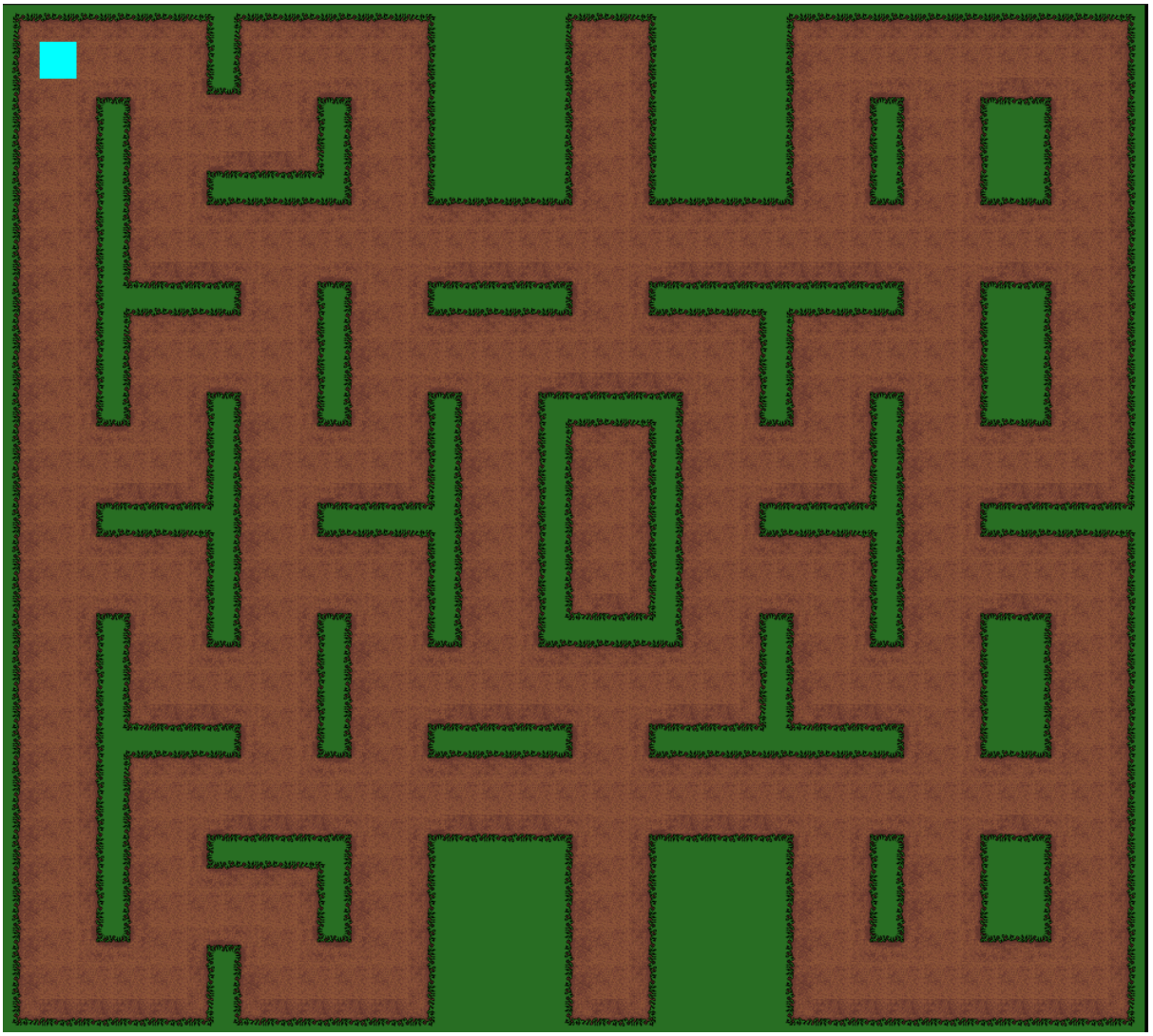
Rapport du jeu - Maze Runner



Groupe 3
ATHERLY Evan - CORDOVANA Enzo - PALOT Thomas - PANAIVA Jérémy

Sommaire

I. Introduction

Contexte de la SAE

Dans le cadre de notre BUT Informatique, nous avons réalisé une SAE portant sur le développement d'un jeu en C++ en utilisant Qt Creator pour la programmation et MinGL comme interface graphique. L'objectif principal était de mettre en pratique les compétences acquises en programmation, gestion de projet, et travail collaboratif.

Objectifs du projet

Le projet consistait à développer un jeu où un joueur incarne un policier devant attraper plusieurs voleurs (contrôlés par des bots). Le joueur se déplace sur une carte à l'aide des touches ZQSD pour toucher les voleurs et les éliminer.

