PROJET C++

DOCUMENTATION TECHNIQUE :

Board.h :

Cette classe représente le plateau de jeu.

- Attributs privés :
- boardSize (int): Taille du plateau.
- territories (std::vector<std::vector<int>>): Zones de territoire pour chaque joueur.
- players (std::vector<Player>): Liste des joueurs.

Méthodes publiques :

- Board(int numPlayers): Constructeur initialisant le plateau avec un nombre de joueurs donné.
- getPlayer(int index) -> Player&: Récupère un joueur par son indice.
- displayBoard(std::vector<std::vector<int>> gridToDisplay): Affiche le plateau de jeu.
- setGrid(std::vector<std::vector<int>> newGrid): Définit une nouvelle grille de territoires.
- getGrid() -> std::vector<std::vector<int>>: Récupère la grille de territoires.
- getBoardSize() -> int: Récupère la taille du plateau.
- putPlayers(): Place les joueurs sur le plateau.
- getPlayersList() -> std::vector<Player>: Récupère la liste des joueurs.

Méthodes privées :

- divideTerritory(): Divise le plateau en zones de territoire pour chaque joueur.
- placePlayers(): Place les joueurs sur le plateau en fonction des zones de territoire.

Board.cpp

Implémentation des méthodes de la classe Board

Board::Board(int numPlayers):

- Ce constructeur initialise la taille du plateau en fonction du nombre de joueurs.
- Il crée une grille de territoires initialisée avec des valeurs par défaut ('.').
- Il réserve la mémoire pour stocker les joueurs.
- Il crée et initialise les joueurs avec des numéros uniques.

Board::getGrid() -> std::vector<std::vector<int>>:

- Renvoie la grille de territoires.

Board::setGrid(std::vector<std::vector<int>> newGrid):

- Définit une nouvelle grille de territoires.

Board::displayBoard(std::vector<std::vector<int>> gridToDisplay):

- Affiche le plateau de jeu avec les joueurs et les zones de territoire.

Board::getBoardSize() -> int:

- Renvoie la taille du plateau.

Board::getPlayer(int index) -> Player&:

- Renvoie une référence vers un joueur par son indice.

Board::putPlayers():

- Place les joueurs sur la grille de territoires.

Board::getPlayersList() -> std::vector<Player>:

- Renvoie la liste des joueurs.

Board::divideTerritory():

- Divise le plateau en zones de territoire pour chaque joueur.
- Les joueurs sont répartis sur différentes zones sans chevauchement.

Board::placePlayers():

- Place les joueurs sur la grille en fonction des zones de territoire.
- Les coordonnées des joueurs sont mises à jour selon leur emplacement sur la grille.

Player.h :

Déclaration de la classe Player

Attributs :

- playerName : Chaîne de caractères représentant le nom du joueur.
- playerNumber : Entier représentant le numéro du joueur.
- xCoordinate et yCoordinate : Coordonnées X et Y du joueur sur le plateau.
- playerColor : Entier représentant la couleur attribuée au joueur.

Méthodes publiques :

- Constructeur Player() : Initialise les attributs du joueur avec des valeurs par défaut.
- setName(const std::string& name) : Définit le nom du joueur.
- getName() const -> std::string : Renvoie le nom du joueur.
- setNumber(int number) : Définit le numéro du joueur.
- getNumber() const -> int : Renvoie le numéro du joueur.
- setCoordinates(int x, int y) : Définit les coordonnées du joueur sur le plateau.
- getXCoordinate() const -> int : Renvoie la coordonnée X du joueur.
- getYCoordinate() const -> int : Renvoie la coordonnée Y du joueur.
- setColor(int color) : Définit la couleur du joueur.
- getColor() const -> int : Renvoie la couleur du joueur.

Player.cpp :

Implémentation des méthodes de la classe Player

Constructeur Player::Player() :

- Initialise les attributs du joueur (playerName, playerNumber, xCoordinate, yCoordinate, playerColor) avec des valeurs par défaut.

Player::setName(const std::string& name) :

- Définit le nom du joueur avec la chaîne de caractères name.

Player::getName() const -> std::string :

- Renvoie le nom du joueur.

Player::setNumber(int number) :

- Définit le numéro du joueur avec la valeur number.

Player::getNumber() const -> int :

- Renvoie le numéro du joueur.

Player::setCoordinates(int x, int y) :

- Définit les coordonnées X et Y du joueur sur le plateau.

Player::getXCoordinate() const -> int :

- Renvoie la coordonnée X du joueur.

Player::getYCoordinate() const -> int :

- Renvoie la coordonnée Y du joueur.

Player::setColor(int color) :

- Définit la couleur du joueur avec la valeur color.

