

Rapport PPII - Groupe 28  
Application de jeu Crossy Road avec IA : Saulcy Road

Membres du groupe : ALLOMBERT-BLAISE Oscar - FALUS Sophie - KOBLER Thibault - RAMBEAUX Erwann

## Principe du jeu :

Le jeu que nous avons créé réplique le gameplay et les règles du célèbre jeu Crossy Road. Le principe est très simple : faire avancer le personnage sur un plateau de jeu le plus loin possible malgré de multiples obstacles sur son chemin. A chaque pas fait vers l'avant, le score est incrémenté de 1. Le meilleur score est sauvegardé, pour motiver le joueur à le battre à chaque partie.

Le personnage peut avancer, reculer, ou faire des pas sur le côté gauche / droit.

Certains obstacles sont mortels : les voitures / camions, les trains et l'eau. D'autres servent uniquement à ralentir le jeu et à rajouter de la difficulté dans sa progression : buissons et arbres.

Le plateau est constitué de plusieurs bandes, qui sont alternées de manière aléatoire à chaque partie pour avoir toujours un plateau différent. Il y a 4 types de bandes différentes : la rivière, avec des nénuphars / des planches pour se déplacer, la route avec les véhicules, la prairie avec les buissons / rochers, et les rails avec le train.

Notre jeu possède également un onglet "Vestiaire", qui permet de choisir l'apparence du personnage : Poulet, Canard ou Crocodile.

## Modes de jeu :

Le jeu possède deux versions différentes : une version textuelle et une version graphique.

La version textuelle est une version visuellement simplifiée du jeu qui se joue dans le terminal. L'utilisateur peut choisir de contrôler le personnage en utilisant les touches ZQSD, ou les flèches. Le personnage est représenté par le symbole \*, les véhicules par > ou <, les rails par =, l'eau par ~, etc.



La version graphique est une version visuellement plus avancée en 2D. Le principe de jeu reste le même, mais l'affichage se fait sur une interface graphique.



Dans chacune des versions, le joueur peut faire le choix de jouer par lui-même, ou alors d'utiliser le "mode IA", qui va remplacer le joueur par un joueur virtuel, qui va jouer à sa place, grâce à un algorithme d'intelligence artificielle.

## Choix techniques pour l'implémentation :

### Représentation des données :

Avant de commencer le développement, lors de la première réunion, nous avons fait un brainstorming concernant la manière dont nous allions représenter les données dans notre code, puis avons réalisé un petit schéma :



Celui-ci a connu de multiples changements lors du développement, mais il nous a aidé à avoir une vision globale sur le schéma de données que nous allions implémenter.

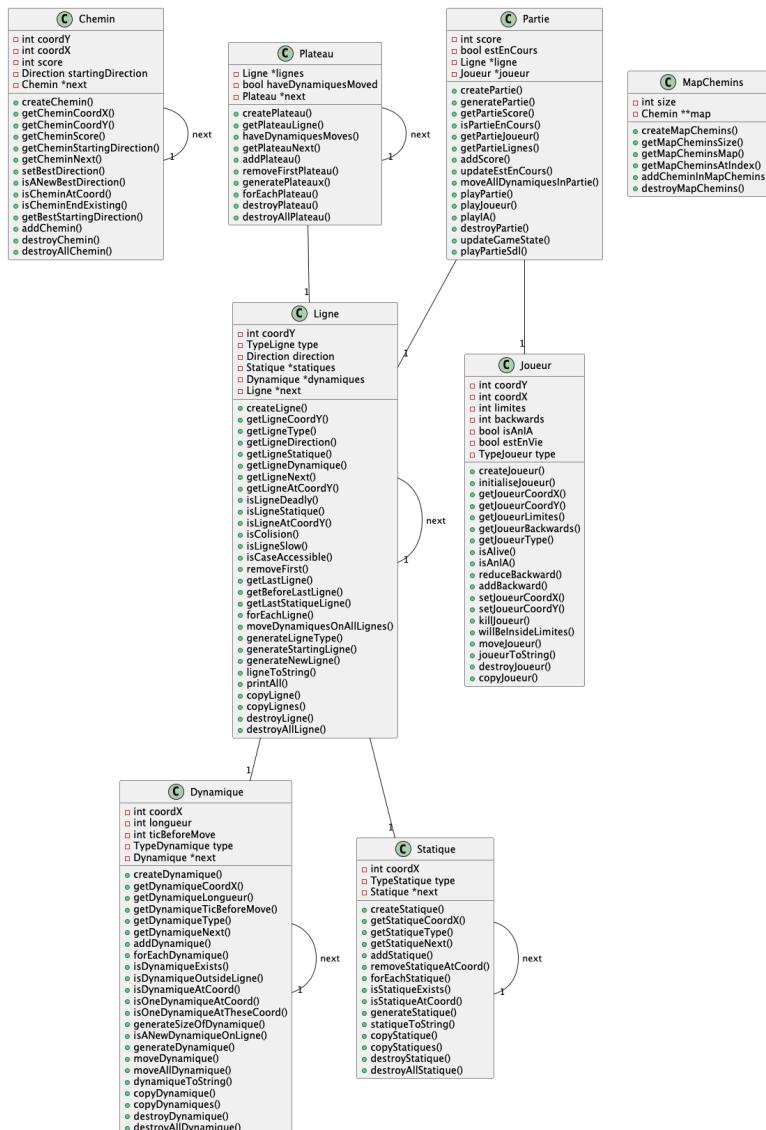
Plusieurs structures sont donc présentes dans notre code : le type ligne codé en liste chaîné. Les lignes ont un type (route, herbe, etc.), sont composées d'éléments (statiques et dynamiques), et connaissent également de manière récursive la ligne adjacente.

