

# YoloCraft

## Présentation du projet :

Cette application nous a été demandée par notre université dans le cadre du cours LIFAP4 Conception et Développement d'Application.

## Description synthétique :

Ce jeu consiste en un parcours d'une carte 2D (de profils).

Le joueur est muni d'un véhicule. Son but est d'atteindre la fin de la carte en se déplaçant de gauche à droite.

## Description détaillée :

L'utilisateur arrivera face à un écran de niveaux.

Il aura donc accès à son véhicule, via son inventaire, qui sera préalablement défini. Dans cet inventaire, le joueur pourra modifier la structure de son véhicule ainsi que le munir d'arme pour se défendre face aux ennemis se présentant devant lui.

Il aura aussi accès un magasin, lui permettant d'acheter des éléments à ajouter à son véhicule, le rendant plus robuste, plus stable, ou plus destructeur selon l'élément ajouté. A l'inverse, chaque élément ajouté possède un poids, ralentissant d'avantage le véhicule.

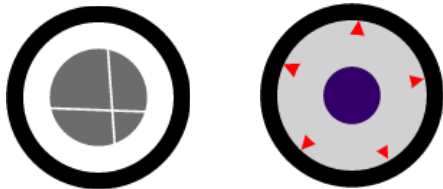
Le but du jeu est donc de jouer entre la destructivité du véhicule face aux ennemis, sa stabilité face au relief de la carte, et son poids pour pouvoir avancer de manière correcte. Si jamais le véhicule est trop lourd, le joueur pourrait ne pas faire face à certaines montées de la carte.

De plus, une liste de niveaux sera accessible et se débloquent en fonction de l'avancée du joueur.

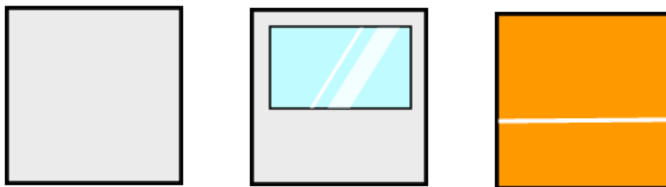
Afin de pouvoir réaliser ses achats dans le magasin, le joueur sera récompensé par une monnaie à la fin de chaque partie selon le comportement du joueur lors de cette partie (distance parcourue, ennemis abattus, bonus gagnés ...).

Les éléments de véhicules achetable sont divisés en 3 catégories pour 3 comportements différents :

- Roues



- Carcasse



- Tourelles (armes)  
(pas d'image disponible pour l'instant)

### Règles exactes :

Le joueur devra atteindre la fin de la carte.

Pour se faire il devra abattre tous les ennemis qui se trouveront sur sa route.

Dans ce même temps, il a intérêt à ramasser les pièces qui se trouveront devant lui afin de les réutiliser pour acheter des pièces afin d'améliorer son véhicule.

Le jeu n'a pas de « fin » car ses limites sont définies par le niveau d'amélioration de la voiture souhaité par le joueur.

Pour jouer, le joueur disposera de son clavier qui lui permettra de déplacer son véhicule et de sa souris pour tirer en cliquant sur sa cible.

### Contraintes :

Le projet ne doit pas excéder le 1<sup>er</sup> Mai, et aucun budget ne nous est attribué pour le développement donc cette application n'aura aucun coût monétaire.

Nous sommes 4 développeurs sur ce projet.

Nous sommes contraints au langage C++, ainsi que la bibliothèque SDL2 pour l'interface homme-machine.

## Déroulement du développement :

Nous avons découpé le développement de l'application en N étapes.

### Étape 1 : Structure de l'application

Créer les différentes fenêtres d'applications même si elles ne sont pas fonctionnelles.

**Livable** : Une coquille applicative non fonctionnelle.

### Étape 2 : Implémentation du Gameplay (section de jeu)

Créer un véhicule pré-établi et instancier une carte sur lesquels le paramétrage du moteur physique se fera.

2.1 : Définition des sprites

2.2 : Mise en place du moteur physique

2.3 : Liaison entre le véhicule et le moteur physique

**Livable** : Une première carte et un premier véhicule accessibles et utilisables.

### Étape 3 : Implémentation du garage (section de personnalisation du véhicule)

Créer une interface de customisation sous forme de grille permettant l'ajout des pièces achetées dans le magasin au véhicule.

3.1 : Mise en place d'un système de stockage des pièces acquises.

3.2 : Mise en place de la sauvegarde du véhicule dans un fichier de données.

3.3 : Mise en place de la personnalisation du véhicule.

3.4 : Mise en place de l'interactivité de l'IHM avec la grille de personnalisation.

**Livable** : Un module de grille interactif où il est possible de poser et reprendre des éléments.

### Étape 4 : Implémentation du magasin (section d'achat de pièces pour le véhicule)

Créer une interface où des pièces de différentes caractéristiques pourront être achetées grâce aux pièces gagnées par partie.

4.1 : Définition des pièces dites « achetable ».

4.2 : Mise en place du système d'achat.

4.3 : Liaison entre le magasin et le garage.

**Livable** : Un module d'achat.

Étape 5 : Implémentation d'autres niveaux et de l'interface de choix du niveau de jeu.

Créer une liste de niveaux accessibles via l'interface menu permettant au joueur de revenir sur des niveaux déjà finis.

