



Projet Long TOB F1 Simulateur 2022

Auteurs :

Timothée BLANCHY

Martin CAISSIAL

Mathieu CAMPAN

Bryan CHEN

Simon HAUTESERRES

Yu-Chi LEE

Sofiane TALEB

Equipe CD-2

23 avril 2022

Résumé

Dans le cadre d'un projet en Technologie Objet, matière de première année en Sciences du Numérique à l'ENSEEIH, notre groupe constitué de 7 étudiants, doit réaliser une application orientée objet, en six semaines. Le travail sera à réaliser en trois itérations de deux semaines. Vous trouverez dans ce rapport une présentation détaillée de notre sujet ainsi que le travail réalisé lors de chaque itération.

Table des matières

I	Présentation du sujet	1
1	Idée de base	2
2	Fonctionnalités	2
2.1	Must Have	2
2.1.1	Gestion partie basique	2
2.1.2	Éléments basiques	2
2.1.3	Courses	2
2.1.4	Améliorations	2
2.2	Should Have	3
2.3	Could Have	3
3	Architecture de l'application	3
II	Suivi des itérations	5
1	Première itération	6
1.1	Travail global réalisé	6
1.2	Détails sur le travail réalisé	6
1.3	Choix de conception et réalisation	7
1.4	Problèmes rencontrés et solutions apportées	7
2	Deuxième itération	7
2.1	Travail global réalisé	7
2.2	Détails sur le travail réalisé	7
2.3	Choix de conception et réalisation	8
2.4	Problèmes rencontrés et solutions apportées	8
III	Organisation de l'équipe et mise en oeuvre des méthodes agiles	9
1	Organisation de l'équipe	10
2	Mise en oeuvre des méthodes agiles	10

Première partie

Présentation du sujet

1 Idée de base

Après plusieurs propositions de sujets, nous avons décidé de créer un jeu de management de F1.

Le concept de l'application est plutôt simple : le joueur prend en charge une écurie en tant que directeur d'équipe. Son objectif : Amener son équipe au sommet.

Pendant les courses, le joueur supervise la stratégie de deux pilotes de son écurie, qui courent automatiquement le long des circuits. La simulation de course prend en compte différents éléments perturbateurs (météo, accident, incident de course) ainsi que des aspects techniques et stratégiques (quand les pilotes doivent se ravitailler, avec quelle gomme ils doivent changer de pneus, quand ils doivent conduire plus vite ou plus lentement, etc). Les choix du joueur influenceront grandement le déroulement de la course et permettront de débloquent des scénarios plus ou moins favorables pour remporter la coupe et décrocher le plus de points possibles.

Entre deux courses, le joueur pourra également augmenter les capacités de ses pilotes en leur faisant suivre des sessions d'entraînement. Il pourra également gérer et améliorer les divers composants de la voiture tels que le moteur, les freins, l'aileron, etc. Tout cela se fera grâce aux infrastructures de l'écurie et à son budget limité, qui varie selon le nombre de victoires ou de points remportés. Le classement de l'écurie et celui des pilotes influenceront donc les capacités techniques et sportives que pourra améliorer le joueur.

2 Fonctionnalités

Vous trouverez dans la liste ci-dessous les différentes fonctionnalités que nous aimerions implémenter à notre application. Vous trouverez également notre degré d'avancement pour chaque fonctionnalité :

- Gestionnaire du jeu
- Lancement du jeu (création ou poursuite de la partie) à partir d'un menu
- Sauvegardes

2.1 Must Have

2.1.1 Gestion partie basique

En cours de création / interface graphique non implémenté

- Gestionnaire du jeu
- Lancement du jeu (création ou poursuite de la partie) à partir d'un menu
- Sauvegardes non implémenté

2.1.2 Éléments basiques

Implémentés

- 1 Voiture
- 2 Pilotes
- 1 Écurie
- 1 Circuit
- 1 Barre de niveau

2.1.3 Courses

Non implémentés

- Lancement de simulations automatiques de courses (résultats, progression)
- Système de récompense après la course
- Génération et affichage du classement

2.1.4 Améliorations

En cours d'implémentation (version textuelle)

- Menu d'améliorations Voiture (niveaux)
- Menu d'améliorations Pilote

- Non implémenté
- Départements R&D : recherche et développement, amélioration de voiture en mieux
- Création et choix d'écuries
- Options : niveau de difficulté
- Choix des pilotes / voitures
- Gestion des courses et évènements aléatoires

- Non implémenté
- Personnalisation de l'écurie
- Personnalisation des voitures
- Quick Time Event pendant les courses
- Génération des circuits

Nous avons organisé notre code pour respecter le modèle MCV (Modele Controle Vue).