Composition de l'équipe

L'équipe était composée de ATHERLY Evan - CORDOVANA Enzo - PALOT Thomas - PANAIVA Jérémy, chacun ayant des rôles définis :

- Programmation des mécanismes du jeu
- Gestion de l'interface graphique
- Documentation et tests

II. Analyse et Approche

Présentation du code de base

Le code initial, issu des travaux pratiques (TPs), nous a servi de fondation. Il contenait les éléments de base pour la gestion graphique et des exemples de logique applicative que nous avons adaptés et enrichis.

Liste des problèmes identifiés et solutions apportées

Bugs corrigés

- Correction des erreurs de collision entre le joueur et les voleurs.
- Résolution des problèmes de gestion des touches ZQSD en cas de saisie rapide.
- Désactivation du déplacement en diagonale.

Améliorations de robustesse

- Optimisation des algorithmes de déplacement des bots pour réduire la charge CPU.
- Validation des entrées utilisateur pour éviter les plantages.
- Communication MVC entre le clavier (contrôleur), afficher le personnage (vue) et la position du personnage (modèle).

Méthodologie utilisée pour le travail en groupe

Mise en place d'un système de gestion de version (Github, Git) pour synchroniser le travail.

Réunions hebdomadaires pour la répartition des tâches et le suivi de l'avancement.

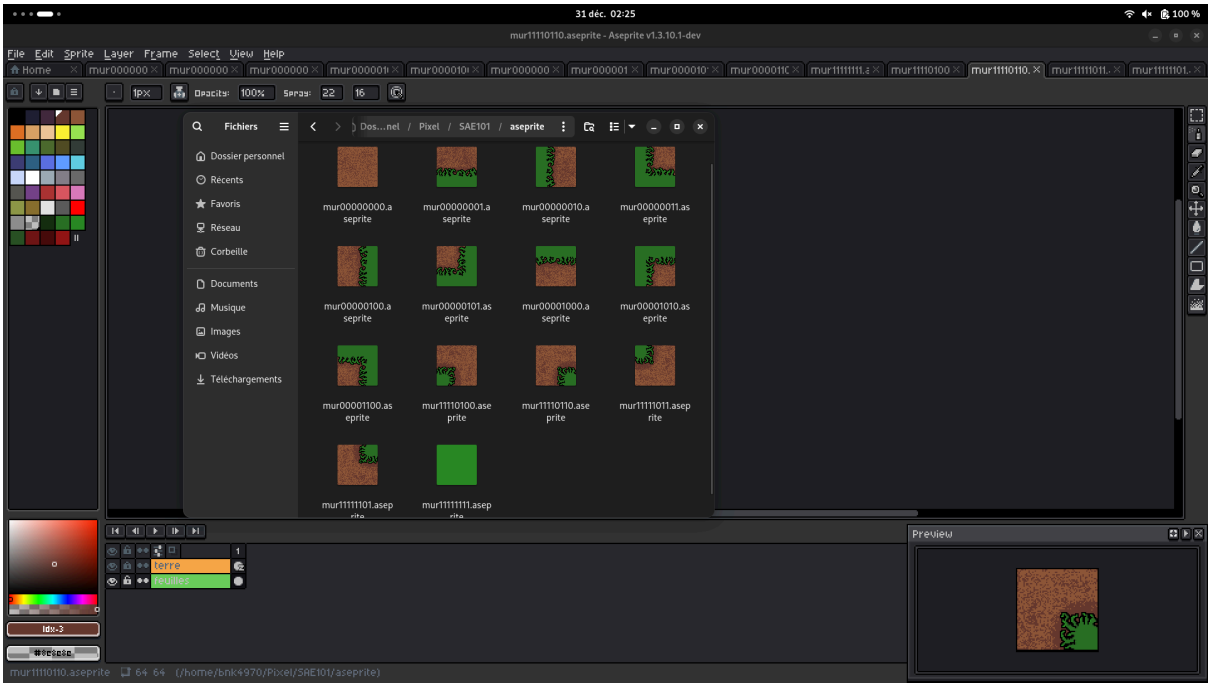
III. Fonctionnalités Implémentées

MinGL(On aurait voulu)

Nous avons utilisé MinGL pour gérer l'affichage graphique du jeu, ce qui a permis une expérience utilisateur fluide seulement si l’utilisation des sprites n’est pas abusive.

Avec les sprites, nous avons pu avoir une carte personnalisée grâce à l’application aseprite pour faire nos pixels arts et les convertir en .si2 avec l’outil développé par Casali Alain.