Player::getColor() const -> int :

- Renvoie la couleur du joueur.

Tile.h :

Déclaration de la classe Tile

Attributs :

- player : Entier représentant le joueur associé à la tuile.
- coordinates : Tuple d'entiers représentant les coordonnées de la tuile.
- shape : Vecteur bidimensionnel représentant la forme de la tuile.

Méthodes publiques :

- Constructeur Tile(std::vector<std::vector<int>> initShape) : Initialise une tuile avec une forme donnée.
- getShape() -> std::vector<std::vector<int>> : Renvoie la forme de la tuile.
- setShape(std::vector<std::vector<int>> newShape) : Définit une nouvelle forme pour la tuile.
- displayTile() : Affiche la forme de la tuile dans la console.
- reverseTile() : Inverse la forme de la tuile.
- rotateTile() : Effectue une rotation de 90 degrés de la forme de la tuile.

Tile.cpp

Implémentation des méthodes de la classe Tile

Constructeur Tile::Tile(std::vector<std::vector<int>> initShape) :

- Initialise la tuile avec la forme spécifiée par initShape.

Tile::getShape() -> std::vector<std::vector<int>> :

- Renvoie la forme actuelle de la tuile.

Tile::setShape(std::vector<std::vector<int>> newShape) :

- Définit une nouvelle forme pour la tuile.

Tile::displayTile() :

- Affiche la forme de la tuile dans la console en utilisant des caractères spécifiques.

Tile::reverseTile() :

- Inverse la forme actuelle de la tuile.

Tile::rotateTile() :

- Effectue une rotation de 90 degrés de la forme actuelle de la tuile.

GameManager.h :

Déclaration de la classe GameManager

Attributs :

- gameBoard : Instance de la classe Board représentant le plateau de jeu.
- actualTile : Entier représentant l'index de la tuile actuelle.
- currentRound : Entier représentant le tour de jeu actuel.
- tileList : Vecteur de tuiles disponibles pour le jeu.
- currentPlayer : Entier représentant le joueur actuel.

Méthodes publiques :

- GameManager() : Constructeur par défaut.
- getCurrentPlayer() -> int : Renvoie l'index du joueur actuel.
- setCurrentPlayer() : Définit le joueur actuel.
- setBoard(Board boardToSet) : Définit le plateau de jeu.
- setCurrentRound() : Incrémente le tour actuel.
- getCurrentRound() -> int : Renvoie le tour actuel.
- getBoard() -> Board : Renvoie l'instance du plateau de jeu.
- setTileList(std::vector<Tile> newTileList) : Définit la liste de tuiles disponibles.
- getTileList() -> std::vector<Tile> : Renvoie la liste des tuiles disponibles.

- getActualTile() -> int : Renvoie l'index de la tuile actuelle.
- setActualTile(int newActualTile) : Définit la tuile actuelle.
- initializeTiles() -> std::vector<Tile> : Initialise et renvoie une liste de tuiles prédéfinies.
- placeTile(int currentPlayer, int currentRound, Tile tileToPlace, int x, int y, GameManager gameManager) -> bool : Place une tuile sur le plateau de jeu.
- checkCollide(std::vector<std::vector<int>> boardToCheck, std::vector<std::tuple<int,int>> tupleListToCheck) -> bool : Vérifie si une tuile entrerait en collision avec une autre.
- printColorCodes() : Affiche les codes de couleur disponibles.
- startGame() -> int : Lance le jeu et initialise le plateau.
- putLastTile(std::vector<Player> playersList) : Place la dernière tuile pour chaque joueur.
- getLastTileCoordinates() -> std::tuple<int,int> : Renvoie les coordonnées de la dernière tuile.

GameManager.cpp

Méthodes implémentées

GameManager::GameManager() :

- Description : Constructeur par défaut de la classe GameManager.
- Fonctionnement : Initialise les attributs currentRound, currentPlayer, tileList, gameBoard et actualTile.

GameManager::setCurrentPlayer() :

- Description : Définit le joueur actuel en fonction du nombre de joueurs dans la partie.
- Fonctionnement : Incrémente currentPlayer jusqu'à ce qu'il atteigne le nombre de joueurs, puis revient à 1.

GameManager::getCurrentPlayer() -> int :

- Description : Renvoie l'index du joueur actuel.
- Fonctionnement : Renvoie la valeur actuelle de currentPlayer.

GameManager::setBoard(Board boardToSet) :

- Description : Définit le plateau de jeu.
- Fonctionnement : Initialise gameBoard avec le plateau boardToSet.

GameManager::setCurrentRound() :

- Description : Incrémente le tour de jeu actuel.
- Fonctionnement : Incrémente la valeur de currentRound de 1.