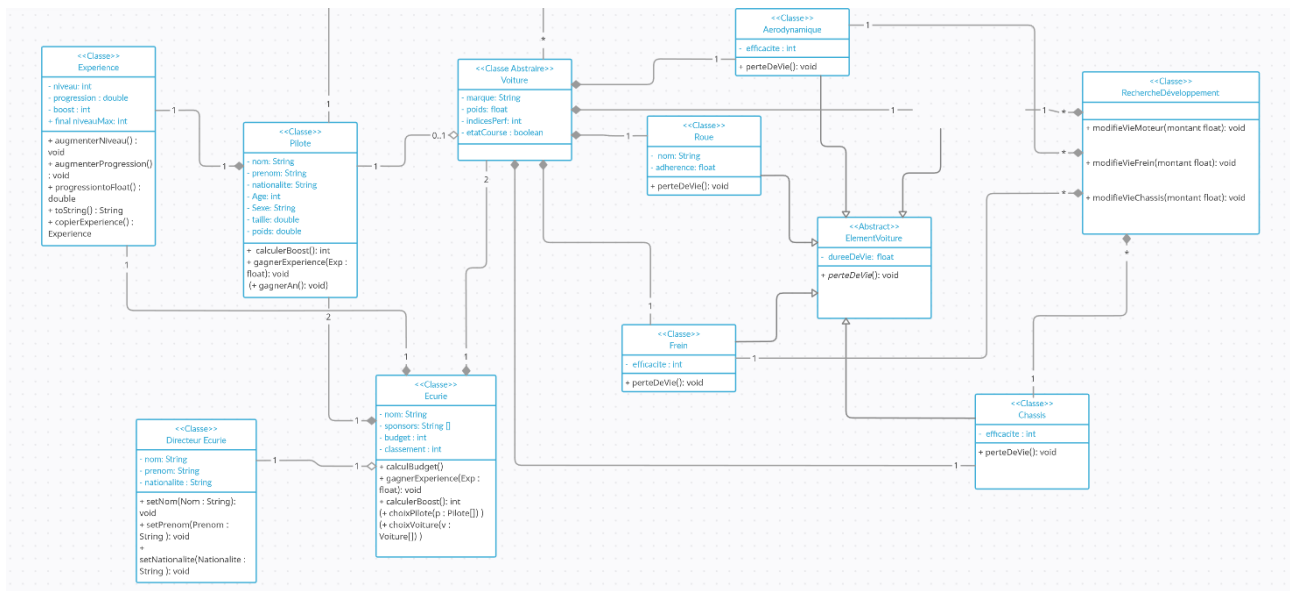


FIGURE 1 – Diagramme de classe de l'application

Deuxième partie

Suivi des itérations

1 Première itération

1.1 Travail global réalisé

Lors de cette première itération, nous avons grandement débuté la construction du modèle en créant la plupart des classes nécessaires à notre application : Pilote, Voiture, Ecurie, DirecteurEcurie, Classement, Circuit, etc.

Nous avons également écrit quelques programmes de tests qui veillent au bon fonctionnement de nos classes : EcurieTest, ExperienceTest, VirageTest, VoitureTest, PiloteTest, etc.

De même, nous avons commencé la construction des menus, qui serviront d'interface avec l'utilisateur.

1.2 Détails sur le travail réalisé

Timothée s'est tout d'abord chargé de la programmation des menus. Un main a été créé afin de lancer le jeu. Ce lancement de jeu est relayé par une classe moteur de jeu qui permet l'affichage d'un menu de lancement qui permet lui-même de jouer ou bien de quitter le jeu.

