

Projet final

Conception d'un jeu vidéo avec la librairie SFML

Présentation

En équipe de trois ou quatre, vous devez programmer un jeu vidéo dont le sujet est libre à vous (le sujet devra cependant être approuvé par l'enseignant) tout en respectant les critères suivants :

- Vous devez obligatoirement inclure votre propre **liste** dans votre jeu vidéo et en faire une utilisation pertinente (pas simplement l'inclure sans vraiment l'utiliser ou pas seulement l'inclure sans que son utilisation ne soit justifiée dans le contexte).
- Vous devez inclure au moins deux autres de vos structures de données programmées dans le cours :
 - Vecteur
 - Matrice dynamique (classe *Map*)
 - Pile
 - File

Ces deux autres structures de données doivent être différentes l'une de l'autre (pas deux fois le vecteur, par exemple) et, ici encore, leur utilisation doit être pertinente dans le contexte de votre jeu vidéo.

- Votre jeu vidéo doit intégrer les interfaces graphiques suivantes :
 - Menu pour paramétrer le jeu avec au moins trois réglages différents (par exemple, réglage du volume pour le son, réglage de la difficulté du jeu, réglage de la vitesse de progression, etc.)
 - Interface du jeu lui-même (interface à travers laquelle une partie se déroule)
 - Menu des statistiques s'affichant en fin de partie avec au moins trois statistiques (par exemple : joueur ayant obtenu le plus haut score, temps de jeu total, nombre de trésors récupérés, nombre d'ennemis éliminés, etc.)
- Votre jeu doit être développé **au moyen de la librairie graphique SFML**.

Voici quelques exemples de jeu dont vous pouvez vous inspirer. Libre à vous de choisir un jeu vidéo à coder parmi cette liste ou de proposer un autre sujet de jeu. **Votre sujet doit cependant être approuvé par l'enseignant.**

- Rush Hour
- Tetris
- Sky Chasers
- Arkanoid
- Space Invaders
- Bomberman
- Ballz
- Crossy Road
- Data Wing
- Etc.

Vous devez partir du projet *Visual Studio* fourni par l'enseignant, qui intègre déjà la librairie SFML.

En plus de coder votre jeu, votre équipe devra produire une vidéo de présentation de celui-ci en respectant les critères suivants :

- La durée de la vidéo doit se situer entre 15 minutes et 20 minutes.
- Vous devez introduire votre jeu (nom et but du jeu). Il vous faut aussi expliquer le fonctionnement de votre jeu (tenez pour acquis que le public cible ne sait pas comment jouer à votre jeu et que vous devez alors lui faire comprendre).
- Dans votre vidéo, vous devez procéder à une démonstration d'une partie typique de votre jeu. Si une partie peut être longue et prendre trop de temps pour la durée de la vidéo, faites une démonstration réduite (ciblez les actions les plus pertinentes qu'un joueur peut faire dans le déroulement d'une partie). Votre démonstration doit inclure les cas suivants :
 - Victoire du joueur
 - Défaite du joueur
 - Quelques actions permises et interdites selon le contexte
 - Quelques cas limites (gestion des collisions, dépassement de la grille de jeu, changement de tableau dans le jeu, attaque qui atteint sa cible et attaque qui ne l'atteint pas, etc.)
- À la fin de la démonstration d'une partie typique, vous devez montrer, dans votre vidéo, le menu de statistiques (et, s'il y a lieu, expliquer les statistiques qui ne seraient pas évidentes à comprendre).
- Vous devez aussi montrer le fonctionnement des différents réglages de votre jeu (les options que vous avez implémentées dans un menu pour qu'un joueur puisse paramétrer le jeu).
- La vidéo doit explicitement montrer où sont utilisées, dans votre code, les structures de données demandées (la liste et les deux autres structures parmi le vecteur, la matrice dynamique, la pile ou la file).
- En plus de les montrer dans votre vidéo, vous devez expliquer à quoi servent les structures de données utilisées (Quelle est l'utilité de chacune dans la cadre du jeu ? Pourquoi avoir utilisé telle structure plutôt qu'une autre ? etc.)
- En guise de conclusion dans votre vidéo, vous devez indiquer les difficultés rencontrées et, s'il y a lieu, les bogues qui n'ont pas été réglés. **Soyez honnête, car il n'y a pas de pénalité à présenter des bogues à moins qu'ils contreviennent grandement au déroulement du jeu** (par exemple : impossible de gagner, jeu qui plante complètement en cours de route, etc.)

Remise

Votre projet doit être remis au plus tard à 23h59, le dernier jour de la session, via *GitHub Classroom* (en faisant un « push » dans la branche « main » de votre dépôt automatiquement généré par *GitHub Classroom*). Cette remise doit inclure :

- Le projet *Visual Studio* complet avec toutes les ressources nécessaires (fichiers, images, etc.), la librairie SFML installée et les dépendances requises de sorte que l'enseignant n'ait pas à reconfigurer votre projet pour qu'il puisse compiler et s'exécuter sur son ordinateur.
- Un lien, dans le fichier *README.md* de votre dépôt, vers la vidéo présentant votre jeu vidéo et l'intégration des structures de données dans votre code (par exemple, un lien YouTube ou OneDrive).

Critères d'évaluation

Votre code sous *Visual Studio* de même que votre vidéo seront évalués selon les critères suivants :

- Logique de programmation / clarté et simplicité du code (20%)
- Choix et utilisation judicieuse des structures de données (20%)
- Respect de la demande (20%)
- Fonctionnalité du programme (20%)
- Qualité de la présentation graphique (10%)
 - Inclut la qualité de la langue écrite dans les interfaces du jeu
- Présence des éléments demandés dans la vidéo (10%)

*** Un programme qui ne compile pas ne pourra obtenir une note de plus de 50%. ***