GameManager::getCurrentRound() -> int :

- Description : Renvoie le tour de jeu actuel.
- Fonctionnement : Renvoie la valeur actuelle de currentRound.

GameManager::getBoard() -> Board :

- Description : Renvoie le plateau de jeu.
- Fonctionnement : Renvoie l'instance de Board stockée dans gameBoard.

GameManager::setTileList(std::vector<Tile> newTileList):

- Description : Définit la liste des tuiles disponibles.
- Fonctionnement : Affecte newTileList à l'attribut tileList.

GameManager::getTileList() -> std::vector<Tile>:

- Description : Renvoie la liste des tuiles disponibles.
- Fonctionnement : Renvoie la valeur actuelle de tileList.

GameManager::getActualTile() -> int :

- Description : Renvoie l'index de la tuile actuelle.
- Fonctionnement : Renvoie la valeur actuelle de actualTile.

GameManager::setActualTile(int newActualTile) :

- Description : Définit l'index de la tuile actuelle.
- Fonctionnement : Affecte newActualTile à l'attribut actualTile.

GameManager::initializeTiles() -> std::vector<Tile>:

- Description : Initialise et renvoie une liste de tuiles prédéfinies.
- Fonctionnement : Crée et retourne une liste de tuiles avec des formes prédéfinies.

GameManager::placeTile(int currentPlayer, int currentRound, Tile tileToPlace, int x, int y, GameManager gameManager) -> bool:

- Description : Place une tuile sur le plateau de jeu.
- Fonctionnement : Vérifie d'abord si la tuile entrerait en collision avec une autre ou déborderait du plateau, puis la place sur le plateau si les conditions sont remplies.

GameManager::checkCollide(std::vector<std::vector<int>> boardToCheck, std::vector<std::tuple<int,int>> tupleListToCheck) -> bool:

- Description : Vérifie si une tuile entrerait en collision avec une autre.
- Fonctionnement : Parcourt la liste des coordonnées de la tuile à placer pour vérifier si elles entrent en collision avec des cases déjà occupées sur le plateau.

GameManager::printColorCodes() :

- Description : Affiche les codes de couleur disponibles.
- Fonctionnement : Affiche les codes de couleur disponibles pour les joueurs.

GameManager::startGame() -> int :

- Description : Initialise le jeu et le plateau de jeu.
- Fonctionnement : Lance le jeu en initialisant le plateau et les joueurs.

GameManager::putLastTile(std::vector<Player> playersList):

- Description : Place la dernière tuile pour chaque joueur.
- Fonctionnement : Demande à chaque joueur de placer sa dernière tuile.

GameManager::getLastTileCoordinates() -> std::tuple<int,int>:

- Description : Renvoie les coordonnées de la dernière tuile.
- Fonctionnement : Demande et valide les coordonnées pour placer la dernière tuile.

GameUi.h :

Méthodes déclarées

void displayCurrentTile(int actualTile, std::vector<Tile> tilesList):

- Description : Affiche la tuile actuelle.
- Paramètres :
    - actualTile : Index de la tuile actuelle.

        - tilesList : Liste des tuiles.

void displayNextTiles(int actualTile, std::vector<Tile> tilesList) :

- Description : Affiche les tuiles suivantes après la tuile actuelle.
- Paramètres :
    - actualTile : Index de la tuile actuelle.
    - tilesList : Liste des tuiles.

char getUserFirstChoice() :

- Description : Récupère le premier choix de l'utilisateur.
- Retour : Choix de l'utilisateur.

char getUserSecondChoice() :

- Description : Récupère le second choix de l'utilisateur.
- Retour : Second choix de l'utilisateur.

void displayFlippedTile(int actualTile, std::vector<Tile> tilesList):

- Description : Affiche la tuile après avoir été retournée.
- Paramètres :
    - actualTile : Index de la tuile actuelle.
    - tilesList : Liste des tuiles.

void displayRotatedTile(int actualTile, std::vector<Tile> tilesList):

- Description : Affiche la tuile après avoir été tournée.
- Paramètres :
    - actualTile : Index de la tuile actuelle.
    - tilesList : Liste des tuiles.

std::tuple<int,int> getVerifyCoordinate(GameManager &gameManager, int actualTile, std::vector<Tile> tilesList):

- Description : Récupère et valide les coordonnées pour placer une tuile sur le plateau.
- Paramètres :
    - gameManager : Instance de GameManager.
    - actualTile : Index de la tuile actuelle.
    - tilesList : Liste des tuiles.

