Tran Sylvain
Guerroui Lina

Rapport Projet C: Jeu d'Infiltration

1) Documentation technique
2) Documentation utilisateur

1) Documentation technique

Nous avons crées au total **13** modules :
-Direction : pour gérer les 4 directions cardinales avec un type énuméré enum.
-Point : pour gérer un point (x, y). Nous avons utilisé des doubles car c'est plus précis que des int.
-Disc : pour représenter un agent (Joueur ou Gardien) qui est un disque.
-Player : pour représenter un joueur et ses compétences magiques (Invisibilité et Accélération).
-Guard : pour représenter un gardien et ses 2 modes (normal et panique).
-Grid : pour représenter la grille de cases de taille 60×45.
-Terrain : pour gérer la subdivision de la salle en plusieurs compartiments (en créant les murs et ouvertures) + le placement de toutes les entités et objets du jeu : les reliques, le joueur, les gardiens et le mana.
-Collision : pour gérer la collision entre les agents et les murs mais également entre les gardiens eux-mêmes pour éviter qu'ils ne se rentrent dedans dans la salle.
-Detection : pour gérer la détection du joueur par les gardiens mais également la détection des reliques récupérées.
-Ranking: pour gérer le score du joueur à la fin de la partie et le tableau de classements (par meilleurs temps et par meilleure consommation de mana).
-GameEngine : c'est le moteur du jeu.
-Graphic : pour gérer l'affichage graphique.
-Main : contient la boucle maîtresse faisant tourner le jeu.

Quelques Précisions :
Nous allons apportées quelques précisions à notre code:

1)
On a choisie de représenter les reliques comme des «InfoCase» dans le module Grid parce qu'on a trouvé que c'était beaucoup plus simple de gérer le cas des reliques comme ceci plutôt de créer par exemple un struct Relic avec pour champ un Point et un enum Status.

En fait, on était parti sur ça au début.
Mais en réfléchissant un peu :
-pour les placer dans la grille.
-gérer le placement des manas → on ne doit pas les poser sur une relique
-la récupération par le joueur.
-détecter la fin du jeu.

Tout cela est beaucoup plus simple en utilisant relique comme «InfoCase»

2)
Dans le module Collision, pour les formules, on a choisie de placer le test simple (celui où la case en face de l'agent est un mur et donc la coordonnée en x ou en y ne doit juste pas dépasser le milieu de la case) à la toute fin des fonctions « travel_limit_in_x» et «travel_limit_in_y»
car quand on va vers le bas ou vers la droite, ce qu'on veux c'est le min() des 3 limites de placement calculables et quand on vas vers le haut ou vers la gauche c'est le min() qu'on veut.

Et en plaçant le test simple à la fin, on n'as pas besoin de Macro Max() ou Min(), on est sur d'avoir le min ou le max.

Nous avions eu aussi un soucis quand on était collé à un mur en bas et qu'on essayait d'aller à gauche, cela bloquait. Pareil quand le joueur était collé à un mur à droite et qu'on essayait de le faire monter en haut.
En mettant des > au lieu des ≥ et < au lieu ≤ dans les formules, ça a résolu le problème

pour éviter le chevauchement avec le point $(\lfloor x_0 \rfloor, \lfloor y_0 \rfloor + 1)$.

c) Si la case en bas à droite $(\lfloor x_0 \rfloor + 1, \lfloor y_0 \rfloor + 1)$ est une case de mur, et que $x_0 \geq \lfloor x_0 \rfloor + 1/2$, alors la coordonée y du centre du disque ne peut pas aller au-delà de

3)
Nous avons changé la structure de données pour les compétences du joueur.
Au début, on avait un enum avec 3 champs NONE, ACCELERATION, INVISIBILITY
mais on a vu qu'on pouvait pas permettre au joueur d'utiliser les 2 compétences à la fois avec un enum et donc ce qu'on a fait, c'est qu'on a mis à la place un type structuré avec deux champs cette fois-ci qui sont des booléens.
On a crée les constantes USED et UNUSED pour indiquer qu'une compétence est active ou non

4)
Pour calibrer la vitesse du joueur et des gardiens, nous avons choisi comme valeur de «v»: 4.4.
On trouve que c'est assez fluide avec cette valeur.
Pour calibrer les autres valeurs de vitesses nous avons gardé celles proposées dans l'énoncé qui sont très bien.

5)
Pour le déplacement du joueur dans la salle, on a décidé de découper la direction en deux 2 parties (et donc pas en 4 : Haut, Bas, Gauche et Droite) :
On a une direction verticale et une direction horizontale. Ce qui veux dire qu'on ne compare que la direction Haut à Bas et que la direction Gauche à Droite (voir la fonction same_direction() dans Player).
Grâce à cela, on permet au personnage de se déplacer en diagonale de manière fluide.
Pour déplacer le joueur en diagonale, on peut appuyer à la fois sur une touche qui gère la direction horizontal et à la fin sur une touche qui gère la direction verticale : par exemple « z » + « d » en même temps pour faire la diagonale droite.

