

Classement de courses

SECTION :	Conception et Réalisation de Jeux Vidéo
ENSEIGNANTE :	Selene Tonon
MATIÈRE :	Projets de développement de jeux
DATE DE L'ÉPREUVE :	10 mars 2021
DUREE DE L'ÉPREUVE :	Deux heures

ÉNONCÉ DU SUJET : *On souhaite programmer un système de classement pour un jeu de courses.*

A/ Chaque course possède un identifiant, et peut avoir un certain nombre de scores enregistrés dans le jeu (maximum 3). Chaque joueur ou joueuse ne conserve qu'un seul score par course (le meilleur ; si un meilleur score arrive par la suite, l'ancien est donc écrasé ; s'il n'est pas meilleur que l'ancien, il est ignoré). Il existe 5 courses dans le jeu (c'est un premier prototype). Chaque score est enregistré sous le pseudo d'un joueur ou d'une joueuse. On ne fera toutefois pas de classe pour le/la représenter.

Les règles du jeu sont très orientées convivialité, dans une ambiance bienveillante, et il n'est pas vraiment possible d'avoir un mauvais temps. Le pire score qu'on puisse avoir est de 999 secondes. On estime qu'en termes de programmation, un score de -1 signifie qu'en réalité la course n'a pas encore été essayée par ce joueur ou cette joueuse et que la valeur n'a donc pas encore été modifiée. Il est possible de calculer automatiquement pour une même course la moyenne des temps enregistrés, d'afficher le pseudo de la personne qui a obtenu le meilleur temps. On peut évidemment afficher les scores (du meilleur au moins bon), ou ajouter un score. Il est en revanche impossible de supprimer un score (si ce n'est en faisant mieux : le moins bon sera alors écrasé).

1/ Modéliser en UML la classe "Course" décrite ci-dessus.

2/ En vous servant de la modélisation, déclarez la classe dans un fichier Course.h.

3/ En vous servant de votre déclaration de la classe, implémentez ses méthodes dans Course.cpp.

4/ Servez-vous de la classe obtenue dans une fonction principale créant des objets de la classe Course et effectuant différents tests, afin de vérifier la pertinence de l'implémentation.

B/ En réalité, des objets de classe Course peuvent intégrer un Championnat. Un Championnat possède un nombre déterminé de courses (quatre) ; toutes seront jouées par les mêmes personnes et donc chacune gardera le même pseudo. On peut visualiser le meilleur temps pour une course, le temps moyen pour chaque course, le meilleur temps toute course confondue, le/la joueur/joueuse possédant le meilleur temps pour chaque course et le meilleur temps total.

1/ Modéliser en UML la classe Championnat, et ses interactions avec la classe Course, ainsi que la modification de cette dernière tenant compte des consignes mentionnées ci-dessus.

2/ Déclarez et implémentez la classe Championnat, et modifiez la classe Course.

3/ Testez les classes Championnat et Course grâce à un programme principal prévu à cet effet.

C/ Comment pourrait-on faire pour rejouer les courses d'un championnat de façon indépendante et tout de même enregistrer ses scores ?

SUPPORTS AUTORISÉS :

- [Documentation de C++](#)
- Supports & notes de cours

RENDU :

Vous vous servirez de git pour gérer les versions du projet. Le jeu doit être téléchargeable à partir de votre compte github. Vous devez pour cela veiller à ce que le dépôt soit public. La nomenclature du dépôt sera de type "CPP_COURSES_[NOM]" (exemple : le dépôt de votre enseignante se nommerait CPP_COURSES_TONON).

Les modélisations UML seront jointes au rendu dans le même dépôt (si numérisées), ou à rendre sur papier libre en ayant bien pris soin d'y noter son nom et prénom.

Il en est de même de la réponse à la partie C.

BARÈMES :

Critères d'évaluation	Notation
Exercice A	/12
Exercice B	/6
Exercice C	/2
Note globale	/20