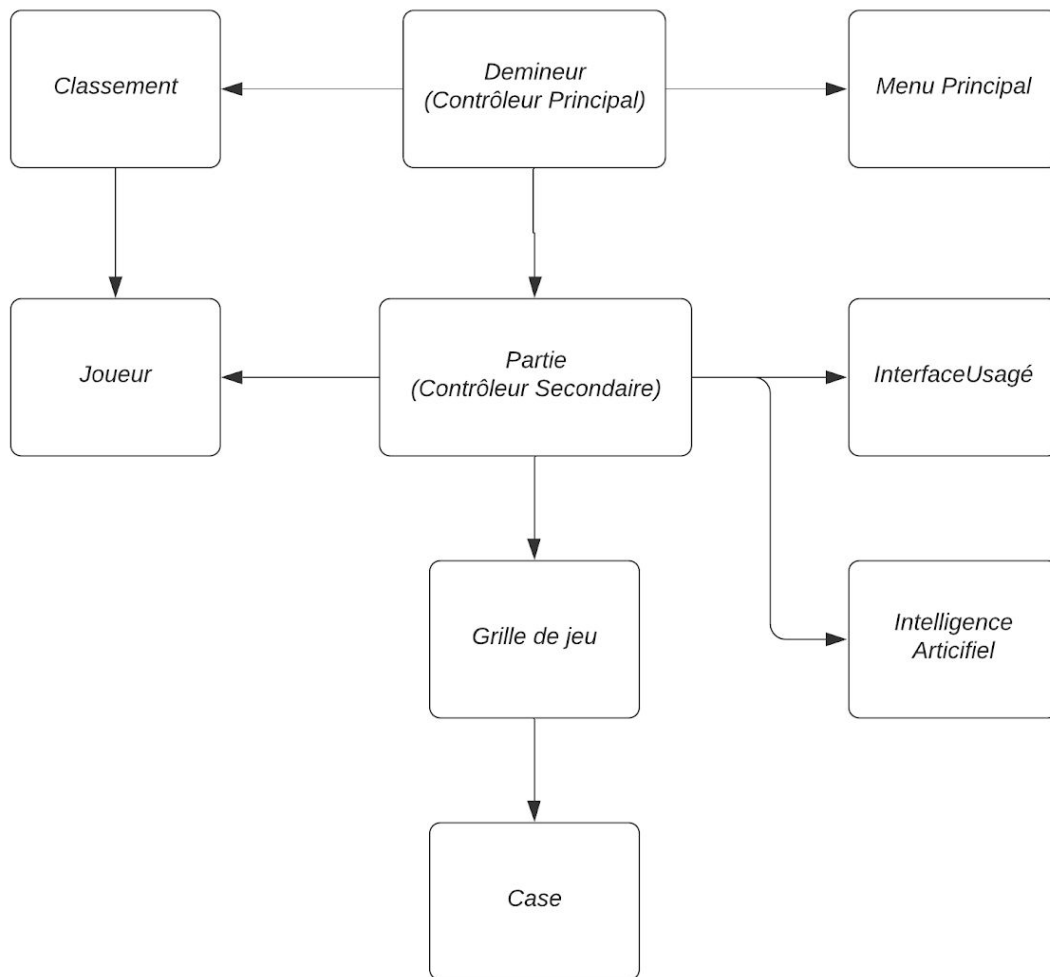
Charles Jacob

Cégep régional de Lanaudière à Joliette
Hiver 2020

Table des matières

Démineur	1
Table des matières	2
Diagramme des classes	3
Documentation des classes	4
Démineur	4
Menu Principal	4
Classement	5
Partie	5
Interface	6
AI	6
Joueur	7
Grille	7
Case	8

Diagramme des classes



Documentation des classes

Démineur

Nom :	Démineur
Desc :	Point d'entrée du programme. La classe Démineur charge son Menu (statique) au démarrage afin de guider le joueur parmi les options. Elle s'occupe aussi d'appeler la classe Classement afin de charger l'information des joueurs faisant déjà partie de la base de données. De là, on peut afficher les classements ou créer une partie avec un nouveau joueur ou un joueur existant.
Donnée membre :	m_Partie : Partie m_EnMarche : Bool [?] m_Classement : Classement
Méthode +:	Aucune

Menu Principal

Nom :	Menu
Desc :	Affiche les options au joueur.(Jouer, Classements, Quitter) Retourne un tableau contenant les paramètres de construction de la partie.
Donnée membre :	Aucune
Méthode +:	AfficherMenu(): OptionsDePartie(): char[3] {Grandeur, Difficulté, AI}

- Possibilité de changer le tableau pour du bitWise

Classement

Nom :	Classement
Desc :	Affiche les meilleurs temps (Top 10) par difficulté.
Donnée membre :	m_Joueurs : List
Méthode +:	AfficherClassement(); MettreAJour(NomJoueur, temps, diff);

Partie

Nom :	Partie
Desc :	Instancie un joueur et une grille du format reçu par le contrôleur Demineur . Elle agit comme contrôleur de partie en appelant les différentes méthodes nécessaire au bon déroulement de cette dernière.
Donnée membre :	Grille m_Grille Joueur m_Joueur AI m_AI String tempsEcoule
Méthode +:	obtenirMetadonneesDeLaPartieActuellementTerminee(): String partiInfo

- EnumDifficulté {facile = %xDeBombe, normal = %xDeBombe, difficile = %xDeBombe}
- EnumGrosueur {petit = length, moyen = length, grand = length}

Interface

Nom :	Interface
Desc :	Affiche la grille, le chronomètre, le nom du joueur, le mode de jeu, le nombre de bombes et le bouton AI si l'option AI a été activé lors de la création de la partie.
Donnée membre :	Aucune
Méthode +:	DessinerGrille(string grille); AfficherChronomètre(string temps); MessageVictoire() : string MessageDefaite() : string

AI

Nom :	AI
Desc :	Le AI peut être instancié ou non lors de la création de la partie. Une fois créé, il peut être appelé manuellement afin de jouer un coup pour aider le joueur ou si la partie créée est en mode automatique, il résoudra la grille au complet.
Donnée membre :	Aucune
Méthode +:	jouerUnCoup() : short[2]

Joueur

Nom :	Joueur
Desc :	Instancie un nouveau joueur et s'occupe de la reconstruction des joueurs existants à partir d'un fichier texte. Seulement les scores sans Ai seront enregistrés.
Donnée membre :	m_Nom : string m_Scores : string[9] { {temps p_Facile, temps p_Moyen, temps p_Difficile} {temps m_Facile, temps m_Moyen, temps m_Difficile} {temps g_Facile, temps g_Moyen, temps g_Difficile} } m_List : Joueur
Méthode +:	obtenirNom() : string m_Nom obtenirScore(string m_Nom) : String[] m_Score modifierScore(char grosseur, char difficulté, string temps) : bool modifier toString() : string * Servira lors de l'écriture dans le fichier texte.

Grille

Nom :	Grille
Desc :	Instancie un tableau de case et disperse les bombes.
Donnée membre :	Case[] champ short nbBombeGrille
Méthode +:	genererGrille(uint grosseur) : Void dispenserBombes(short nbBombeGrille) : Void

Case

Nom :	Case
Desc :	Initialise la valeur d'une case soit bombe ou non et combien de bombe l'entour
Donnée membre :	bool bombe char nbDanger bool[8] casesVoisines
Méthode +:	esTuBombe() : Bool nbDanger() : Char