

Cahier des charges

1 - Contexte et définition du projet

Il s'agit de réaliser notre propre version du jeu « Crossy Roads » permettant à la fois à une IA qu'à un joueur de jouer une partie. Cette version devra se décliner sous deux éditions que nous nommerons « core » et « ui ». La première étant une implémentation se jouant intégralement dans le terminal, que nous pouvons être amené à nommer « CLI » et la seconde se jouant de la même manière mais via une interface graphique utilisant la librairie [libSDL](#).

Cette simulation s'appuie sur une grille générée dynamiquement sur laquelle un joueur incarnant un personnage, ici un goblin avec une cape, doit éviter des obstacles tout en essayant d'atteindre le plus haut score possible. Le jeu s'articule autour de plusieurs types de lignes, créant ainsi un certain dynamisme (routes, rivières, rails, glaces), chacune ayant un comportement qui lui est propre (voitures, plateformes de bois et trains). La difficulté du jeu doit s'avérer être croissante, on pourra la remarquer à la vitesse et la densité des voitures ou encore la diminution de la taille des plateformes de bois.

Ce jeu s'appelle « **Goby Roads** », le but étant d'aider un goblin à rentrer dans son monde.

2 - Objectif du projet

Ce projet a pour but de nous permettre de consolider « nos compétences en développement C, en implémentation de structures de données, en utilisation d'une librairie graphique externe (SDL2), en intelligence artificielle et en gestion de projet, tout en étant un défi ludique et pratique avec la conception d'un jeu ».

Il met aussi en pratique dans une certaine mesure nos compétences en communication étant donné qu'il s'agit ici d'un travail de groupe.

3 - Périmètre du projet

Outre les technologies qui sont expressément inscrites, nous avons une certaine liberté quant à la réalisation de ce projet. Il ne faut cependant pas se perdre dans les conjectures et se concentrer sur la production d'un livrable dans les temps impartis.

4 - Description fonctionnelle des besoins

Les déplacements du joueur au clavier doivent se faire avec les touches z, q, s, d sur l'édition « Core » et avec ces mêmes touches auxquelles on ajoutera les touches fléchées sur l'édition « UI » en respectant les collisions et les règles de chaque terrain.

La grille doit être générée de manière procédurale avec des lignes de type aléatoire mais contrôlé avec des probabilités fixes : 50% de chance d'avoir une route, 40% pour les rivières et 10% pour les rails.

Le score quant à lui doit être géré en se basant sur la progression verticale du joueur, il s'incrémente dès la première ligne atteinte et ne doit jamais être décrémenté.

Un système de collisions entraînant la défaite doit être mis en place, détectant les chocs avec les voitures, les trains ou encore les chutes dans l'eau.

Pour l'édition « UI » des sons doivent être ajoutés pour renforcer l'immersion, ces sons se déclinent en deux différents types : la musique de fond (lorsque l'on joue, perd) et les effets sonores (SFX) joués notamment lors de collisions précédemment mentionnées mais aussi lorsqu'un train est annoncé et lorsqu'il passe.

Cette édition doit aussi inclure différents écrans, un menu permettant de choisir si l'on veut jouer à l'édition « Core » du jeu dans le terminal, si au contraire on aimerait avoir accès à la surcouche graphique obtenue grâce à SDL2 et enfin si on souhaite que l'IA joue à notre place, ces options ne sont pas exclusives et peuvent toutes être sélectionnées en même temps, il y aurait alors un jeu affiché à la fois dans le terminal et dans une fenêtre SDL qui sera joué par l'IA.

Enfin, des tests unitaires doivent être mis en place pour assurer la fiabilité du système, ces tests unitaires porteront exclusivement sur l'édition « Core » du jeu étant donné que l'édition « UI » n'est qu'une surcouche de « Core » : elle ne redéfinit pas les concepts définis, elle les encapsule si nécessaire. Ces tests unitaires se feront avec le framework Unity Testing et doivent être aussi complet que possible avec une couverture élevée.

5 - Délais de réalisation

La date de livraison du projet est fixée au 02/06/2025. Une soutenance est prévue pour le 05/06/2025 à 7h30. Le projet doit être livré avec son code source, un état de l'art sur les algorithmes d'intelligence artificielle applicables et appliqués, la documentation du projet, et les tests unitaires. Ils doivent être présents dans le dépôt GitLab du projet.