

Rapport de Projet Uno

Anthony Boens, Bilel Aloui, Alexis Verhaeghe

22 juin 2016

Étudiants : Anthony Boens, Bilel Aloui, Alexis Verhaeghe

Encadrant : Julien Dehos

Date de début : 06 juin 2016

Date de fin : 22 juin 2016

Objet du projet : Jeu de carte Uno à 4 joueurs

Table des matières

1	Présentation du projet	2
1.1	Analyse de la demande	2
1.2	Spécifications	2
2	Réalisation	2
2.1	Présentation	3
2.2	Architecture générale	4
2.3	UML	4
2.4	Cartes spécifiques	4
2.4.1	Les cartes changement de sens	4
2.4.2	Les cartes passe-tour	4
2.4.3	Les cartes +2	5
2.4.4	Les cartes +4	5
2.4.5	Les cartes changement de couleur	5
3	Bilan	6
3.1	Déroulement du projet	6
3.2	Réalisation des objectifs	6
3.3	Conclusion pour les projets futurs	6

1 Présentation du projet

Notre projet consiste à créer un jeu de carte UNO à 4 joueurs en réseau avec une interface graphique. Au lancement du jeu, un écran de login demande le pseudo et l'adresse IP du serveur. Une fois que les 4 joueurs sont connectés, notre serveur distribue les cartes et la cartes du milieu. Un petit jeton indique la personne qui doit jouer et un petit compteur affiche le nombre de cartes des adversaires. Une fois que qu'un joueur a posé sa dernière carte, le jeu s'arrête et affiche une image « perdant » ou « gagnant ».

1.1 Analyse de la demande

La demande du client nous a imposé de faire le projet en C++ et la partie réseau et graphique en SFML. Le délais de réalisation du projet était de 3 semaines. Les principaux besoins étaient de permettre à un joueur de se connecter sur un serveur et faire une partie en multijoueur de UNO avec toutes les règles du jeu respectées. La première proirité était d'avoir un jeu fonctionnel comprenant toutes les règles (y compris les cartes changement de couleur), la seconde était de pouvoir jouer en réseau.

1.2 Spécifications

Spécification serveur :

1. Lancement du serveur
2. Stockage des clients
3. Distribution des cartes
4. Quand les 4 joueurs sont connectés, envoie de la carte courante (carte au milieu)
5. Lancement du jeu (envoi de la demande de joueur au joueur)
6. Mise à jour de la position du jeton qui indique quel joueur doit jouer
7. Gestion des cartes spéciales
8. Envoie de carte supplémentaire quand le joueur ne peut pas jouer
9. Si le joueur ne peut toujours pas jouer, il passe son tour

Spécification client :

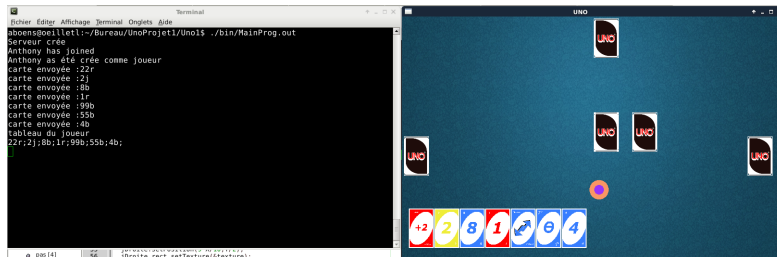
1. Lancement de la fenêtre de login
2. Connexion au serveur si le formulaire est bien rempli (en appuyant sur la touche Entrée)
3. Lancement de l'interface graphique du jeu
4. Possibilité de cliquer uniquement sur les cartes « jouables »
5. Envoie de la carte choisit
6. Affichage de l'image et fin du jeu quand un joueur n'a plus de cartes

2 Réalisation

Le projet a entièrement été réalisé en C++ et plus précisément avec la bibliothèque SFML.

2.1 Présentation

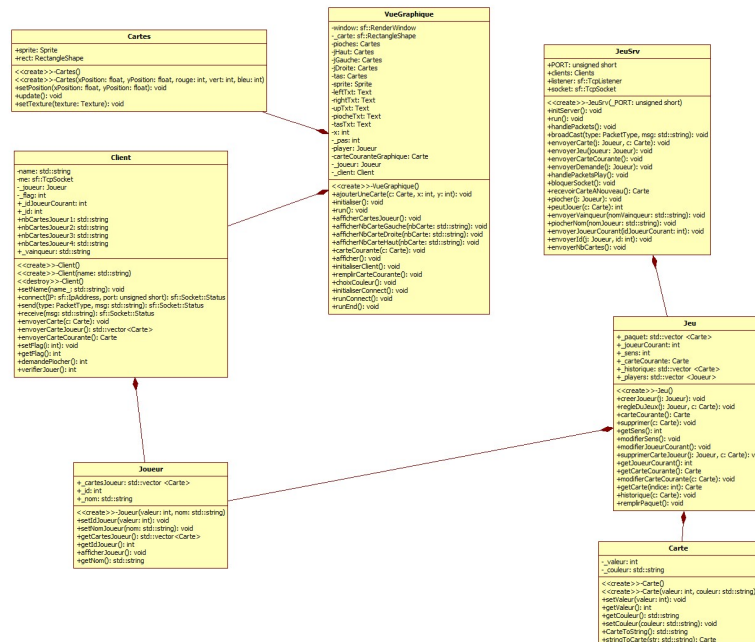
Le logiciel implémenté permet de charger un terminal qui lance un serveur (specification 1 du serveur) et affiche si les clients sont bien connectés (specification 2 du serveur). Dans la seconde fenêtre, on voit l'interface graphique du client avec les cartes distribuées mais pas la carte courante (specification 3 du client).



Avant de lancer l'interface graphique du jeu, un écran de login apparaît et demande le pseudo et l'adresse IP du joueur (specification 1 du client). Le carré de connexion devient vert quand le formulaire est rempli et quand le client appuie sur la touche Entrée, le client envoie son pseudo et se connecte au serveur (specification 2 du serveur).

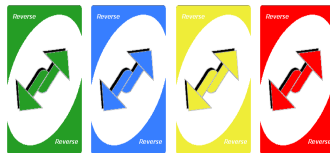


2.3 UML



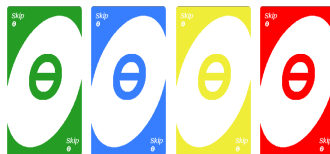
2.4 Cartes spécifiques

2.4.1 Les cartes changement de sens



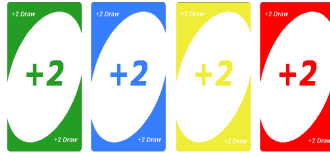
Ces cartes permettent de changement le sens du jeu. Si le jeu tourne dans le sens des aiguilles d'une montre, en jouant cette carte alors le jeu tournera dans le sens inverse.

2.4.2 Les cartes passe-tour



Si ces cartes sont jouées, le joueur suivant ne peut pas jouer, il passe son tour.

2.4.3 Les cartes +2

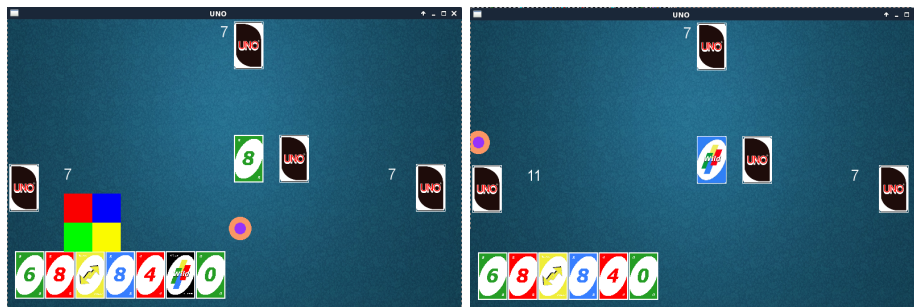


Si ces cartes sont jouées, le joueur suivant prend 2 cartes de plus dans son jeu. Il peut quand même jouer.

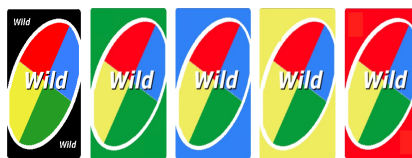
2.4.4 Les cartes +4



Cette carte peut être jouée peu importe le chiffre et la couleur de la carte courante. Dans le jeu du joueur, la carte est noire à la base. Quand la carte est jouée alors un curseur apparaît et demande au joueur de choisir la couleur. En fonction de son choix, la carte courante change de couleur, le joueur suivant prend 4 cartes et il peut jouer si il a la même couleur.



2.4.5 Les cartes changement de couleur



Cette carte peut être jouée peu importe le chiffre et la couleur de la carte courante. Dans le jeu du joueur, la carte est noire à la base. Quand la carte est jouée alors un curseur apparaît et demande au joueur de choisir la couleur (même fonctionne que pour le carte +4).. En fonction de son choix, la carte courante change de couleur, le joueur suivant peut jouer si il a la même couleur.

3 Bilan

3.1 Dérroulement du projet

Nous avons eu plusieurs différences par rapport à ce que nous avions prévu au départ, notamment sur le nombre de joueurs possibles. Au début nous envisagions de faire une partie entre 2 et 4 joueurs mais nous avons préféré de faire une partie uniquement à 4 joueurs pour des raisons techniques et par manque de temps. Nous avons aussi rencontré des problèmes pour assembler les codes de l'interface graphique, du réseau et des règles du jeu. En effet, chacun de nous travaillait sur une partie, indépendamment des autres pendant une semaine. Nous nous sommes rendu compte que ce n'était pas le bon choix donc la deuxième semaines nous avons travaillé ensemble pour comprendre le code de chacun et les associer. A ce moment, on a pu créer un jeu fonctionnel et se concentrer sur les problèmes techniques du jeu. Au début du projet, nous avons voulu passer le pseudo de chaque client et l'adresse du serveur en argument lors de l'exécution. Nous avons préféré créer un menu de login pour que notre code puisse être utilisé par un utilisateur lambda.

3.2 Réalisation des objectifs

fonctionnalité	réalisation
Lancement du serveur et stockage des clients	complète
Distribution des cartes aux clients	complète
Envoie de la carte courante	complète
Lancement du jeu	complète
Interface graphique	complète
Jeton indiquant le joueur qui doit jouer	complète
Gestion des cartes spéciales	complète
Fenêtre de login	complète
Connexion au serveur avec une IP et enregistrement du pseudo	complète
Envoie des cartes vers le serveur	complète
Affichage de la fin du jeu	complète
Blocage de 4 connexions au serveur	Partielle
Possibilité de relancer une nouvelle partie	non
Possibilité de choisir le nombre de joueurs	non

3.3 Conclusion pour les projets futurs

Pour les projets futurs, nous analyserons mieux la demande et feront un MVC mélangeant l'interface graphique, le réseau et le jeu avant de se lancer sur le code. Sinon c'est un projet qui nous a permis d'apprendre de nouvelles choses et surtout apprendre à travailler en groupe sur un projet commun.