Les éléments statiques ont un type (buisson, arbre, nénuphar), des coordonnées en x et connaissent également le prochain élément présent. Les éléments dynamiques fonctionnent de la même manière, mais par exemple avec une notion de longueur en plus et des types différents (voiture, camion, train et bûche).

Nous avons aussi fait une structure pour la partie, qui est constituée d'un score, de lignes et d'un joueur. Le joueur possède des coordonnées x / y, des booléens pour savoir si c'est une IA / si il est encore en vie, etc. De plus, les futurs "plateaux" sont générés et permettent à l'IA d'anticiper à l'avance avec une génération des dynamiques en dehors de la carte.

Enfin, afin d'implémenter le joueur virtuel avec l'IA, nous avons créé une structure "chemin" qui permet de stocker les différents chemins que le joueur peut emprunter afin de les comparer et de trouver le meilleur.

Nous avons donc l'UML final :



### Réalisation du moteur de jeu, la partie “Core” :

Comme le sujet l'exige, le jeu est implémenté dans le langage C. Nous avons commencé par définir dans des fichiers .h toutes les structures détaillées juste au dessus, puis à coder leur logique. Le noyau de la logique du jeu / du moteur est dans le fichier partie.c. La fonction playJoueur appelle une multitude de fonctions provenant de partie et des autres structures, et génère le plateau, initialise le joueur, déplace les objets, gère les mouvements du joueur et ses potentielles collisions avec des obstacles, et se charge du score. Grâce à la définition des structures dans différents fichiers, la modularité que cela nous a apporté à permis de très facilement adapter le jeu au joueur virtuel IA (fonction playIA).

La partie textuelle a été réalisée via des instructions de print dans des fonctions présentes dans ces structures.

Pour faciliter la compilation, l'exécution et le support du jeu sur différentes machines, nous avons fait un Makefile avec une multitude de directives et en utilisant également pkg-config.

### Réalisation de la partie graphique :

Pour réaliser la partie graphique du jeu, nous avons utilisé la librairie C libSDL. Elle permet de facilement dessiner de multiples éléments dans une interface graphique, de gérer les interactions avec le joueur (clavier / souris), de jouer des sons / de la musique, etc...



Avant de commencer à coder ce mode graphique, nous avons dû d'abord dessiner et récupérer différents sprites (les éléments graphiques du jeu). Certains sprites (comme celui du joueur) demandent même de réaliser une “spritesheet”, qui permet de modéliser les mouvements de manière animée.

Une fois les sprites réalisés, nous avons implémenté la logique de l'interface graphique dans le code. Cela se fait majoritairement via un fichier renderer.c, qui se charge de faire l'initialisation de la fenêtre, des ressources, de l'audio, et charger les textures. Il permet également de dessiner le plateau, le joueur, les éléments statiques / dynamiques, de jouer les effets sonores et la musique, et de gérer les entrées de l'utilisateur.

C'est également ici que le menu principal du jeu ainsi que la fonctionnalité “vestiaire” (qui permet au joueur de choisir son apparence) sont implémentés.

Enfin, quand le jeu est quitté ou qu'il y a un problème qui cause son arrêt, nous nous assurons de bien libérer en mémoire toutes les ressources qui ont été allouées.

#### Algorithme d'intelligence artificielle pour le joueur virtuel :

Pour nos choix techniques concernant l'algorithme d'IA, nous avons précisé nos choix et le fonctionnement de notre algorithme dans l'état de l'art, également présent dans ce dépôt git.

#### Gestion de projet :



Pour gérer la planification du projet, les différentes tâches et leur attribution, nous avons utilisé Asana. C'est un outil en ligne simple d'utilisation, qui permet de facilement visualiser les tâches restantes et leurs deadlines.

Avant de commencer le projet, nous avons réalisé une fiche projet, permettant de définir les attendus, les technologies qu'on va pouvoir utiliser, etc. Nous avons également fait une étude du jeu Crossy Road afin de bien comprendre son fonctionnement, ses règles et les déplacements des différents éléments.

Durant toute la réalisation du projet, nous avons rédigé des comptes rendus après chaque réunion, afin de synthétiser les avancées, les discussions et les tâches à réaliser à chaque étape de l'implémentation.