2) Documentation utilisateur

1. Menu du jeu

→ Cliquer sur PLAY pour commencer la partie
→ Cliquer sur QUIT pour quitter le jeu

2. Déroulement d'une partie

Légende des différentes entités/objets :

: joueur

: joueur utilisant sa compétence invisibilité

: joueur utilisant sa compétence accélération

x5 : gardien

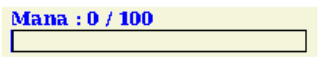
x3 : relique

: traces de mana sur une tuile

: mur

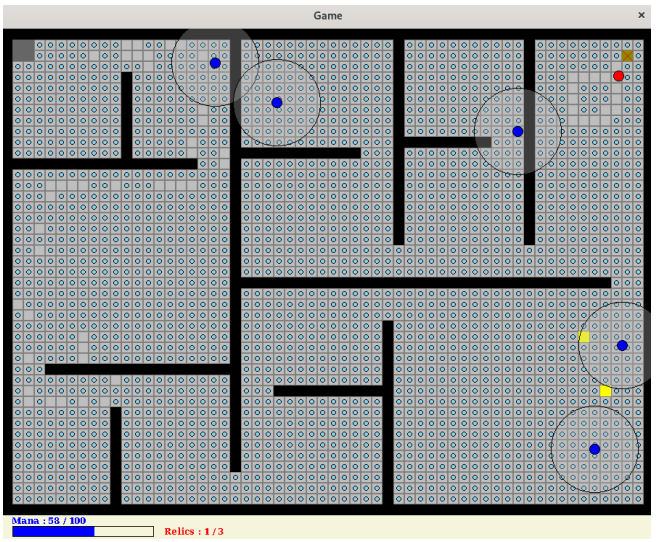
: tuile vide

: entrée de la salle



: barre de mana
-> varie et représente la quantité de mana absorbée par le joueur au cours de la partie

: indicateur sur le nombre de relique déjà récupérée

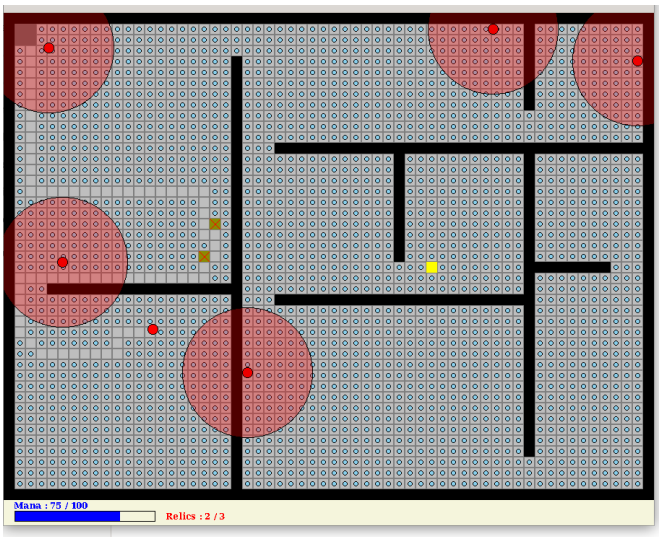


→ Utiliser les touches **Z, Q, S, D** pour faire bouger le joueur
(possibilité de se déplacer en diagonale : Z+Q, Z+D, S+Q, S+D)

Vos compétences magiques :

→ Invisibilité : utiliser la touche **espace** pour vous rendre invisible
→ Accélération : utiliser l'une des touches **shift (droit ou gauche)** pour accélérer

L'usage des compétences consomme du mana !



→ dès qu'un gardien détecte qu'une relique a disparu, tous les gardiens entrent en mode panique : ils deviennent alors plus rapides avec une plus grande zone de détection pendant 30 secondes.

En mode panique, les gardiens deviennent rouges et leur zone de détection aussi

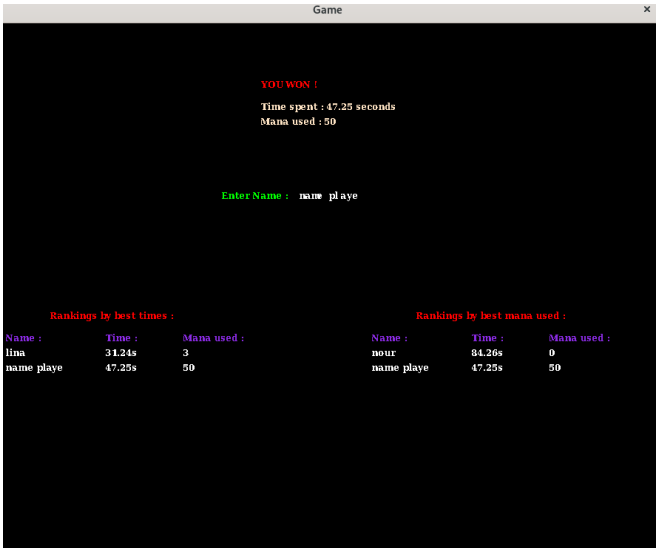
3. Fin de partie

En cas de défaite :

Si un gardien vous a détecté, vous avez perdu !

→ Affichage de votre temps de jeu + la quantité de mana utilisée
→ Affichage des classements par meilleur temps et meilleure consommation de mana

En cas de victoire :



Si vous arrivez à récupérer toutes les reliques et à revenir à l'entrée de la salle sans être détecté par les gardiens, vous avez gagné !

→ Affichage de votre temps de jeu + la quantité de mana utilisée
→ Vous pouvez entrer votre nom dans le tableau des classements
→ Affichage des classements par meilleur temps et meilleure consommation de mana (avec peut-être votre nom si vous faites partie des meilleurs)