



Rapport projet C++ Simulateur PARIS 2024

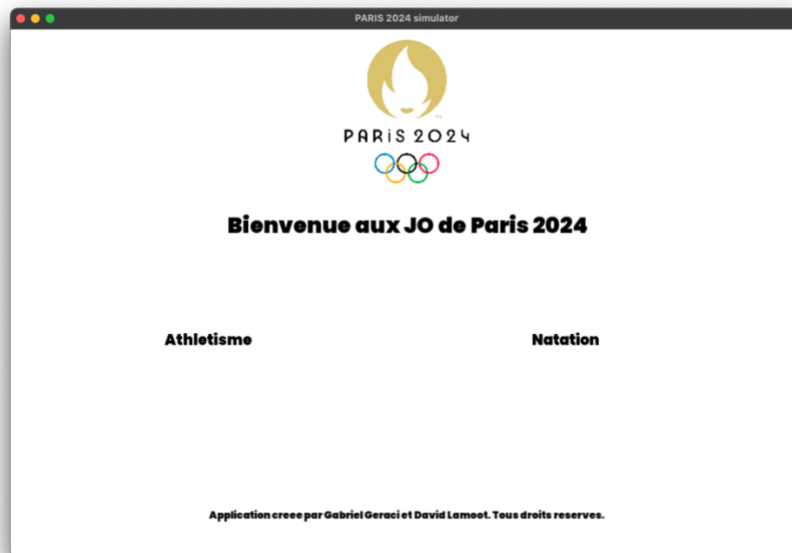
GERACI Gabriel
LAMOOT David

Table des matières

Description du projet :	3
UML :	5
Utilisation des contraintes	5
Fierté du code :	6
Exécution du code	6
Installation de la SFML	6
Windows :	6
Linux :	7
MacOS :	7

Description du projet :

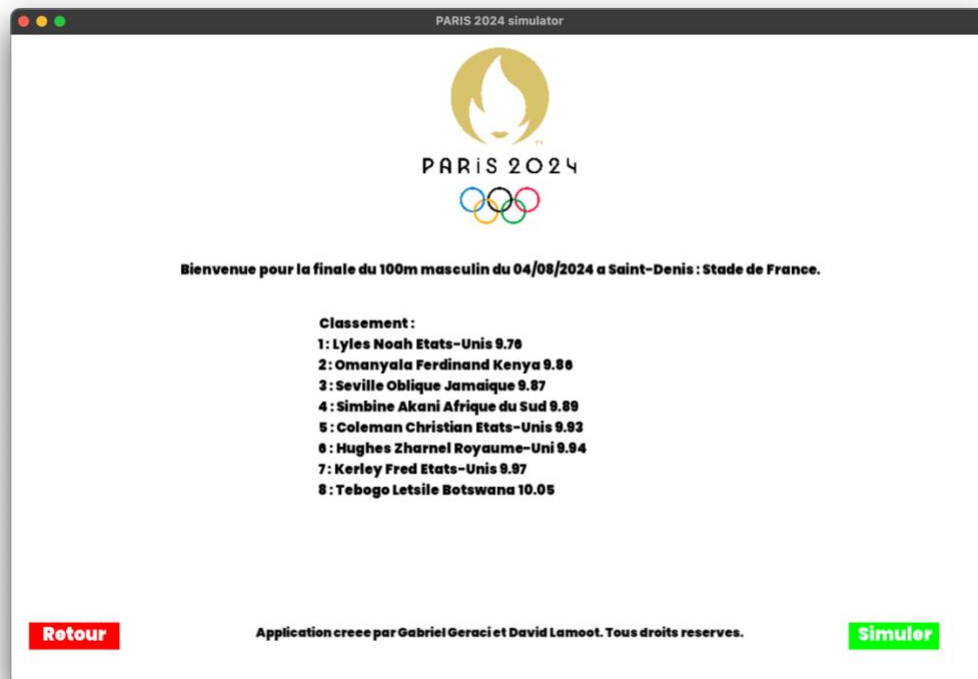
Au travers de ce projet C++ sur le thème des jeux olympiques, nous avons décidé de réaliser un simulateur d'épreuves des jeux olympiques de Paris 2024. Celui-ci, codé en C++, utilise la SFML afin d'offrir à l'utilisateur une expérience simple et agréable.



