POLYTECH
SORBONNE

Projet C++ :
Coupe du Monde

MAIN4

PREVOT Alexia
GUILLEMARE Bastien

Mercredi 27 janvier 2023

1 Description de l'application

Nous avons réalisé une application de révision de la théorie des galops en équitation.

1.1 Rapport au thème

L'application s'appelle "Direction Coupe du Monde".
ALors quel est le rapport avec le thème "Coupe du Monde" ?
En fait, l'application vise à aider les cavaliers à réviser et donc à les aider à passer les différents galops, avec pour objectif final de les préparer aux grands concours, comme la Coupe du monde. En effet, l'accès aux grands concours nécessite d'avoir le galop 7.

1.2 Fonctionnement

L'application comporte un menu principal avec les options "Réviser" et "Quiz". Lorsque l'utilisateur sélectionne "Réviser", il est dirigé vers une page où il peut choisir le niveau de galop (de 1 à 7) qu'il souhaite réviser. Cette page affiche des fiches de révision pour chaque niveau, contenant des informations sur les mouvements et les techniques à maîtriser pour passer chaque galop. Lorsque l'utilisateur sélectionne "Quiz", il est dirigé vers une page où il peut choisir le niveau de galop (de 1 à 7) qu'il souhaite passer. Il doit alors répondre à une série de questions pour préparer l'examen théorique.
Pour la partie révision, l'utilisateur peut passer d'une fiche à l'autre avec des boutons retour ou suivant.
Pour le quiz, en cliquant sur une réponse, soit le bouton devient rouge, ce qui signifie que c'est faux et alors la réponse s'affichera au dessus, soit le bouton devient vers. Dans les deux cas un bouton "suivant" apparaît pour passer à la question suivante. Il n'y a pas de retour en arrière possible pour le quiz, en effet l'utilisateur doit finir le questionnaire avant de pouvoir retourner au menu. Pour le moment, il y a environ 9 questions par niveau.

2 Les contraintes

Le projet devait répondre à un certain nombre de contraintes.

Contraintes :

- 8 classes : c'est ok (voir diagramme UML) ;
- 3 niveaux de hiérarchie : nous n'en avons que 2 avec *Screen* et *Quiz* par exemple. Rajouter une niveau de hiérarchie dans ce projet serait inutile ou aurait nécessité d'approfondir l'application en ajoutant de nouvelles fonctionnalités (manques de temps pour cela);

- 2 fonctions virtuelles : par exemple *HandleEvent* et *Draw* dans la classe *Screen* qui permettent de gérer les événements de la fenêtre et de dessiner des objets dessus. Elles sont redéfinis dans la classe fille *Quizz* par exemple ;
- 2 surcharges d'opérateurs : le + et le << de la classe question ;
- 2 conteneurs différents de la STL : *list* pour la liste de questions dans *Quizz* et *vector* pour les propositions des questions dans *Question* ;
- diagramme de classe UML complet : voir section 3 ;
- commentaire du code : le code est commenté en anglais ;
- pas d'erreurs avec Valgrind : c'est ok ;
- pas de méthodes/fonctions de plus de 30 lignes : c'est ok sauf pour *RunQuestion* qui atteint les 40 lignes et *Update* à 37 lignes dans le dossier *Quizz.cpp*. Ces fonctions nécessites des beaucoup conditions à gérer (if, else, else if), qui occupe la majorité des lignes et donc cela n'empêche pas leur bonne compréhension. On a estimer que la création de sous fonctions dans leurs cas serait inutile et n'améliorerait pas la lisibilité ;
- utilisation de tests unitaires : ils testent la bonne récupération des questions du fichier xml avec la classe *Question*
- utilisation d'un Makefile avec une regle "all" et "clean" ou autre outil de build automatique : c'est ok.

Ainsi, la majorité des contraintes sont respectées même si certaines ne le sont pas à 100%. Nous aurions sans doute eu besoin d'un peu plus de temps pour travailler sur ce projet et le compléter afin de tout respecter.

