

---

# *UE Technologie Objet - Projet Long*

## Mini-Golf Simulator 7

Rapport - Version Finale

---



**Groupe :** E33

**Membres :**

Théo COUTURIEUX

Cyril FIGUIN

Théodore NAKACHE

Dewi RICQUEBOURG

Valentin THEDON

**Année Universitaire :** 2019-2020

**Semestre :** 6



# Préface

Ce rapport fait suite à la troisième et dernière itération du projet long de Technologie Objet, enseigné à l'Ecole Nationale Supérieure d'Electrotechnique, d'Electronique, d'Informatique, d'Hydraulique et des Télécommunications dans le département Sciences du Numérique en première année.

Il est accompagné d'un manuel utilisateur, d'une archive contenant le code du projet en cours, de planches de présentation de notre projet et d'un fichier .txt de répartition des points.

# Sommaire

<b>Introduction</b>	<b>2</b>
Rappel du sujet	2
Objectifs du projet long	2
<b>Fonctionnalités</b>	<b>2</b>
Mode de jeu classique	3
Mode de jeu Coopératif	3
Éditeur de terrains	3
Boutique du jeu	3
<b>Découpage de notre application</b>	<b>3</b>
Modèle	3
Vue	3
Contrôleur	3
<b>Diagrammes de notre projet</b>	<b>4</b>
<b>Choix de conceptions</b>	<b>4</b>
La librairie SLICK2D pour l'interface graphique:	4
Librairies additionnelles pour de nouvelles fonctions	4
<b>Problèmes rencontrés et solutions apportées</b>	<b>5</b>
<b>Présentation de notre groupe et répartition des tâches</b>	<b>5</b>
<b>Conclusion</b>	<b>5</b>



## **Introduction**

### *Rappel du sujet*

Le premier objectif de ce projet était de choisir en équipe un sujet de projet long. Après avoir proposé plusieurs sujets, c'est le Simulateur de Mini-Golf qui a été retenu.

Les fonctionnalités sont énoncées dans le fichier fonctionnalités.pdf et sont plus détaillées dans le fichier manuel-utilisateur3.pdf.

### *Objectifs du projet long*

Ce projet long à pour objectif :

- Mise en application du cours Technologie Objet enseigné à l'ENSEEIH
- Mise en oeuvre des Méthodes Agiles SCRUM enseignées à l'ENSEEIH pour améliorer le travail d'équipe



## Fonctionnalités

Nous proposons à l'utilisateur différentes fonctionnalités. Le détail se trouve sur notre manuel utilisateur.

### *Mode de jeu classique*

Il s'agit du jeu de mini golf classique. Deux joueurs essaient de réaliser le parcours pour avoir les meilleures performances. A la fin d'un parcours, un tableau des scores apparaît et les joueurs peuvent quitter la partie. Ce mode respecte les règles du mini golf classique.

**Nous avons bien réussi à la mettre en place et elle fonctionne. (Fini à l'itération 3)**

### *Mode de jeu Coopératif*

L'objectif est le même que précédemment, faire le moins de coup possible par trous mais cette fois-ci en cumulant le score des deux joueurs.

**Le modèle est disponible mais pas l'interface graphique pour ce mode. (Commencé à l'itération 2)**

### *Éditeur de terrains*

Nous proposons à l'utilisateur de créer son propre parcours selon ses envies.

- Ajout du point de départ et d'arrivée (nécessaire à la création d'un terrain)
- Ajout de textures (sable, green, eau, ...)
- Ajout d'obstacles
- Ajout de tapis roulant

Il donnera par la suite un nom à son terrain et précisera en combien de coup maximum il devra être effectué.

**Nous avons bien réussi à mettre en place cette fonctionnalité (Fini à l'itération 3)**

### *Boutique du jeu*

Enfin, nous proposons une boutique où l'utilisateur pourra, via un système monétaire propre au jeu, acheter de nouvelles balles (plus performantes par exemple) et des nouveaux parcours.

