

Jeu « Banquise »

Présentation du jeu

Le jeu se fera tour par tour.

Un tableau de $N \times N$ case représente une banquise (ou l'absence de banquise).

1 à 4 joueurs se déplacent sur celle-ci (chacun leur tour).

Des « glaçons » sont posés un peu partout (sur la banquise).

Le ou les joueurs doivent aller d'un point A à un point B de la banquise, en se déplaçant sans tomber dans l'eau.

Si aucun chemin n'existe, ils peuvent pousser (donner un coup dans) un glaçon. Celui-ci se déplace (vecteur) jusqu'à tomber dans l'eau ou être arrêté par un obstacle. Un glaçon qui tombe à l'eau agrandit la banquise.

Un glaçon en déplacement peut « tuer » un joueur s'il le touche dans sa course.

Réchauffement climatique :

- Chaque glaçon peut fondre (% de risque, à chaque tour)
- Chaque élément de la banquise peut fondre (% de risque) si la case touche déjà de l'eau (bord de la banquise)

Les déplacements des joueurs se feront via les touches Z,S,Q,D pour haut, bas, gauche et droite.

Une case de la banquise est définie par :

- Est de la glace ou de l'eau (absence de banquise).
- Est occupé par un joueur (pointeur sur) ou non (null).
- Contient un objet (posé sur la banquise, donc allocation mémoire) ou non (null).
- Est le point de départ ou d'arrivée.

Un objet peut être :

- Un glaçon (posé ou en déplacement sur la banquise donc)
- Un rocher (stoppe un glaçon en déplacement)
- Un ressort (inverse le sens du déplacement d'un glaçon)
- Un marteau (occupant 2 cases, 1^e case : axe de rotation, 2^e case : tête du marteau) qui pivote quand un glaçon en mouvement heurte la tête du marteau, ce qui arrête le glaçon, celui-ci fait un $\frac{3}{4}$ de tour sur lui-même (autour de la 1^{ere} case qui sert d'axe) et tape le glaçon à son tour entraînant au final un changement de direction de celui-ci (modif vecteur) de 90°. Une animation sera optionnelle (en gérant la rotation du marteau sur son axe de 90° par tour, avec 3 tours « d'attente » pour le glaçon)
- Un piège (fixe le joueur pendant un tour)
- Autre... (?)

Un glaçon est défini par :

- Sa position
- Un vecteur (dx et dy à 0 si immobile)

Un glaçon occupe une seule case du jeu.

Un joueur est défini par :

- Son nom
- Une couleur
- Un id
- Une position (x,y)
- Un vecteur
- Un score (distance parcourue ? nb de glaçons poussés ? de joueurs « tués » à coup de glaçon ?)

Quelques fonctions (liste non exhaustive) :

- Une fonction choisissant aléatoirement les points de départ et d'arrivée, parmi les cases de la banquise (éviter une case d'eau).
- Une fonction initialisant la banquise, etc
- Une fonction positionnant les joueurs au plus près du point du point A (parcours du tableau) en début de partie
- Une fonction appliquant le réchauffement climatique
- Une fonction détectant si la partie peut être gagnée (ie, qu'il reste suffisamment de glaçons pour agrandir la banquise si pas de chemin direct)
- Une fonction récursive détectant si un chemin direct existe

Vous allez devoir créer beaucoup d'autres fonctions, à vous de voir ce dont vous aurez besoin pour gérer votre jeu.

➔ Calcul de la complexité de chaque fonction (à mettre en commentaire dans votre code).

Affichage :

L'idée est que chaque objet du jeu soit « imprimé » (ajouté) sur une structure en mémoire représentant le jeu. Par défaut, le tableau représentant la banquise convient parfaitement.

Ainsi vous avez à créer une fonction qui affiche cette représentation du jeu, permettant d'afficher à chaque tour son « état » à l'écran.

La création de chaque nouvel élément (rocher, marteau...) du jeu entraînant une maj de cette fonction d'affichage, afin que celui-ci puisse apparaître.

Donc au début, la fonction « afficheJeu » ne saura afficher qu'une banquise sans joueur ni objet, puis au fur et à mesure, elle permettra d'afficher tous les éléments que vous allez ajouter dans votre jeu, dans l'ordre d'implémentation de ceux-ci.

L'affichage se fera en mode texte via des printf dans un premier temps.

Démarche de développement :

1. Créer un type nécessaire au jeu (la banque en premier lieu)
2. Créer la/les fonctions utile(s) à sa manipulation (exemple initBanque, FonteBanque...)
3. Créer une fonction permettant l'affichage de cet objet
4. Tester son fonctionnement
5. Passer à l'objet suivant du jeu, etc.

Vous devez créer un .h et un .c par type d'élément.

Votre main.c réalisant des <include> de tous les .h.

Evaluation :

Vous allez devoir travailler en binôme (sauf nombre impair d'étudiants ou cas exceptionnel).

Votre note sera construite à partir de :

- Lisibilité du code
- Efficacité du code
- Indentation
- Commentaires décrivant le rôle de chaque fonction
- Auteur de la fonction en commentaire (pour savoir qui fait quoi)

Votre projet devra être déposé sous forme d'archive « **nom1_nom2.zip** », et déposé via les deux comptes des deux étudiants constituant le groupe, avec le même nom d'archive.

A la correction, le correcteur va trier toutes les archives par nom et je dois donc les trouver en double (sauf pour les monômes –cas exceptionnel-).

Date de dépôt non encore définie. Ce sera après les vacances de février.