3 Diagramme UML

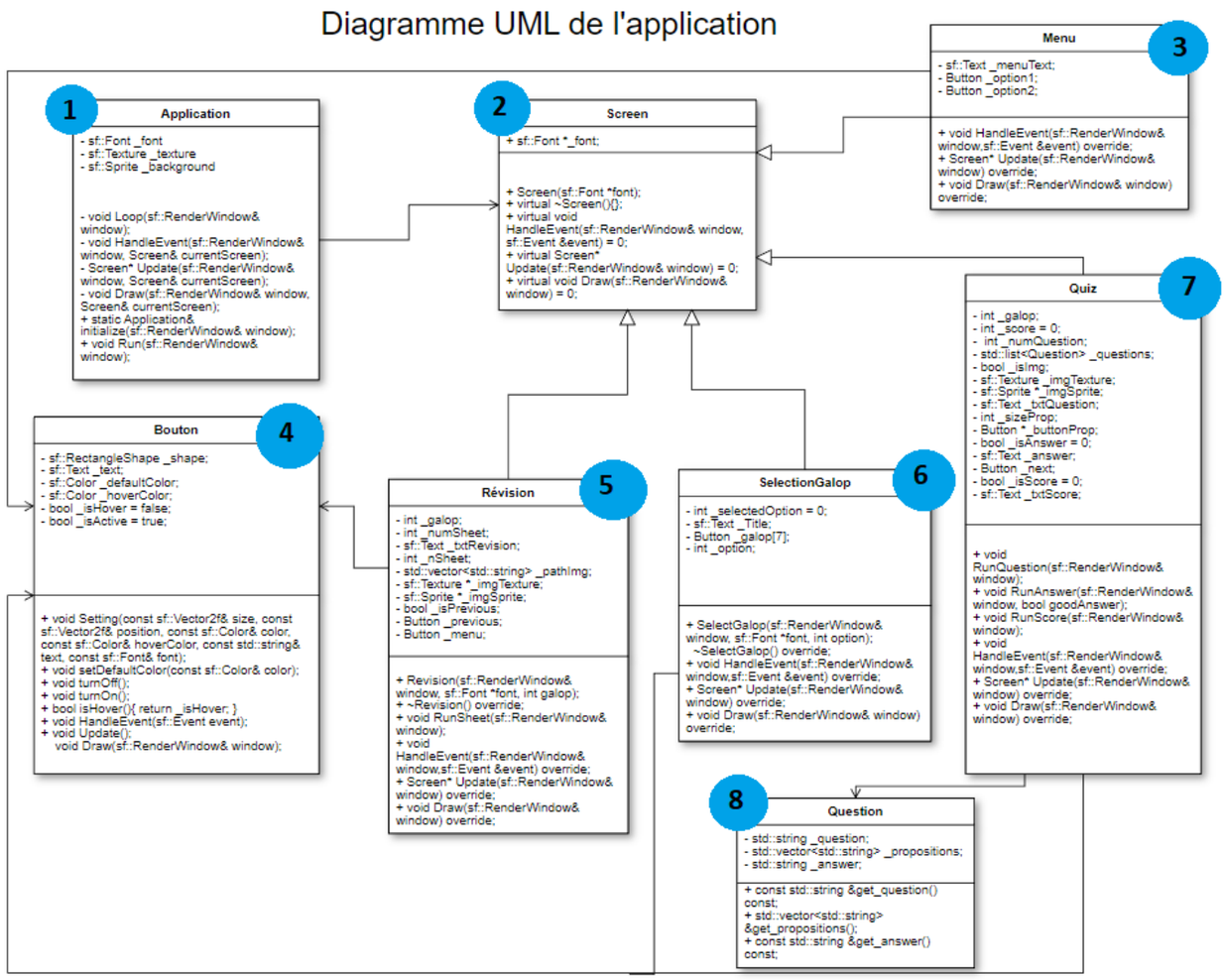

Figure 1: Vu globale du diagramme UML

Comme la taille est trop petite sur la vue globale du diagramme UML, voici chaque classe détaillée :