**Par manque de temps, cette fonctionnalité n'est pas disponible. Le système d'inventaire en revanche est implémenté et utilisé dans le pré-jeu de partie Classique. (Commencé à l'itération 3)**



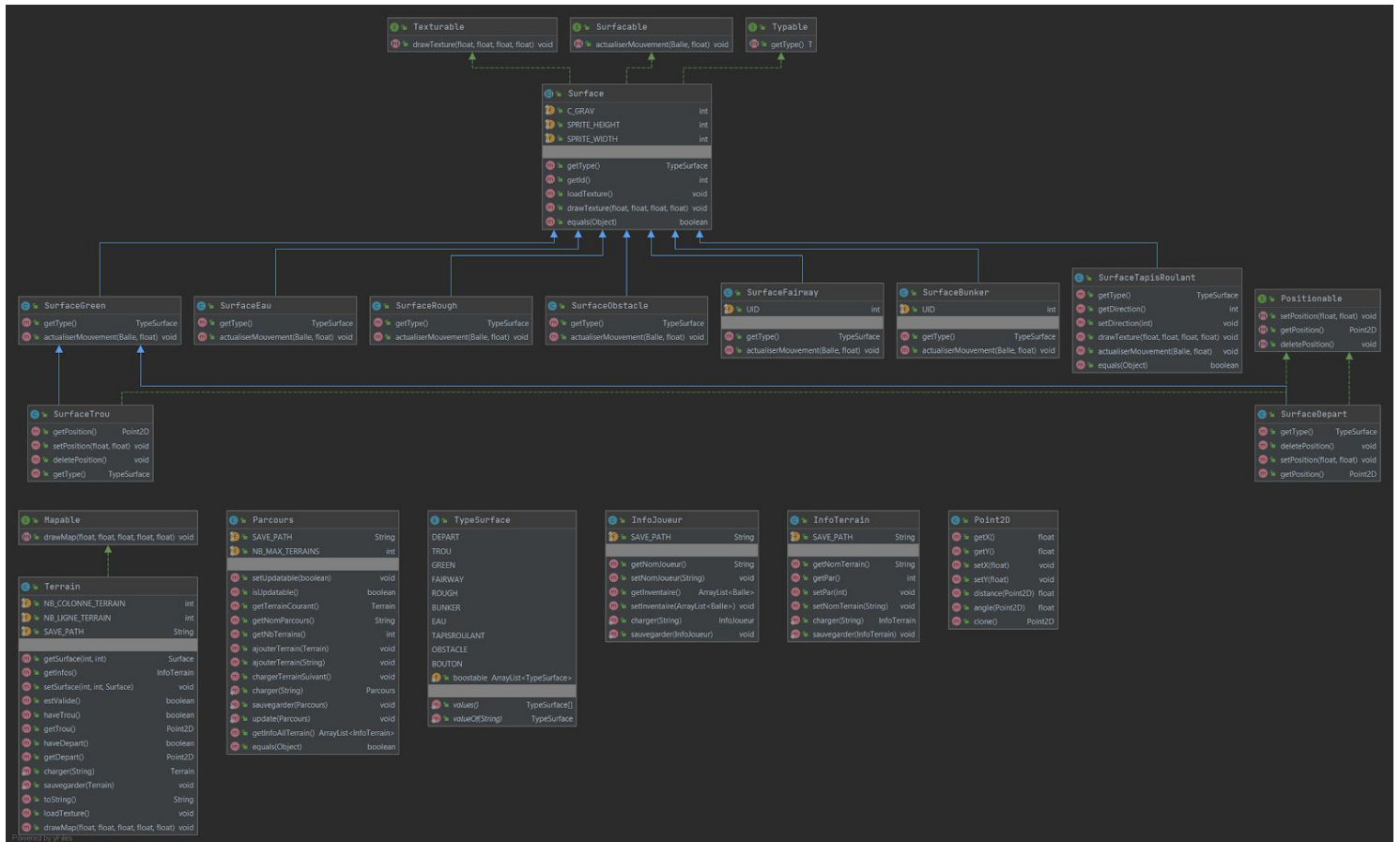
## Découpage de notre application

Nous avons tout d'abord tenté de découper notre programme pour qu'elle suive un patron MVC. Mais le choix de l'interface graphique nous à amener à l'adapter.

Nous avons découpé notre application en 5 packages.