#### Répartition des tâches :

Membre du groupe	Tâches réalisées
Erwann	<ul style="list-style-type: none"><li>- Contribution à la conception de la représentation des données</li><li>- Développement de la partie graphique du jeu</li></ul>

	<ul style="list-style-type: none"> <li>- Réalisation de sprites</li> <li>- Ecriture de tests unitaires</li> </ul>
Oscar	<ul style="list-style-type: none"> <li>- Contribution à la conception de la représentation des données</li> <li>- Etat de l'art + étude du fonctionnement de Crossy Road</li> <li>- Gestion de projet : Asana, Fiche projet, comptes-rendus de réunion, etc.</li> <li>- Développement de la fonctionnalité vestiaire et du menu principal.</li> <li>- Réalisation de sprites</li> </ul>
Sophie	<ul style="list-style-type: none"> <li>- Contribution à la conception de la représentation des données</li> <li>- Etat de l'art + étude du fonctionnement de Crossy Road</li> <li>- Gestion de projet</li> <li>- Développement de la fonctionnalité vestiaire</li> </ul>
Thibault	<ul style="list-style-type: none"> <li>- Contribution à la conception de la représentation des données</li> <li>- Développement de la partie "Core"</li> <li>- Développement de la partie textuelle du jeu</li> <li>- Développement de l'algorithme d'IA</li> <li>- Ecriture de tests unitaires</li> </ul>

## Analyse Post-Mortem :

### Atteinte des objectifs :

Objectif	Niveau d'atteinte
Qualité de la modélisation et des structures de données utilisées	<b>Complètement atteint</b> Les structures de données implémentées permettent de modéliser correctement le jeu et ses différents éléments.
Implémentation correcte de la logique du jeu et respect des règles de Crossy Road	<b>Complètement atteint</b> La logique du jeu est implémentée correctement, en suivant les règles principales de Crossy Road.
Structure du code propre, et respect des bonnes pratiques de programmation	<b>Complètement atteint</b> Commentaires, structures de données dans plusieurs fichiers, compilation séparée, Makefile, etc.

Fonctionnalité de l'intelligence artificielle	<b>Partiellement atteint</b> Le joueur virtuel est quasiment invincible sur la version textuelle, mais est moins performant sur la version graphique.
Qualité de l'interface utilisateur et fluidité des interactions avec le jeu (version textuelle et version graphique)	<b>Complètement atteint</b> La version graphique est fluide et pertinente par rapport à Crossy Road. Les interactions avec le joueur sont également fluides, dans les deux versions du jeu.
Réalisation de tests unitaires qui englobent tout le périmètre du jeu et qui assurent sa robustesse	<b>Complètement atteint</b> Présence de tests unitaires qui couvrent l'intégralité du jeu.
Gestion de projet durant toute la durée du projet, documentée.	<b>Complètement atteint</b> Réalisation d'une fiche de projet, de comptes-rendus de réunion, d'une maquette, et utilisation d'Asana pour planifier et répartir les tâches.

#### Commentaires personnels :

Erwann : "J'ai trouvé ce projet globalement sympathique, notamment parce qu'il nous a permis de mener à bien un projet complet et de mobiliser les différentes compétences acquises. En revanche, j'ai trouvé l'aspect créatif assez peu stimulant : le fait de devoir reproduire un jeu déjà recréé des centaines de fois limitait l'originalité du projet et réduisait un peu ma motivation. D'un point de vue technique, le projet m'a aussi paru relativement simple, en raison du manque d'innovation nécessaire à sa réalisation. Cela dit, le travail en équipe s'est très bien passé, avec une communication fluide et efficace dès que le besoin se faisait sentir."

Oscar : « Ce projet m'a permis de découvrir de nouvelles choses, comme la réalisation de sprites, les différentes possibilités pour implémenter une IA dans un jeu ou encore la libSDL. C'est également toujours bien de pouvoir appliquer les notions de gestion de projet que nous avons apprises dans un cadre concret. La collaboration et la communication dans le groupe s'est bien passée. »

Sophie : « Ce projet m'a permis de devenir à l'aise avec le C, et d'en apprendre sur les différents algorithmes d'IA. Au départ, c'était plutôt difficile de comprendre comment marche la libSDL mais une fois que j'ai compris j'étais contente. Il n'y avait pas de soucis avec la gestion de projet, on communiquait bien et sans conflit.»

Cependant, j'aurais aimé plus coder, en touchant à plusieurs aspects du jeu. »

Thibault : “ Participer à ce projet a été une réelle opportunité de mettre en pratique mes connaissances en langage C, tout en les approfondissant. J'ai particulièrement apprécié les aspects techniques comme la mise en place de structures de données optimisées au contexte et le développement d'une IA à la fois complète et complexe. De plus, la collaboration dans le groupe s'est bien passée.

En revanche, le format du projet en groupe de quatre a parfois limité l'implication sur l'ensemble des aspects du projet, tout le monde ne peut pas forcément toucher à tout ce que je trouve dommage et peut freiner l'apprentissage global. Pour un projet comme celui-là, un travail en binôme aurait peut-être été plus formateur pour chacun.”