Notre application à un fonctionnement assez simple : pour se déplacer au sein de celle-ci, il faut se déplacer en cliquant sur les noms. Par exemple, simulons une épreuve du 100m masculin : il faut cliquer sur athlétisme, sprint puis 100m masculin. Après avoir suivi ces différentes étapes, nous arrivons ici :



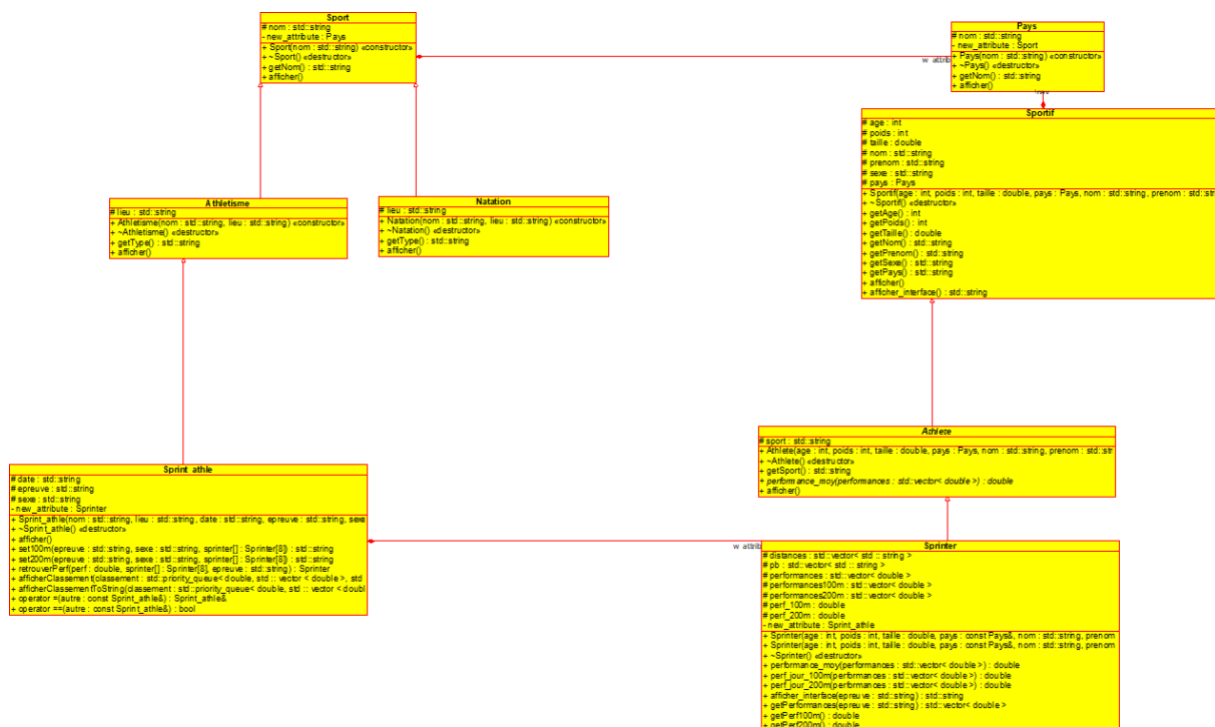
Afin d'obtenir les résultats, il suffit de cliquer sur le bouton « Simuler » (le bouton « Retour » sert quant à lui à revenir en arrière dans l'arborescence de l'application).



Les résultats de la course sont obtenus au travers de différentes fonctions dont on expliquera le fonctionnement plus en détail dans la suite de ce document.

Pour le moment, seules les épreuves de sprint du 100m et 200m (masculin et féminin) sont simulable.