- Jeu : il est constitué de tous les éléments qui relèvent du modèle de la partie jouable de l'application. Les classes essentielles au jeu son Balle, Joueur et Arbitre.
- Golf : ce paquet rassemble tous les éléments qui constitue la représentation et le stockage des terrains. On y retrouve notamment les classe Parcours, Terrain, Surface
- Interface : il regroupe tous les éléments graphiques utilisés. Il est notamment le paquet de la classe principal ViewController.
- Éditeur : il contient les éléments relatifs à l'éditeur de terrain.
- Exceptions : Elle regroupe toutes les classes d'exceptions spécifiques à l'application.

## Package Golf



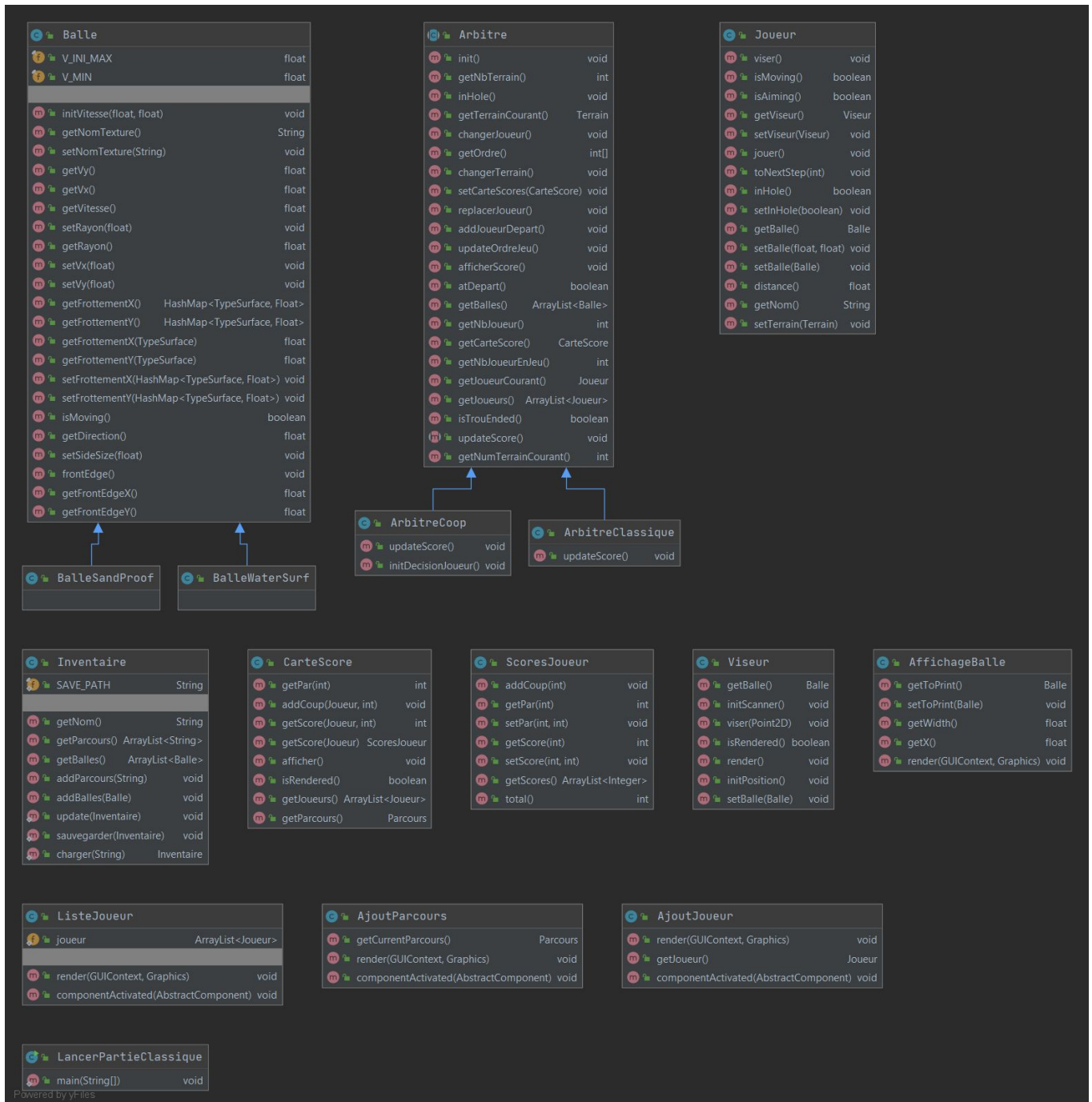
Page 5

# Mini-Golf Simulator 7

## Rapport Final



## Package jeu



(voir packageJeu.png dans livrables)