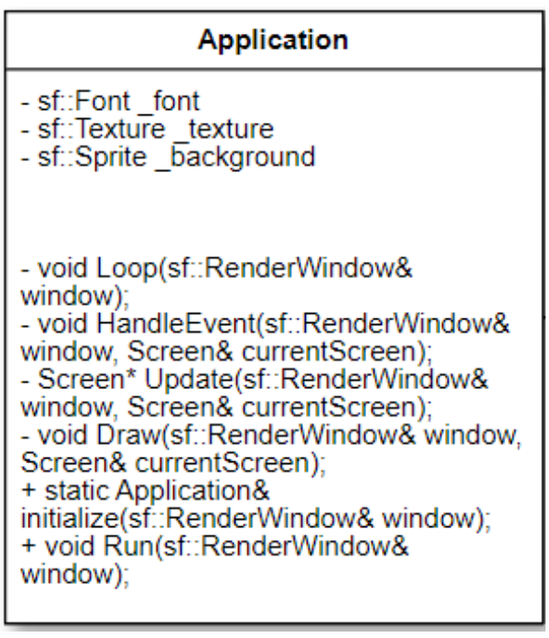

Figure 2: Application

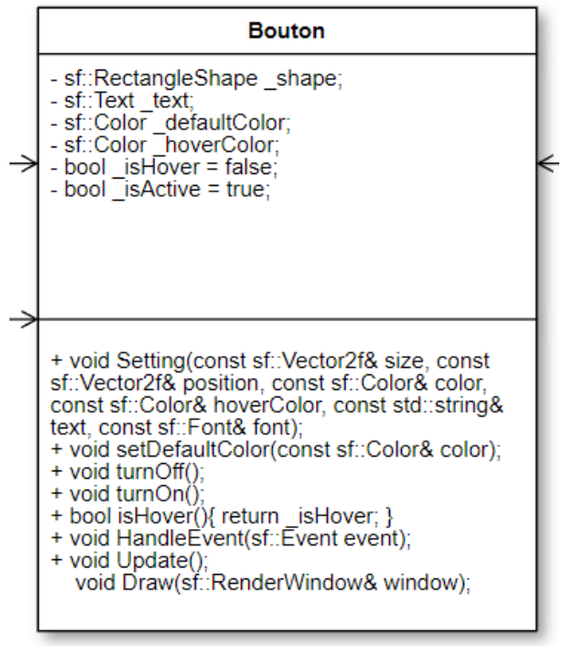

Figure 3: Bouton

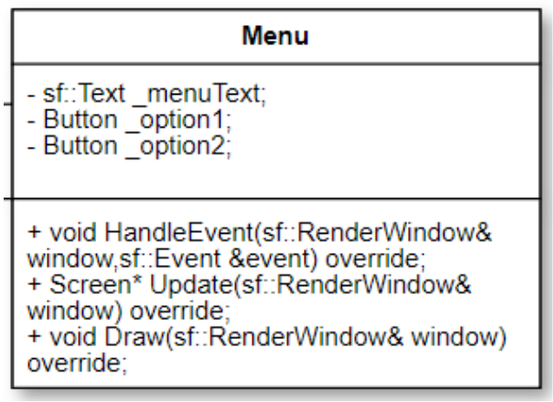

Figure 4: Menu

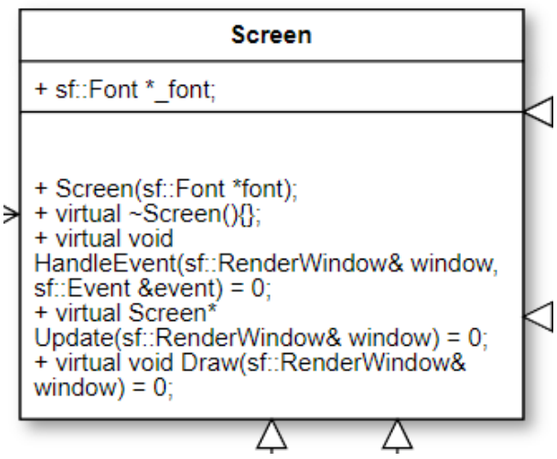

Figure 5: Screen

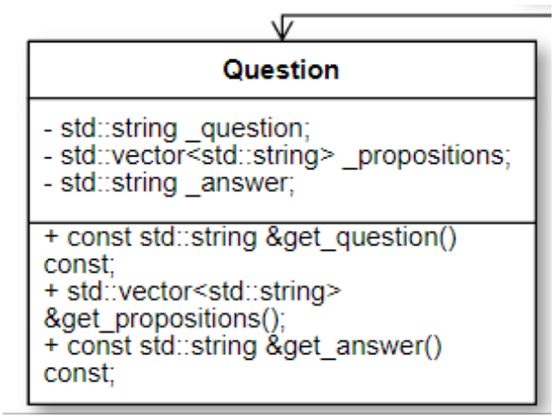

Figure 6: Question

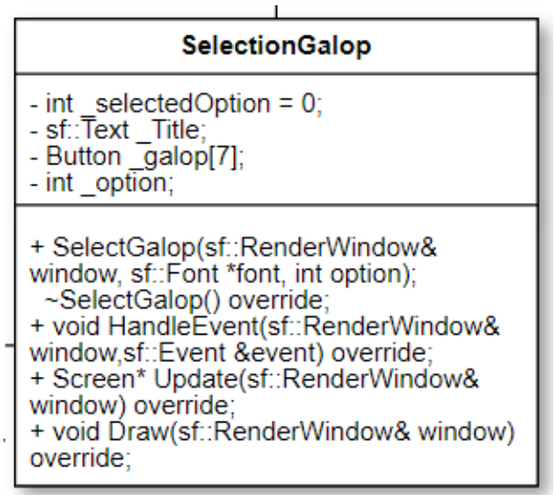

Figure 7: SelectGalop

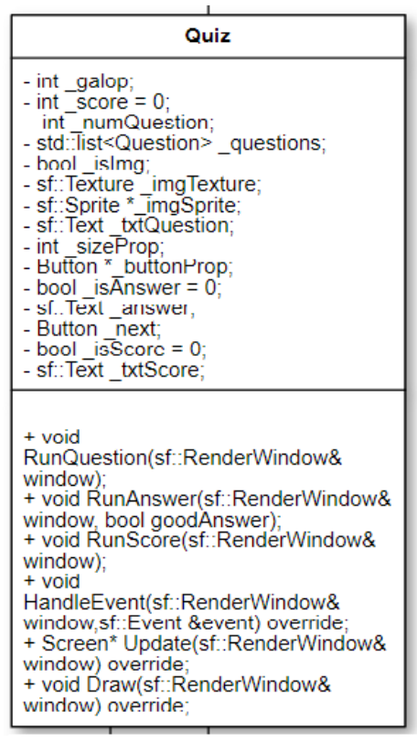

Figure 8: Quiz

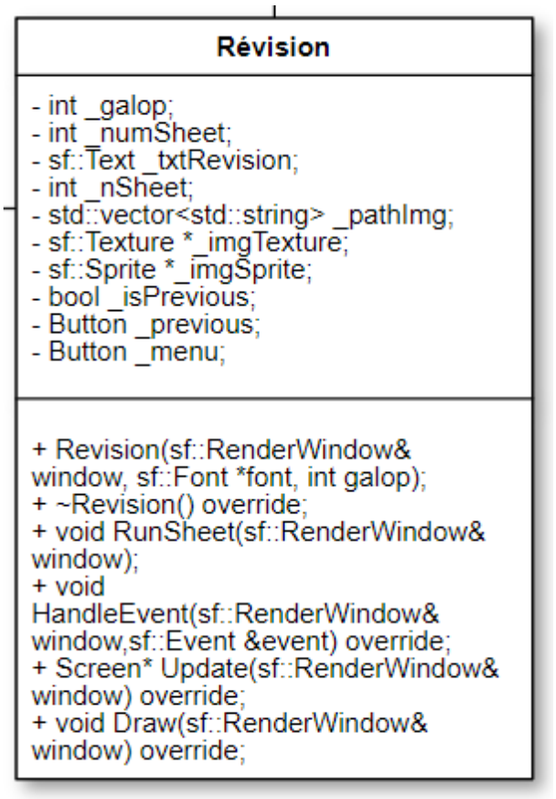

Figure 9: Revision

4 Procédure d'installation (bibliothèques ...) et d'exécution du code

Ces procédures sont expliquées dans un fichier *README.md*.

4.1 Procédure d'installation

Les bibliothèques suivantes sont nécessaires au fonctionnement du programme :
- la bibliothèque graphique SFML : https://www.sfml-dev.org/
- la bibliothèque GNU : https://www.gnu.org/

4.2 Procédure d"exécution du code

Il faut cloner le projet en écrivant dans un terminal Linux :
" git clone https://github.com/Alexia57/WorldCup.git "
Pour compiler et lancer le projet :
" make " " ./projet "
Pour compiler et lancer les tests unitaires :
" make testcase " " ./testcase "

5 Fiertés

Nous sommes tous les deux très content de ce que nous avons produit. Nous étions partis de loin comme on ne savait pas utiliser d'interface graphique mais les heures passées pour apprendre en valaient le coup. L'application est optionnelle et assez agréable à regarder et utiliser et c'est tout ce que nous voulions !

Pour ma part, Alexia, c'était un projet qui me tenait à coeur comme le sujet touche à ma passion et par ailleurs mes amis ou famille dans ce domaine sont pressés de voir à quoi ressemble cette application (qu'on espère pouvoir peaufiner et rajouter plus de question par la suite). Je suis aussi très contente de voir que Bastien a aussi été très motivé pour ce projet et je pense qu'il aura appris des choses sur le monde de l'équitation !

Enfin, on a caché des photos personnelles dans le quiz, les avez-vous trouvé ?