UML :



Utilisation des contraintes

Dans le cadre de ce projet, nous avons la contrainte d'utiliser 8 classes réparties sur 3 niveaux de hiérarchie. Cette contrainte majeure nous a permis de réfléchir sur comment réaliser au mieux notre code afin que celui-ci soit aisément améliorable par l'ajout d'épreuves olympiques afin de rendre notre simulateur complet. Nous savions qu'implémenter toutes les épreuves des jeux olympiques en rendant la simulation réaliste allait être une tâche complexe, par conséquent nous avons orienté au maximum notre conception d'application afin qu'il soit simple d'ajouter des épreuves en ajoutant simplement des classes.

Les contraintes nous ont également poussé à utiliser 2 conteneurs de la STL. Dans le cadre de notre projet, nous avons utilisé les vectors et les priority queue. L'avantage de l'utilisation des vectors nous permettait d'avoir des tableaux dynamiques. Dans le cas des priority queue, cela nous permettait de trier un tableau de performance dans l'ordre croissant afin de réaliser le classement des différentes épreuves.

Fierté du code :

Ce projet nous a procuré de la fierté de par la conception d'une interface graphique mais également par la volonté d'avoir des résultats réalistes. Cette partie n'était pas évidente à obtenir : pour ce faire, nous avons décidé dans un premier temps de regarder le classement mondial du sprint puis nous avons choisis les 8 meilleurs athlètes de leur discipline en supposant que ceux-ci disputeraient la finale des jeux olympiques. Par la suite, nous avons établi un code qui permet de simuler leur éventuelle performance lors des jeux : l'idée derrière cela était de créer une base de données regroupant leur 5 meilleures performances sur l'année 2023 puis, avec ces données, réaliser une moyenne de ces performances. Avec cette moyenne obtenue, nous générons ensuite une valeur aléatoire encadrant cette moyenne (encadrement plus ou moins important selon la discipline) et cela nous permet d'obtenir la performance potentielle de chaque athlète et par conséquent d'établir un classement entre eux. Pour ce faire, nous avons mis en place une priority queue un peu particulière car c'est le nombre le plus petit, correspondant au vainqueur car cela équivaut au temps pris pour courir la distance de l'épreuve le plus faible, qui sort en premier puis ainsi de suite afin d'obtenir le classement de notre finale olympique. Cette partie représente une certaine fierté car nous avons longtemps réfléchi comment obtenir une simulation réaliste et nous sommes très heureux d'y être parvenu.

Exécution du code

Afin d'exécuter le code correctement, il faudra plusieurs choses :

- Dans le fichier main, mettre le chemin correspondant au dossier Poppins et au logo des jeux olympiques respectivement aux lignes 531 et 27.
- Installer la SFML sur votre ordinateur, cette étape peut être différente selon votre environnement.
- Si des problèmes persistent, un exécutable est présent dans le dossier afin que vous puissiez tester l'application

Installation de la SFML

Pour installer la SFML, il existe différentes façons de procéder selon votre environnement. Ici, nous allons voir comment réaliser cela dans le cas où vous êtes sur Windows, linux ou MacOS.

Windows :

Dans un premier temps il vous faudra installer la version adéquate depuis le [site officiel de la SFML](#). Dans un second temps, il faudra configurer votre environnement en ajoutant les fichiers d'en-tête correspondant aux différentes bibliothèques de la SFML.

Linux :

Ici, vous pouvez installer la SFML directement depuis votre terminal en utilisant la commande : « `sudo apt-get install libsfml-dev` ». Ensuite veuillez configurer votre projet afin que celui-ci prenne en compte les différentes bibliothèques de la SFML.

MacOs :

Ici, de même, vous pouvez installer la SFML sur votre ordinateur directement depuis votre terminal avec la commande « `brew install sfml` ». Après avoir réalisé cela, configurez votre projet pour inclure les en-têtes SFML et lier les bibliothèques SFML.