Après cela, l’idée est de parcourir une matrice pour obtenir des tuiles et observer les cellules voisines et déterminer quel type de mur afficher.



Fonctionnalités principales ajoutées

Lecture des fichiers de configurations

- Chargement dynamique des paramètres du jeu (taille de la carte, nombre de voleurs, etc.) depuis un fichier de configuration.

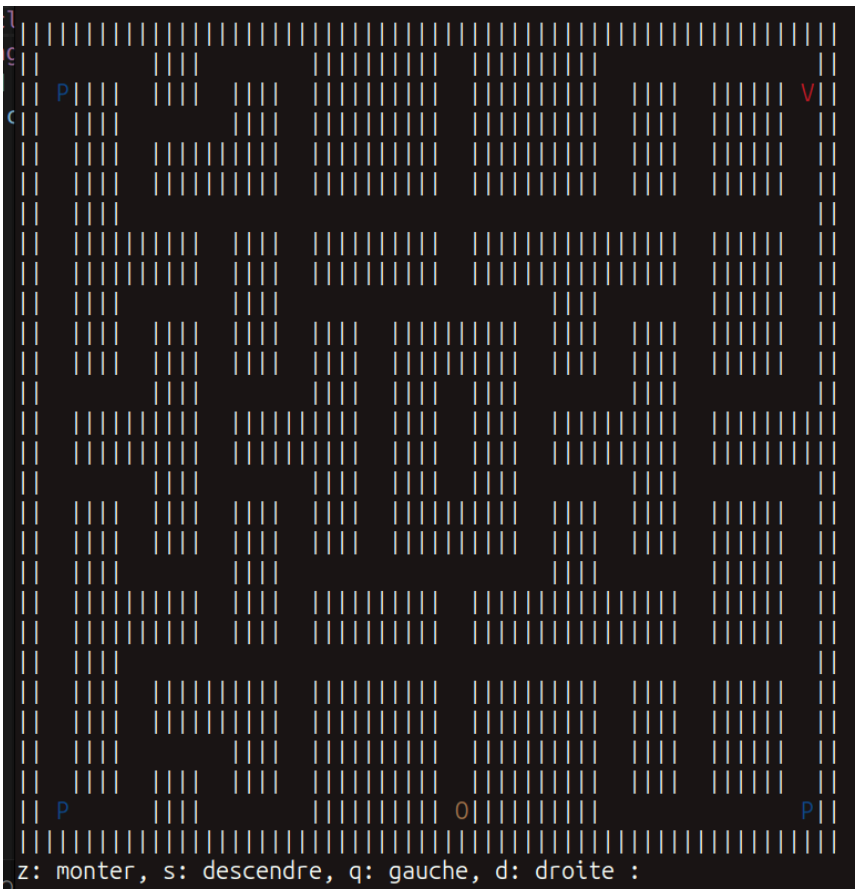
Niveaux de jeu

- Ajout de plusieurs niveaux avec une difficulté croissante (augmentation du nombre de voleurs et complexité des déplacements).

Points spécifiques

- **Murs** : Ajout d'obstacles statiques pour complexifier les déplacements..

Terminal (Rendu Final)



IV. Documentation Générée

Présentation des outils utilisés

Nous avons utilisé **Doxygen** pour générer la documentation du code source. Cet outil a permis de créer des fichiers HTML et PDF clairs et bien structurés.

V. Résultats et Tests

Résultats obtenus après implémentation

Le jeu est fonctionnel avec toutes les fonctionnalités prévues. Les tests montrent une bonne fluidité et une expérience utilisateur intuitive.

Tests effectués

Fonctionnels

- Validation des déplacements du joueur et des bots.
- Test des interactions (collision joueur-voleurs, téléportation, etc.).

Limites

- Gestion des cas extrêmes, comme un grand nombre de voleurs ou une petite taille de carte.
- Tests sur différentes configurations pour s'assurer de la compatibilité.

VI. Répartition des Points en Équipe

Même nombre de points chacun

VII. Conclusion et Perspectives

Bilan de la SAE

Ce projet nous a permis de renforcer nos compétences en C++, développement collaboratif et gestion de projet. Nous avons réussi à développer un jeu complet respectant les objectifs fixés.

Améliorations potentielles futures

- Ajout d'un mode multijoueur pour enrichir l'expérience de jeu.
- Création de niveaux générés aléatoirement pour une rejouabilité accrue.
- Amélioration de l'intelligence artificielle des bots pour les rendre plus imprévisibles.
- Implement sur Mingl