- Retour : Tuple de coordonnées valides.

void actualTileSet(int actualTile, std::vector<Tile> tilesList, GameManager &gameManager):

- Description : Passe à la tuile suivante dans la liste.
- Paramètres :
    - actualTile : Index de la tuile actuelle.
    - tilesList : Liste des tuiles.
    - gameManager : Instance de GameManager.

void displayUserAction(GameManager &gameManager, int actualTile, std::vector<Tile> tilesList):

- Description : Affiche l'action de l'utilisateur sur la tuile actuelle.
- Paramètres :
    - gameManager : Instance de GameManager.
    - actualTile : Index de la tuile actuelle.
    - tilesList : Liste des tuiles.

GameUi.cpp :

Implémentation des méthodes

void GameUi::displayCurrentTile(int actualTile, std::vector<Tile> tilesList) :

- Description : Affiche la tuile actuelle.
- Paramètres :
    - actualTile : Index de la tuile actuelle.
    - tilesList : Liste des tuiles.

void GameUi::displayNextTiles(int actualTile, std::vector<Tile> tilesList) :

- Description : Affiche les tuiles suivantes après la tuile actuelle.
- Paramètres :
    - actualTile : Index de la tuile actuelle.
    - tilesList : Liste des tuiles.

char GameUi::getUserFirstChoice() :

- Description : Récupère le premier choix de l'utilisateur.

- Retour : Choix de l'utilisateur.

char GameUi::getUserSecondChoice() :

- Description : Récupère le second choix de l'utilisateur.
- Retour : Second choix de l'utilisateur.

void GameUi::displayFlippedTile(int actualTile, std::vector<Tile> tilesList):

- Description : Affiche la tuile après avoir été retournée.
- Paramètres :
    - actualTile : Index de la tuile actuelle.
    - tilesList : Liste des tuiles.

void GameUi::displayRotatedTile(int actualTile, std::vector<Tile> tilesList):

- Description : Affiche la tuile après avoir été tournée.
- Paramètres :
    - actualTile : Index de la tuile actuelle.
    - tilesList : Liste des tuiles.

std::tuple<int,int> GameUi::getVerifyCoordinate(GameManager &gameManager, int actualTile, std::vector<Tile> tilesList):

- Description : Récupère et valide les coordonnées pour placer une tuile sur le plateau.
- Paramètres :
    - gameManager : Instance de GameManager.
    - actualTile : Index de la tuile actuelle.
    - tilesList : Liste des tuiles.
- Retour : Tuple de coordonnées valides.

void GameUi::actualTileSet(int actualTile, std::vector<Tile> tilesList, GameManager &gameManager):

- Description : Passe à la tuile suivante dans la liste.
- Paramètres :
    - actualTile : Index de la tuile actuelle.
    - tilesList : Liste des tuiles.
    - gameManager : Instance de GameManager.

void GameUi::displayUserAction(GameManager &gameManager, int actualTile, std::vector<Tile> tilesList):

- Description : Affiche l'action de l'utilisateur sur la tuile actuelle.
- Paramètres :
    - gameManager : Instance de GameManager.
    - actualTile : Index de la tuile actuelle.
    - tilesList : Liste des tuiles.

Run.h

Classe Run

Méthodes :

void run() :

- Description : Méthode principale pour exécuter le jeu.
- Fonctionnement :
    - Initialise une partie (GameManager).
    - Démarre le jeu en appelant startGame() de GameManager.

Run.cpp

Fonctions et Méthodes :

void displayRound(int roundNumber) :

- Description : Affiche le numéro du round.
- Paramètres :
    - roundNumber : Numéro du round.

void displayPlayerName(const Player& player):

- Description : Affiche le nom du joueur.
- Paramètres :
    - player : Référence vers l'objet Player.

void displayPlayerAction(const Player& player):

- Description : Affiche l'action du joueur.
- Paramètres :
    - player : Référence vers l'objet Player.

void Run::run() :

- Description : Méthode principale pour exécuter le jeu.
- Fonctionnement :
    - Initialise une instance de GameManager.
    - Initialise une instance de GameUi.
    - Initialise une liste de tuiles (tileListInit) à partir de GameManager.
    - Récupère une liste de joueurs à partir du plateau.
    - Démarre le jeu en appelant startGame() de GameManager.
    - Effectue une boucle pour chaque round :
    - Affiche le numéro du round.
    - Chaque joueur joue à tour de rôle :
    - Affiche le nom du joueur.
    - Affiche l'action du joueur en utilisant displayUserAction() de GameUi.
    - Actualise le round et le joueur actuel.
    - Une fois les rounds effectués, les joueurs placent une dernière tuile 1*1 en appelant putLastTile() de GameManager.

main.cpp :

Fonction main() :

- Description : Point d'entrée du programme.
- Fonctionnement :
    - Initialise une instance de la classe Run.
    - Appelle la méthode run() de l'objet start, démarrant ainsi le jeu.