PROJET C++
OLYMPATLAS

Karim QORAR

Robert UT

TABLE DES MATIERES

INTRODUCTION

Les Jeux Olympiques, symbole universel de compétition sportive, d'excellence et d'unité, ont toujours captivé l'attention du monde entier. Dans le cadre de notre projet en langage C++, nous avons entrepris de développer une application avec interface graphique sur le thème des Jeux Olympiques. Notre objectif était de créer une application sous la forme d'une carte interactive, offrant aux utilisateurs une expérience immersive et informative autour de cet événement.

Dans ce compte rendu, nous détaillerons le processus de conception, les défis rencontrés et les fonctionnalités mises en œuvre pour donner vie à cette carte interactive. Nous mettrons par ailleurs en valeur les contraintes imposées par le projet, les bibliothèques utilisées et le diagramme UML de l'application. L'application est disponible sur le dépôt git suivant : https://github.com/utrobert2/olympatlas

1. DESCRIPTION DE L'APPLICATION

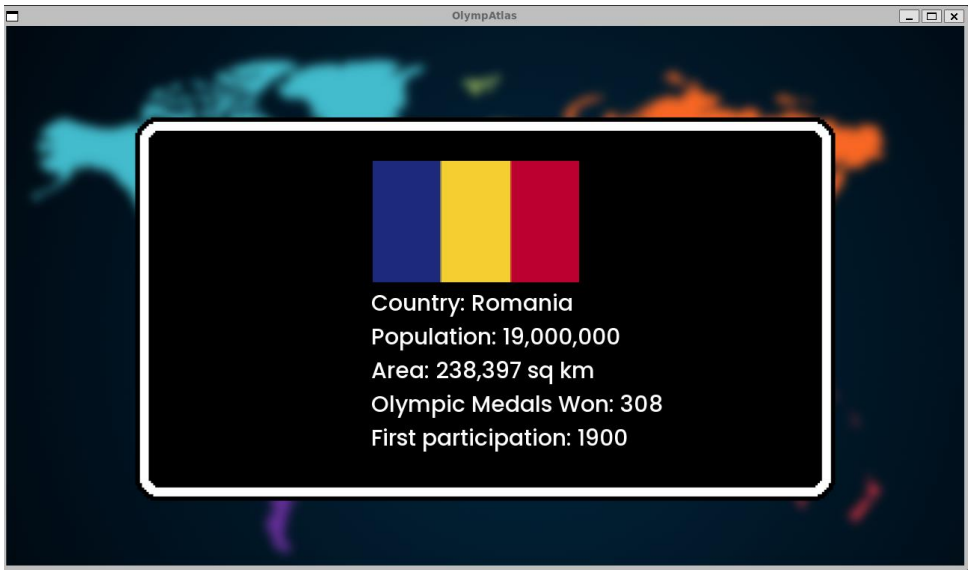
Lors du lancement de l'application, un menu principal accueille l'utilisateur.

En pressant la touche *S*, ce dernier débute son exploration et peut sélectionner le continent de son choix sur la carte.

Cette interaction permet de zoomer sur la région sélectionnée, offrant une vue détaillée des pays du continent.

Il est ensuite possible de sélectionner un pays en cliquant sur son marqueur, et les informations suivantes seront affichées :

- Drapeau
- Population
- Superficie
- Médailles olympiques aux JO d'été gagnées



2. CONTRAINTES

2.1 CLASSES

Un certain nombre de contraintes ont été imposées par le sujet, nous allons entrer en détail dans la gestion de celles-ci et la façon dont elles sont utilisées.

Tout d'abord, un certain nombre de classes ont dû être implémentées, au nombre de 8 :

- *Game* : classe principale qui gère l'application en elle-même, permet de gérer l'ajout des pays et de lancer les fonctions de la classe *WorldMap* pour l'affichage.
- *WorldMap* : classe qui contient les fonctions pour afficher la carte du monde et les continents.
- *ContinentMap* : La classe *ContinentMap* est une sous-classe de Continent. Elle représente une carte d'un continent spécifique.
- *SoundManager* : classe permettant de gérer le son.
- *Menu* : classe permettant d'afficher le menu principal, passe le relais par la suite à la classe *Game*
- *SFMLutilities* : classe qui gère les objets de la bibliothèque graphique SFML ; définit les zones cliquables correspondant à chaque continent/pays ainsi que l'affichage des différentes pages, des informations des pays, etc.
- *Country* : Classe définissant un pays avec son nom et ses autres informations.
- *Continent* : Classe abstraite qui hérite de la classe *ContinentMap* et qui contient l'information sur le nom du continent et les pays qui le composent.