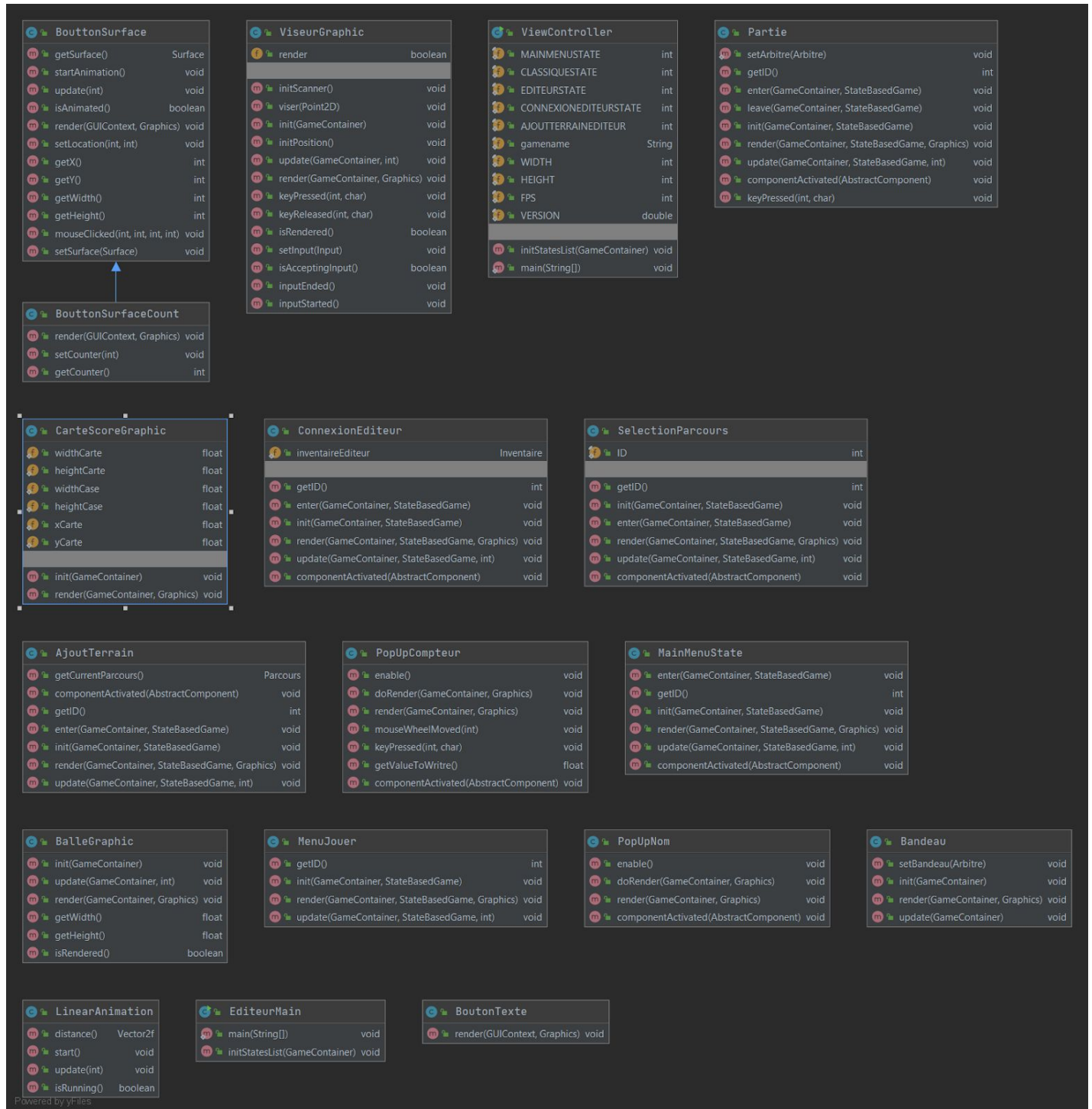


# Mini-Golf Simulator 7

## Rapport Final



## Package interfaces (Vue + Controller)

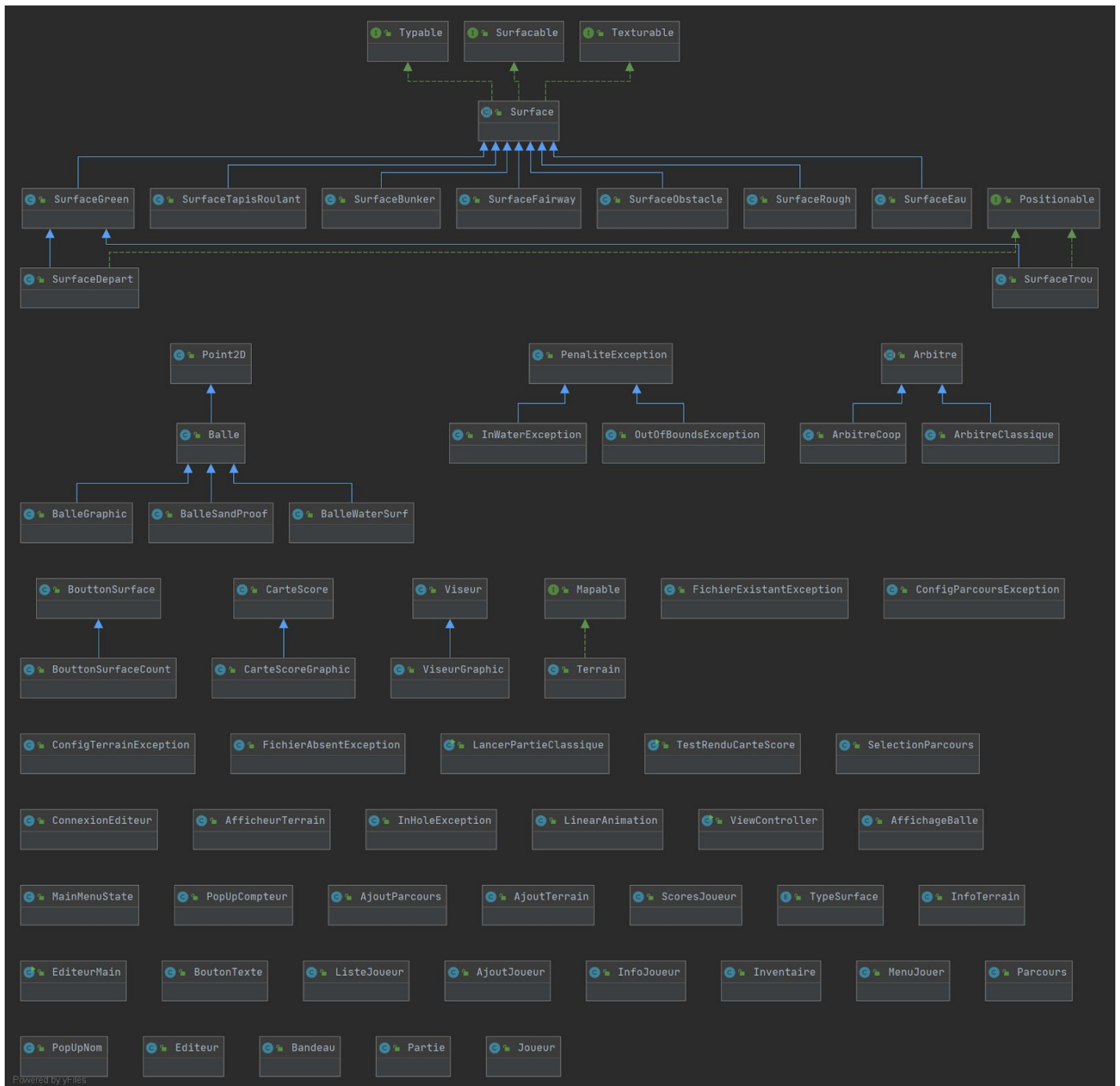


(voir packageInterfaces.png dans livrables)





