



Cycle ingénieur - 2ème année

Programmation C++

Rapport de projet d'initiation à la programmation C++

par

Thomas Fiou

Ruben Aloukou

Maxime Dubus

Salim Darouèche

Rayan Rabei

Enseignant : Dominique Yolin

Groupe : ING2 GSI 3

I. Présentation du sujet

Sujet : Développer un logiciel pouvant être soit un « casual gaming » soit un utilitaire, mettant en application les principes de POO. Le sujet est à la discrétion des étudiants afin qu'ils s'approprient pleinement le projet, il n'est qu'un prétexte pour leur permettre de mettre en pratique les aspects théoriques du C++ vus en cours.

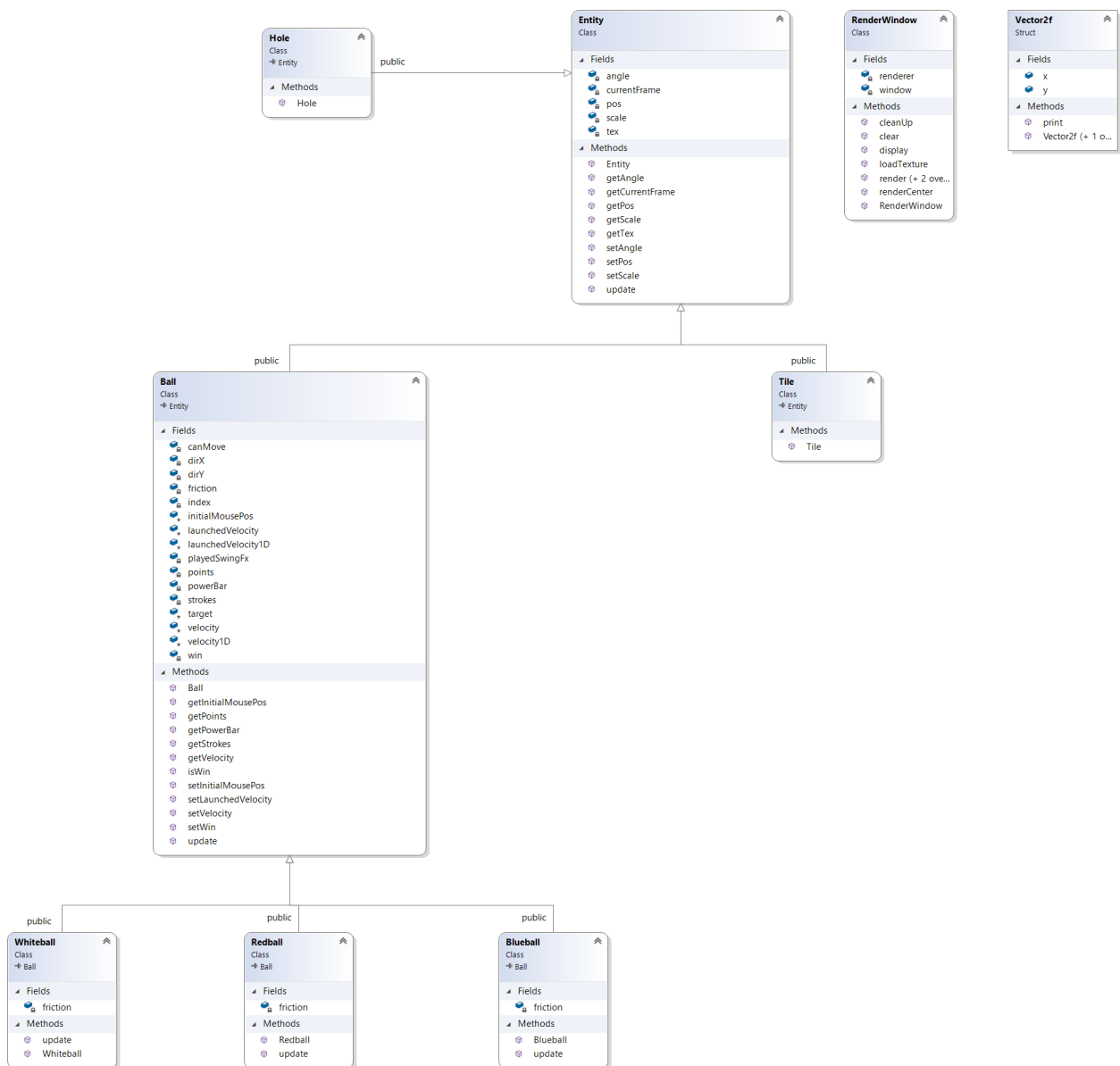
Contraintes :

- Par groupe de 3 à 5 étudiants
- Utilisation de diagramme UML
- Commentaires et convention de nommage requis
- Utilisation d'une librairie externe (Raylib, Gtk, SDL etc.)
- Programmation orientée objet :
 - Encapsulation des données et des méthodes requise
 - 8 classes attendues au minimum
 - 2 niveaux d'héritages attendus au minimum
 - Polymorphisme requis
- Enregistrement de données sur un fichier texte (.txt) requis
- Surcharge d'opérateur requis
- Travail collaboratif sur Github (commits réguliers, messages explicites etc.)

