

Tile Twister

“Regardez ces tuiles. On dirait des trucs anciens” — Jumanji

Introduction du sujet

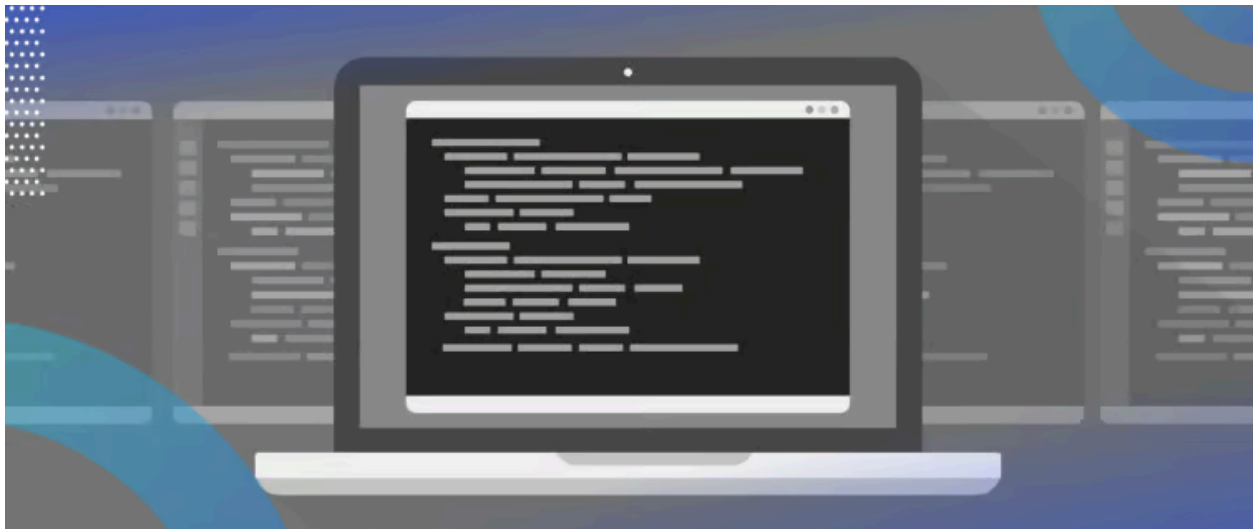
2048 est un jeu de réflexion créé en mars 2014 par le développeur indépendant italien **Gabriele Cirulli** et publié en ligne sous licence libre via GitHub le **9 mars 2014**. Sur une grille, des tuiles numérotées attendaient patiemment d'être fusionnées. Chaque joueur vise le chiffre : **2048**. Mais derrière ce nombre se cache bien plus qu'une simple somme mathématique. C'est un défi stratégique où chaque mouvement compte, où la planification et la patience sont aussi cruciales que la réflexion.





Étape 1 : logique de jeu

D'abord, plongeons dans l'univers du code pour implémenter les règles du jeu à l'aide des classes. Nous découvrirons les secrets de la bibliothèque standard et de ses containers. Grâce à la programmation orientée objet, architecturez votre projet et le structurer en plusieurs fichiers pour une gestion plus claire et modulaire.



Pour démarrer le jeu, initialiser une grille de jeu vide de **4x4** et générer deux tuiles aléatoires de valeur **2** ou **4**. Ensuite, permettre à l'utilisateur de déplacer les tuiles horizontalement (gauche/droite) et verticalement (haut/bas) en utilisant les touches du clavier. Fusionner les tuiles de même valeur lorsqu'elles entreront en collision après un déplacement, et générer une nouvelle tuile aléatoire après chaque déplacement valide. Le jeu détectera la fin lorsque la grille sera pleine sans aucun mouvement valide.

Créer des situations de jeu et observer les événements (haut/bas/gauche/droite) comparant les résultats obtenus avec ceux attendus pour tester l'intégrité du jeu.



Étape 2 : SDL

Donner vie à notre projet, le transformant de l'abstrait vers le visuel tout en conservant son essence et ses fonctionnalités d'origine – **un relooking numérique bien mérité !**

Pour la partie graphique, nous mettons en place un moteur de rendu simple, incluant une **classe Window** pour initialiser la SDL et créer une fenêtre avec des paramètres. Utiliser une classe abstraite **GameObject** pour représenter les éléments affichés à l'écran avec des coordonnées et une taille, permettant l'affichage sous forme géométrique ou par image SDL.

Réutiliser la logique du jeu pour afficher la grille avec des graphismes pour les tuiles, et permettre à l'utilisateur d'interagir avec le clavier pour déplacer les tuiles. Ajouter une animation sur les mouvements des tuiles pour une expérience visuelle immersive.



Bien sûr, nous n'oublierons pas l'importance des tests d'intégration pour éviter les régressions tout au long du développement. Nous les mettons en place avec autant d'enthousiasme qu'un détective traquant les bugs, assurant ainsi la solidité et la fiabilité de notre création.

Pour aller plus loin...

Afin de rendre le jeu plus "fun" :

- Ajouter une animation de mouvement des tuiles
- Permettre sauvegarde/chargement de partie
- Objectif différent (atteindre 1024, 8192)
- Tuiles spéciales (bloquées, bonus)

Compétences visées



- Installer et configurer son environnement de travail en fonction du projet.
- Développer des interfaces utilisateur
- Développer des composants métier
- Contribuer à la gestion d'un projet informatique
- Analyser les besoins et maquetter une application
- Définir l'architecture logicielle d'une application
- Préparer et exécuter les plans de tests d'une application

Rendu

Votre travail est évalué en présentation avec un support et une revue de code. Le slide doit être composé de :

- De l'organisation de votre équipe
- De vos problèmes rencontrés ainsi que les solutions apportées
- La démonstration jouable de votre jeu

Le projet est à rendre sur <https://github.com/prenom-nom/TileTwister>

Base de connaissances

- [Tutoriel STL](#)
- [Jeu du 2048](#)
- [Cours de C++ avancé](#)
- [Tutoriel SDL](#)