Lorsque que le joueur sélectionne l'option jouer, on aboutit à l'affichage d'un menu principal avec une brève présentation du jeu. Il nous permet pour le moment de jouer le jeu (lancer une course) avec une écurie de base uniquement. Le lancement de la course n'aboutit pour le moment à rien, il est en attente de l'implantation de circuit et classement.

Timothée est également en réflexion pour la programmation de sous-menus permettant de gérer et améliorer les pilotes et la voiture d'une écurie.

Mathieu a été chargé d'implémenter la classe Ecurie.

Une écurie est définie par un nom, une description, des sponsors, mais également d'un budget qui permettra l'amélioration d'une voiture, d'un classement définissant sa renommée et bien évidemment de deux pilotes et d'une voiture.

Un programme de test est en cours d'écriture par Sofiane.

Martin quant à lui s'est occupé de la classe Voiture.

Une voiture est définie par une marque, un certain poids, des indices de performances, d'un booléen indiquant si elle est actuellement en course et bien évidemment d'un pilote.

Un programme de test a également été réalisé, notamment pour vérifier de la bonne construction d'un objet voiture.

Les éléments de la voiture tels que l'aérodynamique, le châssis ou encore les roues ont été créés sans être implémentés par Sofiane.

Sofiane s'est occupé d'implémenter la classe Pilote.

Un pilote est composé d'un nom, un prénom, une taille, un poids, une nationalité, son sexe et son expérience. Ces éléments nous serviront à calculer sa performance lors des courses.

Simon s'est occupé de la création de la classe Circuit. Elle a pour but de simuler les différents temps de passage d'une certaine voiture sur les différents secteurs du circuit.

Un circuit est défini par un nom, une description et un nombre de virages. Chaque circuit comporte une ligne de stand et est constitué de 3 secteurs. Chaque secteur est lui-même constitué de virages et de lignes droites.

De nombreuses mesures ont été réalisées à l'aide du jeu F1 2021 et permettent le calcul d'un temps minimal sur chaque section de circuit et devront permettre très prochainement de simuler un temps réaliste de course avec une voiture de base.

Finalement Bryan s'est chargé de la classe `Expérience` qui servira autant pour l'expérience de l'écurie que des pilotes. Définie par un niveau, une progression et un boost, elle permettra d'améliorer les temps des pilotes durant la course, à mesure qu'ils prennent de l'expérience.

Des programmes de test ont également été réalisés.

1.3 Choix de conception et réalisation

1.4 Problèmes rencontrés et solutions apportées

Un diagramme de classe très complexe. Une architecture qui l'est d'autant plus : il était difficile de savoir par où commencer. Nous avons rencontré quelques conflits lors du commit sur svn mais rien d'insolvable.

2 Deuxième itération

2.1 Travail global réalisé

Lors de cette deuxième itération, nous avons tout d'abord poursuivi la création de classes importantes dans notre application telles que `Classement`, `Roue`, `Frein` et `Châssis`.

Nous avons également commencé à "relier" les classes fonctionnelles entre elles notamment en créant un "Moteur de course" qui permet de lancer des courses avec l'écurie du joueur contre des compétiteurs, dans un circuit basique.

Finalement un début de travail sur l'interface graphique a été réalisé. A ce jour, nous pouvons lancer un `main()` qui aboutit à l'affichage de l'écran d'accueil de notre jeu. Les boutons présents sur cet affichage sont fonctionnels et permettent progressivement de "jouer" à notre jeu de manière très simplifiée.

2.2 Détails sur le travail réalisé

Simon et Mathieu ont tout d'abord implémenté un "moteur de course" qui permet de lancer des courses et de récupérer grâce à un *map* les temps de course et performances associés à chaque pilote de la compétition. Pour cela des classes comportant une liste complète des pilotes et des écuries basiques ont également dû être réalisés par Simon. Le moteur de course a par la suite été testé par un programme de test et semble aujourd'hui bien fonctionner.