II. Présentation du projet

A. Fonctionnalités principales

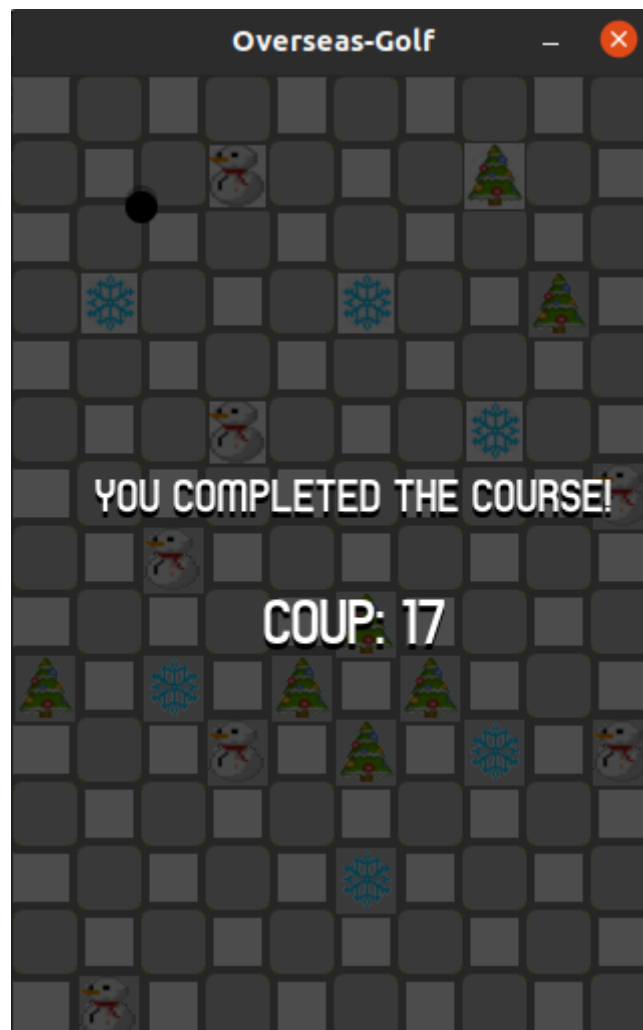
Diagramme UML :



Lorsque le joueur lance ce jeu de style mini-golf, il se retrouve sur un écran d'accueil. Le joueur peut alors lancer la partie avec le ***clic gauche***. Le premier niveau est alors chargé. La balle est déplaçable à l'aide de la souris : cliquer et maintenir le ***clic gauche*** affiche sur l'écran une flèche indiquant la direction que la balle va prendre ainsi qu'une barre de puissance pour le coup à venir. En déplaçant la souris on peut donc orienter son coup et ajuster la puissance de celui-ci. Une fois le trou atteint, le niveau suivant est chargé. Tout au long des niveaux, le compteur de coup nous indique notre performance sur le parcours.



Une fois le dernier niveau terminé, un écran de fin apparaît. On y voit notamment le score obtenu après avoir fini tous les niveaux. Le joueur peut alors relancer le jeu s'il le souhaite. Ses scores sont enregistrés sur un fichier **scores.txt** et ils sont à chaque fois accompagnés de la date et de l'heure à laquelle ils ont été obtenus.



B. Composantes externes utilisées

Simple DirectMedia Layer (SDL) est une bibliothèque logicielle libre, utilisée pour créer des applications multimédias en deux dimensions pouvant comprendre du son comme les jeux vidéo, les démos graphiques, les émulateurs, etc. Nous avons préféré utiliser cette bibliothèque pour notre jeu car nous avons trouvé son utilisation simple et elle disposait d'une grande documentation sur internet.

Pour notre projet, nous avons choisi d'utiliser les modules complémentaires de la bibliothèque SDL2 suivants :

















SDL_ttf : Permet l'utilisation des polices de caractères TrueType Font (TTF)

SDL_image : Prendre en charge plusieurs formats d'images comme JPEG, PNG etc.

SDL_mixer : Prendre en charge plusieurs formats de sons comme MP3

C. Gestion de projet

Répartition des tâches et des rôles de chacun :

Gestion et organisation git					
Modélisation UML du projet					
Programmation orienté objet					
Enregistrement des scores					
Construction des niveaux					
Respect du clean-code					

Code couleur :

 : Ruben  : Thomas

 : Maxime  : Salim  : Rayan

Planning :

Sprint 1 : 21 février 2022 - 27 février 2022

- Gestion et organisation git (création et mise en place de méthode de travail)
 - Github depuis Git Classroom
- Modélisation UML du projet (nombre de classe, relation etc.)
 - Réalisé à l'aide du logiciel StarUML

Sprint 2 : 18 avril 2022 - 24 avril 2022

- Création des classes, surcharge d'opérateur et des relations d'héritage
 - Réalisé sous Visual Studio Code
- Enregistrement des scores sur fichier texte (.txt)
 - Réalisé avec les librairies ofstream

Sprint 3 : 25 avril 2022 - 1 mai 2022

- Respect du clean-code (convention de nommage, commentaire de code etc.)
- Construction des niveaux (positionnement des obstacles et des trous)
 - Réalisé avec la librairie externe SDL

Convention de nommage :

Exemple : ~~void update()~~ → **void void_update()**

Exception pour les pointeurs : ~~Vector2f p_pos~~ → **Vector2f pvect_pos**

Autres exemples : **SDL_Texture t** → **SDL_Texture text_t**