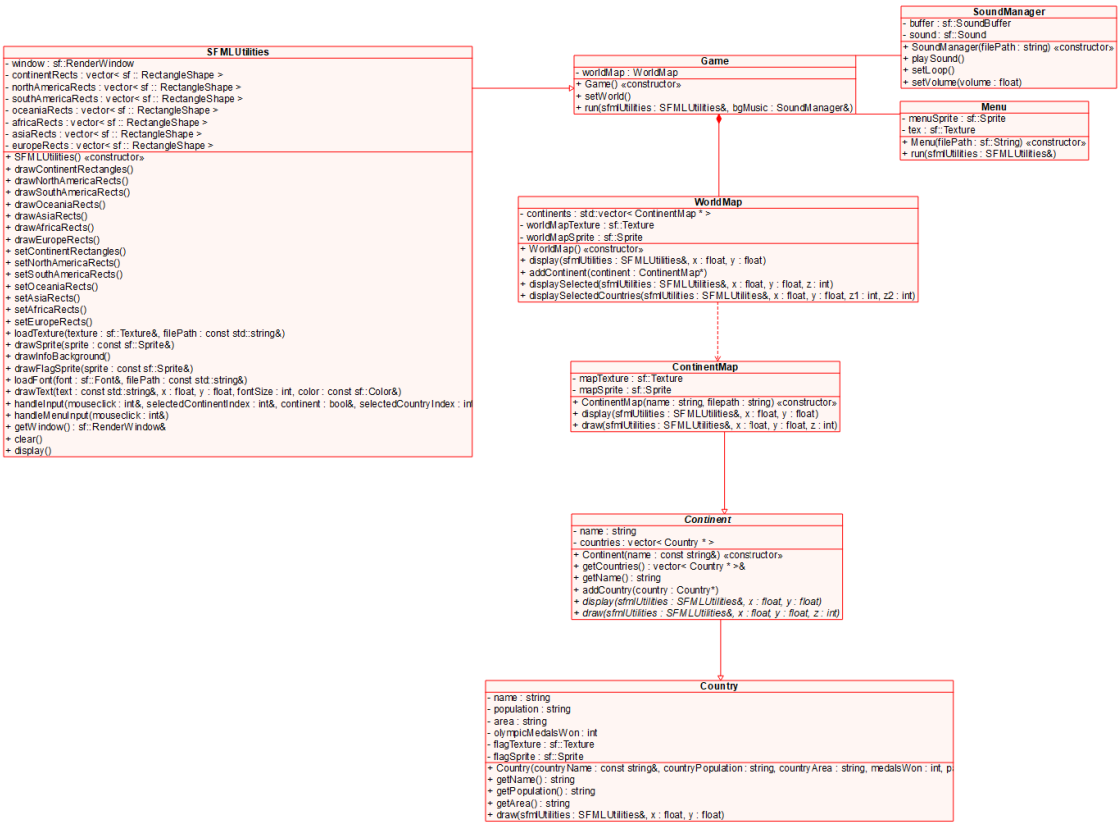
2.2 FONCTIONS VIRTUELLES

Nous avons implémenté de manière virtuelle dans la classe abstraite *Continent* les fonctions virtuelles *display* et *draw*. La classe *ContinentMap* peut ensuite les utiliser pour afficher les cartes des continents pour *display* ainsi que les informations des pays pour *draw*.

```
// Fonctions virtuelles
virtual void display(SFMLUtilities& sfmlUtilities, float x, float y) = 0;

virtual void draw(SFMLUtilities& sfmlUtilities, float x, float y, int z) = 0;
```

2.3 DIAGRAMME UML



3. PROCEDURE D'INSTALLATION ET D'EXECUTION DU CODE

3.1 ENIVRONNEMENT DE TRAVAIL ET BIBLIOTHEQUE SFML

L'environnement de développement utilisé est le suivant : programmation via les IDE Visual Studio Code et CLion et navigation avec WSL Ubuntu. L'installation de la bibliothèque SFML s'est faite simplement via l'instruction suivante :

```
sudo apt-get install libsfml-dev
```

Il est ensuite facile d'utiliser la SFML en incluant les modules utiles dans les fichiers du projet :

Module graphique dans *SFMLutilities* :

```
#include <SFML/Graphics.hpp>
```

Module audio dans *SoundManager* :

```
#include <SFML/Audio.hpp>
```

3.2 MAKEFILE

La compilation du projet est gérée par le *Makefile* qui prend en compte les options de la SFML et compile tous les fichiers contenant les classes.

```
CXX = g++
CXXFLAGS = -std=c++11 -Wall
SFMLFLAGS = -lsfml-graphics -lsfml-window -lsfml-system -lsfml-audio
EXEC = myGame
SRC = main.cpp

all: $(EXEC)

%.o: %.cpp
    $(CXX) -c $< -o $@

%.o: %.h
    $(CXX) -c $< -o $@

$(EXEC): $(SRC)
    $(CXX) $(CXXFLAGS) $(SRC) -o $(EXEC) $(SFMLFLAGS)

clean:
    @echo "** Removing object files and executable..."
    rm -f $(EXEC)
```

4. FIERTES ET CONCLUSION

Ce projet de création d'une carte interactive sur le thème des Jeux Olympiques à l'aide du langage C++ et de la bibliothèque SFML a été une expérience enrichissante et stimulante.

Notre principale fierté est la structuration du code avec la POO. Le code est modulaire et réutilisable, il est possible de rajouter des pays, des informations, des sons, etc., sans que cela pose des problèmes ou nécessite de revoir l'entièreté du code. Le rendu final, l'esthétique de l'application est aussi très satisfaisante, elle est facile a prendre en main et agréable à regarder.

À travers cette aventure, plusieurs aspects du développement orienté objet ont été abordés, mettant en lumière les compétences acquises et les défis surmontés. Même si l'application développée n'est pas révolutionnaire, ce fut un très bon exercice dans le cadre de notre formation en informatique.