Ensuite, Sofiane et Bryan se sont attelés à la création de la classe `Classement`. Cette classe permet de générer un classement des pilotes suite à une course réalisée entre les pilotes de l'écurie du joueur et ceux des écuries adversaires. Cette classe ordonne les pilotes les uns par rapport aux autres selon leur temps de course. Sofiane et Bryan ont également réalisé un programme de tests afin de vérifier le bon fonctionnement de leur classe.

Mathieu et Martin se sont ensuite chargés d'implémenter les éléments de la voiture (`Roue`, `Frein`, `Châssis`) sur lesquels le joueur pourra agir en réalisant des améliorations de la voiture mais également pourront intervenir lors d'événements aléatoires au cours de la course. Ceux-ci modifieront la performance globale de la voiture et influera sur les temps de course de l'écurie. Les classes sont encore en cours d'implémentation même si la classe `Roue` est fonctionnelle et a été testé par un programme de test.

En parallèle du travail réalisé sur les classes majeures de l'application, Timothée et Yu-Chi ont démarré un travail sur l'interface graphique de l'application. En suivant un modèle passif `Vue-Contrôleur`, ils sont parvenus à implanter un début d'interface graphique simple mais fonctionnel. En lançant un `main()` on aboutit à l'affichage de l'écran d'accueil de notre jeu. Un bouton "Jouer" nous permet d'arriver à un menu Sauvegarde dans lequel le joueur peut créer, charger ou supprimer une partie de jeu. Pour le moment la création d'une partie résulte uniquement à l'écurie de base. Puis, en cliquant sur une partie déjà existante, l'utilisateur peut choisir d'améliorer sa voiture, ses pilotes ou bien lancer une course. Ces trois fonctionnalités ont été implanté et fonctionnent sur notre interface graphique. En achetant une amélioration de voiture, le budget de l'écurie est bien modifié. Le bouton "Lancer une course" affiche pour le moment le classement résultant de la course.

Bryan a également commencé à chercher des images pouvant nous servir pour le graphisme de notre interface. Nous les utiliserons lors de la prochaine itération.

Pour finir, Timothée a réalisé la sauvegarde d'une partie en utilisant la sérialisation. Cette sauvegarde est encore à tester.

2.3 Choix de conception et réalisation

Utilisation de la serialisation à la place de Json pour la sauvegarde

2.4 Problèmes rencontrés et solutions apportées

Pas de problèmes majeurs rencontrés lors de cette itération.

Troisième partie

Organisation de l'équipe et mise en oeuvre des méthodes agiles

1 Organisation de l'équipe

Lors de cette première itération, un grand travail d'organisation a été réalisé au sein de notre équipe. Une certaine rigueur et efficacité se sont rapidement mises en place. Les séances de TD de Méthodes Agiles nous ont notamment permis de longuement discuter et débattre sur les fonctionnalités que nous attendions tous dans notre application et ce travail nous a permis de rapidement procéder à l'écriture du code.

Une première grande réunion nous a permis de procéder à la construction de notre diagramme de classe, qui nous a servi de base pour l'implémentation de nos différentes classes.

Nous nous réunissons dès que nous le pouvons dans une salle informatique de l'ENSEEIHHT afin d'avancer le projet. Nous avons atteint une dynamique de groupe exemplaire et efficace et qui nous permettra, nous l'espérons, d'implanter une application correcte et répondant à nos attentes.

2 Mise en oeuvre des méthodes agiles

Nous avons tout d'abord créé un pitch en séance de méthodes agiles sur notre projet, ce qui nous a permis de nous assurer de la cible de notre produit et des particularités par rapport à d'autres jeux de F1.

Ensuite lors de la séance suivante, nous avons classé les fonctionnalités selon leur nécessité au fonctionnement du programme et l'enthousiasme du côté utilisateur, ainsi que selon le principe MOSCOW (Must, Should, Could, Won't).

De plus, nous avons fait de nombreuses réunions en présentiel et distanciel pour discuter des fonctionnalités, répartir les tâches, discuter de notre approche et proposer diverses solutions / modifications.

Nous avons également mis en place un document Excel où chacun peut noter les heures qu'il compte passer sur le projet pendant la semaine et ce sur quoi il compte travailler. Ainsi, nous avons également une réunion chaque semaine pendant laquelle nous nous répartissions le travail.