**Livable** : Une interface de choix pour le niveau à jouer.

Étape 6 : Implémentation d'IA génériques.

Implémentation d'un système d'intelligence artificielle pour contrôler les ennemis.

6.1 : Définition des véhicules ennemis

6.2 : Définition d'un système de contrôle des ennemis.

6.3 : Mise en place d'une génération d'ennemis.

6.4 : Placement des différents ennemis et du système de contrôle.

**Livable** : Un système de gestion des ennemis.

Étape 7: Finalisation de l'interface graphique

Amélioration graphique de l'application, et mise en place des sprites adaptés selon les boutons/icônes nécessaires pour l'IHM.

**Livable** : Une IHM de meilleure qualité.

### Bonus :

Si nous avons suffisamment de temps restant, nous prévoyons d'implémenter différentes fonctionnalités supplémentaires.

- Un système de boîtes « bonus » :

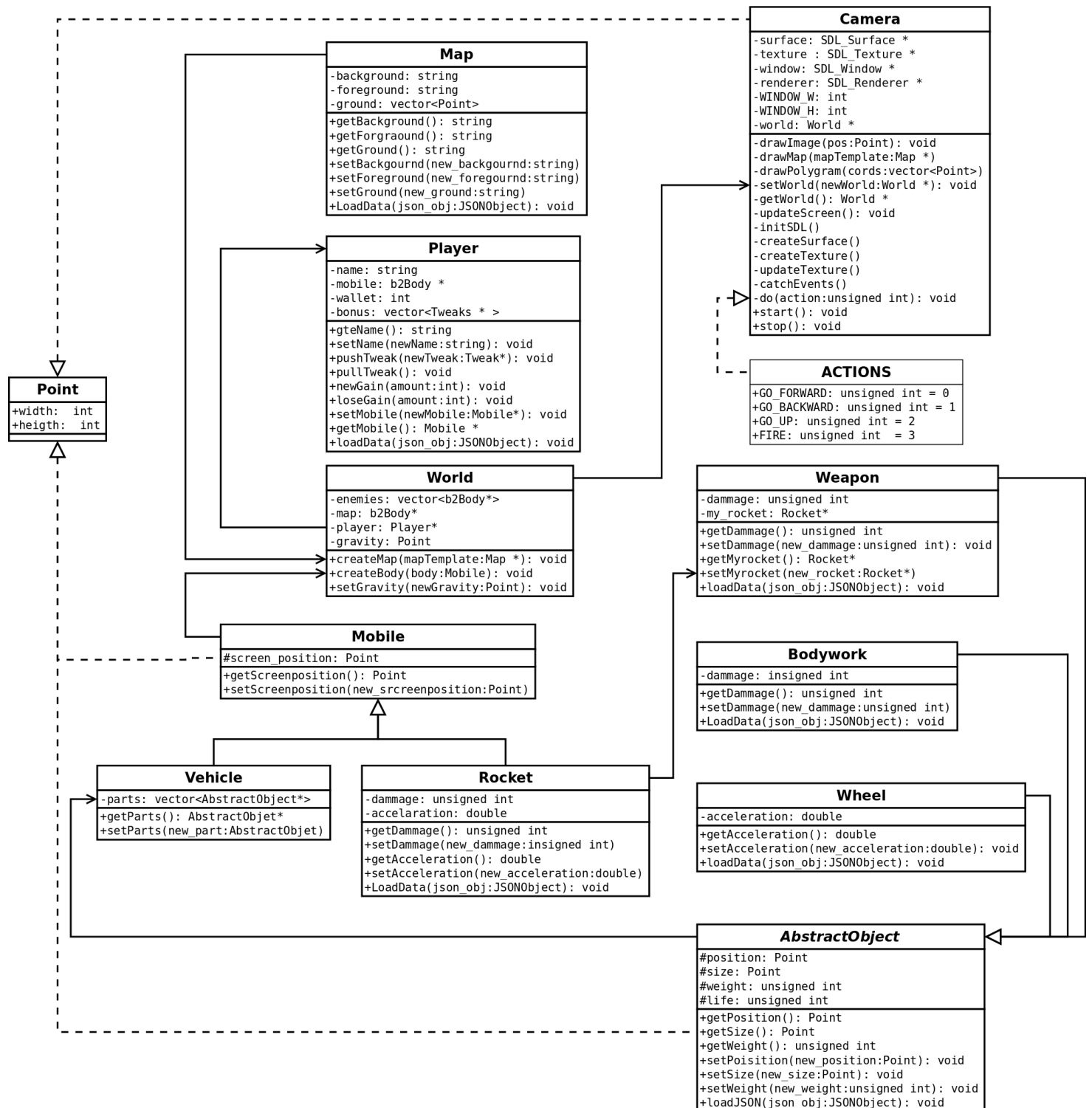
C'est l'implémentation de bonus aléatoires se trouvant sur la carte de manière récurrente grâce auxquelles le joueur obtiendra un avantage sur son véhicule de manière temporaire.

Exemple : Des tirs plus puissants, une vitesse accrue ...

- Des « Cheat » Code :

Un système capturant les entrées claviers lors de parties permettant au joueur de tricher en acquérant des avantages définitifs pour le reste de la partie.

Exemple : Invincibilité, Tirs continus ...



## Diagramme des Gantt:

