

Langage C : Projet Final – Rapport du projet :

A. Comment lancer votre projet ? Quelles sont ses dépendances ?

1. Lancement du projet :

Pour installer notre projet de UNO en langage C, il faudra tout d'abord cloner le repo git à l'aide de la commande :

```
git clone https://github.com/romainlancelot/uno.git
```

Ensuite, afin de compiler le code, il faudra vous rendre dans l'arborescence dans le répertoire « src » à l'aide la commande :

```
cd src
```

Faites ensuite un « make »

```
make
```

Ensuite, sélectionnez et exécutez le fichier « uno »

```
./uno
```

2. Dépendances :

Notre projet n'admet aucune dépendance, mis à part celles aux librairies installées nativement sur Linux.

Notre SDL n'étant pas encore fonctionnel à 100%, nous n'avons pas encore besoin des librairies SDL2 pour lancer le jeu.

B. Le contexte et l'organisation de votre groupe pour répondre à la problématique : avez-vous établi un planning ? Qui a fait quoi et pourquoi ?

1. Planning :

Afin de nous organiser et de pouvoir répondre à la problématique dans les délais, nous avons initialement prévu un planning :

Date prévue	Etapes	Date effective
5 décembre	Créer les assets	5 décembre
12 décembre	Distribution des cartes	12 décembre
19 décembre	Système de défausse des cartes	12 décembre
26 décembre	Règles générales	2 janvier
2 janvier	SDL (partie 1)	En cours
9 janvier	SDL (partie 2 (au cas où))	En cours
16 janvier	Stocker les scores dans un fichier	En cours
23 janvier	Création du bot	12 février
30 janvier	Mise en réseau du jeu	En cours
5 février	Bonus (si on a le temps)	En cours
12 février	Finalisation du rendu final	19 février

Ce planning ne prenait pas suffisamment en compte les délais supplémentaires dont nous avons eu besoin pour certaines étapes au cours desquelles nous avons rencontré des difficultés.

Néanmoins, nous avons prévu des dates suffisamment larges dans le temps afin que puissions rebondir à temps pour que les principales fonctionnalités du projet soient prêtes pour le 24 février.

2. Distribution des tâches entre les membres du groupe :

Nous n'avions pas précisément prévu de distribuer les tâches entre les différents membres du groupes. Nous souhaitions avancer sur le projet en simultané, et c'est ce qui a été fait. Si nous devons forcément noter qui a précisément travaillé sur quelle partie du projet, voila à quoi cela ressemblerait :

- Création des assets de cartes : Julien, Romain et Ayman
- Système de distribution de cartes : Romain et Ayman
- Système de défausse des cartes : Julien et Ayman
- Système de dépose des carte sur le tapis : Julien
- Règles générales du UNO : Romain et Julien
- Gestion des couleurs : Romain et Julien
- Effets des cartes : Julien, Romain et Ayman
- Mise en forme d'une partie : Romain
- Menu d'affichage : Romain
- Affichage général : Romain, Julien et Ayman
- Gestion du nombre de joueurs : Romain et Ayman
- Gestion des actions des joueurs lors d'une partie : Romain et Julien
- Création du bot : Romain et Julien
- Gestion du bot : Romain
- Découpage du code en modules : Romain et Ayman
- Makefile : Romain
- Connexion en réseau (en cours) : Romain
- SDL (en cours) : Ayman et Romain
- Stockage des scores dans le fichier (en cours) : Romain

Dans l'ensemble, les tâches ont réellement été réparties, néanmoins Romain est celui qui a le plus fait avancer le projet, du à son niveau plus avancé en langage C par rapport à ceux de Julien et d'Ayman. Romain est venu débloquent le code à chaque fois que notre groupe bloquait sur un point.

C. Pourquoi avez-vous organisé le code de cette manière ? Quels sont les éléments du cours ou du langage C que vous avez utilisé pour le mener à bien ? Ces choix ont-ils été pertinent pour avancer plus rapidement dans votre code, garder en maintenabilité ?

1. Organisation du code :

Le code a été découpé en 4 modules (+2 en comptant la partie réseau toujours en développement) :

- Cards : Ce module contient diverses fonctions, notamment celle permettant la création des cartes en début de partie, celle qui permet de réapprovisionner la pioche avec les cartes dans la défausse lorsqu'elle est vide, celle qui permet de supprimer une carte piochée de la pioche, celle qui transfère la carte présente sur le tapis dans la défausse à

chaque tour, et enfin la fonction qui permet de poser une carte choisie par un joueur sur le tapis.

- Game : Ce module contient tous les paramètres d’affichage du jeu, mais également les divers effets des cartes spéciales (inversion, skip, +2...) et la gestion du nombre de joueurs.
- Main : Dans ce module, nous ne retrouvons que le strict minimum permettant de lancer une partie, c’est-à-dire l’affichage du menu de départ, duquel le jeu peut être lancé.
- Players : Ce module contient les fonctions permettant de tier les cartes des joueurs, celle permettant de piocher une carte de façon aléatoire, la vérification de victoire probable d’un joueur.
- Server : Module en cours, contenant les prérequis pour héberger une partie en réseau.
- Client : Module en cours, contenant les prérequis pour rejoindre une partie en réseau.

2. Elements du cours utilisés :

Afin de pouvoir mener à bien ce projet, nous avons eu recours au support de cours et aux différents chapitres. Les divers chapitres particulièrement utiles du cours ont été :

- Chapitre sur les structures (création des cartes, des joueurs...).
- Chapitre sur les pointeurs (récupération des cartes dans les fonctions...).
- Chapitre sur la programmation modulaire : découpe du projet en modules, comme décrit dans le point C.1.

3. Pertinence de ces choix :

Ces choix ont été pertinents car ils nous ont permis de mieux nous y retrouver au cours de l’avancement du projet.

Néanmoins, la découpe en module a aussi engendré quelques pertes de temps lorsque nous avons du coder le bot, puisque notre code préalablement écrit sur lequel nous souhaitions nous baser pour gérer le système de « joueur » robot était dispatché entre différents modules. Si nous devions tout recommencer à zéro, je pense que nous intégrerions directement notre code en module.

D. Quelles fonctionnalités proposez-vous dans ce projet ? Comment les avez-vous optimisées ? Valorisez l’apport de vos connaissances en algorithmique et leur intérêt pour ce projet en langage C.

Les fonctionnalités apportées dans ce projet sont toutes celles que nous pouvons attendre d’un jeu de UNO classique.

En plus de celles-ci, la possibilité de jouer en solo contre l’ordinateur est un réel plus, puisque cela n’est pas possible dans la vie réelle. La possibilité de jouer en multijoueur à distance (toujours en cours de développement) est aussi un point positif, puisque ça nous permet de jouer au jeu à distance avec d’autres personnes.

Apport en connaissance et en intérêt pour le projet :

Romain : « J’ai pu perfectionner ce que je savais déjà et découvrir la gestion du réseau en langage C. Mes connaissances m’ont permis de porter le groupe lorsque nous rencontrions des difficultés. Ce

projet m'a plu mais je suis légèrement déçu de ne pas avoir pu terminer à temps tout ce que nous souhaitions faire. ».

Ayman : « Ce projet de UNO m'a permis de mieux comprendre le langage C en pratiquant sur un travail concret dont je voyais la progression et la concrétisation au fur et à mesure que celui-ci avançait. J'ai notamment pu découvrir le SDL, ce qui m'a beaucoup plu. ».

Julien : « J'ai de grandes difficultés à apprécier ce langage, et j'avais eu du mal à m'y intéresser avant ce projet, car je n'en voyais pas bien l'intérêt. Faire ce projet m'a permis de me rendre compte dès le début de mes lacunes en langage C. Cela m'a poussé à reprendre le cours depuis le début, et j'estime avoir fait d'énormes progrès, et d'être fier de ce que nous avons réalisé. ».

E. Avez-vous rencontré des difficultés techniques, organisationnelles, relationnelles pour réaliser ce projet ? Comment les avez-vous dépassées ?

1. Difficultés techniques :

Les principales difficultés techniques que nous avons eues sont liées à l'installation de SDL et à la mise en réseau. Nous sommes toujours en train de les dépasser à l'heure actuelle.

2. Difficultés organisationnelles et relationnelles :

Comme précisé en B.1., nous n'avons pas pu respecté les dates butoirs que nous nous étions fixées pour chaque étapes du projet en amont. Néanmoins, nous avons su réagir et nous organiser en conséquence. Nous n'avons eu aucune difficultés relationnelles, nous parlions beaucoup entre nous du projet, que ce soit en classe ou lors d'appels sur discord.

F. La concordance du projet avec votre formation. Quelles sont les aptitudes qui ont été requises de vous et comment votre formation à l'ESGI (ce cours ou d'autres) vous a aidé ou fait défaut ? Avez-vous aimé réaliser ce projet ? Pourquoi ?

- Romain : « Les aptitudes requises par ce projet sont celles que nous vues en cours. J'ai été très polyvalent lors de l'aboutissement de ce projet et cela m'a permis de m'améliorer en langage C, mais également dans ma gestion des imprévus. J'ai beaucoup aimé réalisé ce projet, car nous pouvions choisir le sujet, et cela m'a motivé de voir l'avancement concret étape par étape. ».
- Ayman : « J'ai dû rapidement m'habituer au rythme de travail de l'ESGI. Mes connaissances en algorithmie m'ont permis d'avoir une vision d'ensemble du projet, mais le projet m'a aussi aidé à mieux comprendre le C. J'ai apprécié réaliser ce travail car nous étions en équipe et que le sujet me plaisait. ».
- Julien : « Le support de cours m'a été d'une grande aide pour ce projet, car je n'avais pas vraiment réussi à suivre le rythme des cours (j'avais l'impression que nous avions tout très vite en cours au début d'année, et cela m'avait fait prendre beaucoup de retard, j'avais perdu ma motivation dans cette matière.). Le UNO que nous avons fait m'a beaucoup plu, car j'ai vraiment beaucoup appris au cours de ces 3 derniers mois, et j'ai

vu directement l'évolution de mon niveau en langage C. Merci également pour le support de cours qui explique tout à la perfection, et qui m'a énormément aidé chaque fois que j'en ai eu besoin. ».