

# Travail 2 — Serpents et échelles

---

## 1 Instructions générales

- La date de remise est le 9 novembre à 23 h 59.
- Le travail s'effectue individuellement ou en équipe de 2.
- Les modalités de remise sont décrites à la fin de ce document.
- La grille de correction se trouve à la fin de ce document.

## 2 Énoncé

Pour ce travail, vous devez créer un jeu de « Serpents et échelles » en utilisant la SFML. La fenêtre d'affichage doit avoir 800 x 800 et être composée de 18 cases de haut par 20 cases de large (pour un total de 360 cases).

Au démarrage du programme, avant que la fenêtre graphique n'apparaisse

- Une question dans la console demande à l'utilisateur d'entrer le nombre de joueurs. Ce nombre doit être entre 2 et 6.
- Ensuite, une question demande le prénom de chacun des joueurs

Chaque fois que c'est le tour d'un joueur à jouer, un message l'indique au bas de la fenêtre graphique (le message est affiché en utilisant la couleur associée au joueur).

- Quand le joueur appuie sur la barre d'espace, une valeur entre 1 et 6 est choisie au hasard.
- Un dessin d'un dé montrant la bonne valeur remplace le message
- Le pignon du joueur avance, case par case, du nombre de cases approprié
- S'il s'arrête sur la tête d'un serpent, une musique appropriée se fait entendre et le jeton du joueur se retrouve en bas (après un déplacement animé)
- S'il s'arrête sur le pied d'une échelle, une musique appropriée se fait entendre et le jeton du joueur se retrouve en haut (après un déplacement animé)
- S'il atteint la dernière case, un message indique qu'il gagne. Encore une fois, le message est affiché en utilisant la couleur associée au joueur. Appuyer sur « N » permet alors de débiter une nouvelle partie.

Voici quelques instructions

- Les couleurs utilisées pour les joueurs sont bleu, jaune, orange, rose, rouge et vert
- Les jetons des joueurs sont de simple rond
- Le jeton d'un même joueur est toujours placé au même endroit relativement à la case où il se trouve
- Les serpents et les échelles sont dessinés en ligne droite

- Chaque case est numérotée de 1 à 360 et ce numéro est inscrit en blanc
- La même image est utilisée pour tracer toutes les cases. C'est un carré vert foncé avec une bordure noir

Instruction particulière aux structures de données.

- Une liste simplement chaînée de structure contient les informations sur les serpents
  - Case de la tête
  - Case de la queue
  - sf::Sprite utilisé pour le dessiner
- Une liste simplement chaînée de structure contient les informations sur les échelles
  - Case du haut
  - Case du bas
  - sf::Sprite utilisé pour le dessiner
- Une liste circulaire contient les informations sur les joueurs
  - Case actuelle
  - Couleur
  - Prénom
  - ...
- Un tableau à deux dimensions contient les informations sur les cases
  - sf::Sprite utilisé pour dessiner la case
  - sf::Text utilisé pour dessiner le numéro

Votre programme doit absolument définir et utiliser les fonctions suivantes

#	Déclaration
1	<code>unsigned int NumeroCaseAColonne( unsigned int aNumeroCase );</code>
2	<code>unsigned int NumeroCaseALigne ( unsigned int aNumeroCase );</code>

Votre programme doit absolument définir un module nommé « Liste\_Chemin » qui sera utilisé pour la liste des serpents et la liste des échelles. Ce module doit exporter les fonctions suivantes.

#	Déclaration
1	<code>void Liste_Chemin_Afficher( Liste_Chemin * aListe, sf::RenderWindow * aWindow );</code>
2	<code>void Liste_Chemin_Ajouter ( Liste_Chemin * aListe, Chemin * aChemin );</code>
3	<code>void Liste_Chemin_Liberer ( Liste_Chemin * aList );</code>
4	<code>Chemin * Liste_Chemin_Chchercher_Bas ( Liste_Chemin * aList, unsigned int aCase );</code>
5	<code>Chemin * Liste_Chemin_Chchercher_Haut( Liste_Chemin * aList, unsigned int aCase );</code>

Votre programme doit absolument définir un module nommé « Liste\_Joueur ». Ce module doit exporter les fonctions suivantes.

#	Déclaration
1	<code>void Liste_Joueur_Afficher( Liste_Joueur * aListe, sf::RenderWindow * aWindow );</code>
2	<code>void Liste_Joueur_Ajouter ( Liste_Joueur * aListe, Joueur * aJoueur );</code>
3	<code>void Liste_Joueur_Liberer ( Liste_Joueur * aListe );</code>
4	<code>Joueur * Liste_Joueur_Premier( Liste_Joueur * aListe );</code>

### 3 Modalité de remise

Le travail doit être remis via LÉA avant le 9 novembre à 23 h 59.

Les fichiers sources et l'exécutable doivent être remis dans un dossier compressé en format .zip.

Par la suite chaque étudiant doit me rencontrer individuellement pour m'expliquer brièvement comment le programme utilise les structures de données.

