**Projet informatique : INSACAR**

**SOMMAIRE**

**INTRODUCTION**

Dans le cadre du projet informatique nous avons choisi de développer un jeu de voiture dans lequel un ou deux joueurs contrôlent une voiture sur un circuit en vue de dessus dans le but de faire le meilleur temps possible. De nombreuses fonctionnalités ont ainsi été imaginées comme, un déplacement de la caméra, un HUD, des collisions, etc.

L'objectif était de pouvoir exploiter ce que nous avions appris en algorithmique et sur le langage Pascal lors de notre première année, et de le combiner aux bases sur la SDL vues pendant ce semestre, afin de pouvoir obtenir un jeu que nous considérons satisfaisant autant en termes de graphisme qu'en termes de gameplay.

Nous avons rencontré et surmonté de nombreuses difficultés durant la conception de ce programme, sur lesquelles nous reviendront durant ce rapport.

Dans un premier temps, nous reprendrons le cahier des charges réalisé auparavant en faisant le bilan sur les fonctionnalités implémentées. Ensuite, nous traiterons de la conception globale avec l’évolution de l’analyse descendante et des structures de données utilisées. Puis, nous détaillerons le fonctionnement des différentes procédures et fonctions. Enfin, nous reviendrons sur notre expérience du travail de groupe lors de ce projet.

**I/ Cahier des charges**

Nous allons donc commencer par reprendre le cahier des charges que nous nous étions imposés ainsi que le calendrier des versions associées.

Fonctionnalités attendues :

* Menu principal
* Affichage tutoriel
* Consultation des scores (utilisation de fichiers).
* Configuration de sa partie (choix du circuit, de sa voiture).
* Mode contre-la-montre
* Mode 1 vs 1
* Contrôle voiture(s) grâce au clavier.
* Déplacement caméra à travers le circuit.
* Mouvements physiques de la voiture (déplacement, collisions, accélération, etc.)
* HUD (ensemble d’informations affichées à l’écran pendant la course (vitesse, tours, temps, etc.)).

Calendrier des versions prévues :

* Version n°1 / date prévue : mi-octobre

* Contrôle voiture grâce au clavier.
* Déplacement libre (sans collisions) de la voiture à travers un circuit
* Version n°2 / date prévue : mi-novembre
* Menu principal
* Affichage tutoriel
* Consultation des scores (utilisation de fichiers).
* Configuration de sa partie (choix du circuit et de sa voiture).
* Mode contre-la-montre
* Déplacement caméra à travers le circuit (mode contre-la-montre).
* HUD
* Version n°3 / date prévue : début décembre
* Plusieurs circuits
* Collisions
* Mode 1 vs 1
* Déplacement caméra à travers le circuit (mode 1vs1)
* HUD du joueur 2.

L’ensemble des fonctionnalités prévues ont été ajoutées à notre jeu. Le cahier des charges a donc été respecté.

Concernant le calendrier, il n’a pas été suivi dans son intégralité. En effet, si la version n°1 a été réalisée à temps, nous avons pris un peu de retard sur les deux autres.

Ce léger retard s’explique par le fait que nous avons implémenté l’ensemble des procédures et fonctions directement dans l’optique de réaliser un mode à 2 joueurs. Ce choix a finalement été judicieux car nous avons pu facilement adapter notre programme pour ce mode. La transition entre la version 2 et 3 s’est donc bien déroulée.

**II/ Conception globale** 1) Analyse descendante

Voici l’analyse descendante finale qui représente l’architecture de notre jeu.

ANALYSE DESCENDANTE

En observant l’analyse descendante réalisée en Septembre, on remarque que celle-ci n’a pas subi de profonds changements. En effet, l’architecture globale de notre jeu n’a pas changé. Nous avons surtout effectué des modifications sur les structures de données (voir ci-après) et sur les entrées-sorties de nos procédures et fonctions. De plus, les seules fonctions qui ont été ajoutées sont des outils (unit tools) permettant de faciliter l’implémentation et d’améliorer la clarté du code (exemple : procédures ajouter\_enfant, ajouter\_physique, updatePseudo, etc.).

2) Structures de données

**III/ Détail des procédures et fonctions**

**IV/ Travail en groupe et utilisation d’un logiciel de gestion de versions**

Au cours de ce projet, nous nous sommes assez bien réparti le travail selon les capacités de chacun. De plus, nous nous sommes souvent réunis en salle informatique pour travailler et avancer ensemble. Cet aspect du projet s’est bien déroulé.

Le point le plus difficile a été de coordonner les différentes versions de chacun. En effet, travailler à plusieurs sur le même code source peut rapidement mener à des problèmes comme des modifications qui écrasent celle d’une autre personne par exemple.

Pour éviter cela, nous avons décidé d’utiliser un logiciel de gestion de version. Ces logiciels permettent de travailler à plusieurs sans risquer de perdre des informations si deux personnes travaillent en même temps sur un même fichier. De plus, les logiciels de gestion de version permettent de suivre l’évolution d’un code source à chaque modification. Cela peut permettre de revenir en arrière dans les versions en cas de problème par exemple.

Nous avons donc utilisé « Git » un des logiciels de gestion de versions les plus connus. Ce dernier a été difficile à prendre en main au début mais nous avons vite observé son intérêt dans le projet.

Finalement, cet outil a grandement facilité la collaboration dans notre projet et sans lui il aurait été très difficile de travailler efficacement en groupe.

**CONCLUSION**

Pour conclure, ce projet informatique nous a permis d’approfondir et mettre en pratique nos connaissances en Pascal et plus généralement en algorithmique.

De plus, ce projet aura été l’occasion de découvrir et prendre en main la bibliothèque SDL permettant de réaliser des jeux 2D assez satisfaisants en termes de graphiques.

Enfin, nous avons pu d’expérimenter le travail de groupe sur un projet informatique et apprendre des méthodes pour travailler efficacement avec notamment l’utilisation d’un logiciel de gestion de version.