## Global



(voir global.png dans livrables)



## Choix de conceptions

Initialement, nous avons décidé d'utiliser la bibliothèque Swing (ou JavaFX) pour l'élaboration de notre interface graphique. Néanmoins, nous nous sommes rendu compte qu'elle était limitée pour la création de jeux en 2D. Notre recherche de librairie pour l'interface graphique nous a mené à choisir la librairie Slick2D.

### *La librairie SLICK2D pour l'interface graphique:*

Slick2D est une librairie de programmation de jeux vidéo en java s'appuyant sur [OpenGL](#) via [LWJGL](#). [source : <http://www.shionn.org/tutoriels-slick-2d>]. Nous nous sommes donc documentés sur l'utilisation de cette nouvelle librairie pour pouvoir coder notre application.

### *Librairies additionnelles pour de nouvelles fonctions*

Pour ajouter de la musique sur notre interface, nous avons choisis les librairies Jorbis et Jogg. L'utilisation de ces bibliothèques ont été simples, nous ne détaillerons pas cette partie dans ce présent rapport.

## Problèmes rencontrés et solutions apportées

Ce projet s'inscrit dans une démarche d'apprentissage de réalisation de projets. De ce fait nous avons mis en pratique les fondements des méthodes agiles. Comme attendu, nous avons dû réadapter de nombreux détails de la méthodologie tout au long des itérations.

La solution apportée est une attention importante apportée à l'agilité qui ne sera vraisemblablement pas autant nécessaire dans les futurs projets.

Pour réaliser un jeu de golf complexe, nous avons dû maîtriser des bibliothèques différentes de celles apprises en cours. Comme chacun s'est concentré sur certaines bibliothèques plutôt que d'autres, il a été plus compliqué de comprendre et de modifier certaines classes en conséquence. Pour palier à cette difficulté, nous avons mis en place des réunions d'équipe hebdomadaires et nous avons régulièrement travaillé en binôme.



Nous nous sommes rendus compte tardivement que la bibliothèque Slick2d n'était pas compatible avec le patron MVC. Deux choix s'offraient à nous :

- Changer de bibliothèque
- Revoir le modèle (Arbitre et Joueur ainsi que les classes qui en découle)

Nous avons choisi de revoir le modèle pour l'adapter à Slick. Il nous semblait que ce choix serait plus rapide au vue de ce que nous avons déjà accompli.

Nous avons finalement pu faire fonctionner notre jeu après ces modifications.

## **Présentation de notre groupe et répartition des tâches**

Notre groupe se compose de **Théo COUTURIEUX**, **Cyril FIGUIN**, **Théodore NAKACHE**, **Dewi RICQUEBOURG** et **Valentin THEDON**, étudiants en première année à l'ENSEEIH.

Pour ce qui est de la répartition des différentes tâches de notre projet, nous avons décidé d'utiliser Trello, en respectant la méthode Agile SCRUM.

Malgré les conditions de communication plus difficile qu'il n'était prévu nous avons su trouver des moyens diversifiés pour rester en contact (Mise en place de conversation Discord ainsi que des appels hebdomadaires).

Cyril s'est occupé de l'affichage du menu et de la mise en place du contrôleur de phases de jeux et de la rédaction globale des rapports. Valentin s'est occupé de la structure des parcours et terrains, de leur édition et de leur sauvegarde. Théo s'est occupé du modèle, arbitre, joueur et les classes qui en découle, de la cohérence globale ainsi que de l'interface graphique du gameplay de partie classique. Théodore s'est occupé des surfaces, de la jouabilité d'une partie classique terminal et de l'inventaire et enfin Dewi s'est occupé de la cohérence vue-contrôleur.

Pour chaque itération, un rapport personnel a été écrit par chaque membre de l'équipe afin d'explicitier ses tâches au cours de la période en cours.