RAPPORT DE LA MISE EN PRATIQUE APPLIQUÉE
L3 II

2019 - 2020

Sujet : Un jeu simple en réseau et en C

Tuteur : Laurent Lemarchand

Maxime Yonnet - Anaël Quéant

Sommaire

I. Présentation du sujet

A. Introduction

Le projet se déroule depuis début novembre jusqu'à début Janvier, les groupes étaient composés de 1 à 4 personnes (en fonction des sujets), dans ce projet là nous étions deux, Maxime Yonnet et Anaël Quéant. Chaque semaine on envoyait un mail au tuteur pour lui faire part de l'avancement du projet.

B. Choix du Sujet MPA

Nous avons choisi ce sujet de Mise en Pratique Appliqués afin de consolider nos connaissances et compétences dans le langage C. De plus, faire un jeu est toujours quelque chose de plaisant et de motivant pour travailler. Cela semblait donc être le projet parfait pour nous.

C. Contraintes du jeu

Le but de ce projet était donc de faire un jeu dans le langage C pouvant être joué en réseau.

Le projet possédait 2 contraintes:

L'affichage d'une interface graphique et du mise en place des pions via l'interface et non des coordonnées à taper au clavier.

La mise en place de partie en réseau via l'IP afin d'y jouer depuis 2 ordinateurs différents.

D. Choix du jeu

Plusieurs jeux nous avaient été proposés, mais il nous fallait un jeu simple dans lequel les règles de celui-ci n'allait pas être un obstacle pour la programmation du jeu. Nous avions donc choisis de réaliser un Puissance 4 et si nous avions le temps, faire ensuite un jeu d'échecs. Le Jeu est donc un puissance 4, chaque joueur place un jeton l'un après l'autre, lorsqu'un joueur a 4 de ces jetons cotes à cotes (que ce soit horizontalement, verticalement ou diagonalement, il gagne la partie.

Finalement, l'idée des échecs en C n'a pas été retenue au vu du temps restant après la mise en place du Puissance 4.

II. Présentation du travail

A. Organisation de l’espace du travail

Pour ce projet, nous avons créé un Gitlab afin de pouvoir partager rapidement le code et rapidement le montrer à notre tuteur. Nous discutions sur Messenger en écrit et Discord pour les communications vocales. Nous avons tous deux utilisé le sub-system Ubuntu disponible sur le windows store afin de programmer sur un environnement linux. Maxime utilisait VI pendant que Anaël utilisait nano pour éditer les fichiers, enfin nous compilons avec l’utilitaire GCC. Nous travaillions durant nos heures libres chez nous.

B. Partage du travail

Pour ce qui est de la répartition du travail, pendant que Anaël travaillait sur comprendre le fonctionnement des sockets et le tester, Maxime s’occupait de faire un puissance 4 fonctionnel et de créer une interface graphique. Ensuite Anaël a implémenté ce qu’il a fait en réseau et Maxime a fait en sorte d'intégrer le réseau au jeu déjà existant.

C. Déroulement d’une partie

Une fois le programme exécuté, l’utilisateur est amené sur une page lui proposant de choisir s' il veut jouer en local ou en réseau. Si il choisit local, la partie est lancée et chacun leur tour les deux joueurs utilisent le clavier pour choisir l’emplacement ou le jeton sera mis (annexe 4) à la fin de la partie, le jeu est quitté. Si l’utilisateur choisit le réseau il aura alors le choix entre serveur ou connexion. La partie serveur demande à l’utilisateur de choisir le port et une fois celui-ci entré, l’utilisateur attend d’avoir une connexion et ensuite la partie est lancée. La partie connexion demande à l’utilisateur d'entrer l’ip et le port, une fois cela fait, la partie est lancée. Une fois la partie finie, un écran affichant le numéro du joueur gagnant et le programme est ensuite quitté (annexe 1).

D. Fonctionnement de l’application

L’application peut se jouer à deux sur un même ordinateur, ou bien à deux sur deux ordinateurs distincts. Dans le cas ou le jeu se joue en réseau, un ordinateur sert de serveur et l’autre se connecte dessus, chaque ordinateur calcul eux même le déroulement de la partie, seulement la coordonnée X du jeton est envoyé, et le joueur qui reçoit l’information place le jeton et vérifie si un joueur à gagné ou non (annexe 2).

L’application se sert des sockets afin de réaliser la connexion entre les deux joueurs. Un socket est créé par le serveur qui accepte les connexion, lorsqu’une connexion est acceptée elle est placée sur un autre jeton qui permet d'envoyer et recevoir des informations avec l’autre joueur. Lorsqu’un joueur se connecte, il créer un socket afin d’envoyer des informations et en recevoir (annexe 3), à la fin de la partie tous les sockets sont fermés.

Pour ce qui est de l’aspect graphique, nous avons utilisé la librairie ncurses car elle est simple d’utilisation et permet de faire à peu près ce que l’on veut. Nous avons donc essayé de faire une interface qui s’adapte à la taille des écrans. (annexe 4) Pour l’affichage de la grille, le programme parcourt chaque case du tableau et affiche la case selon la couleur du jeton. Ensuite en dessous du tableau, il affiche le mot “ICI” à tous les emplacements où le joueur peut jouer tout en affichant la case sélectionnée par le joueur et en respectant si c’est au joueur de jouer ou non.

E. Amélioration possible

Si nous avions plus de temps pour faire ce projet, nous pourrions améliorer l'aspect graphique, afin de faire qu'après une victoire les joueurs soient ramenés sur l'écran d'accueil, ou bien faire une vraie interface graphique à l'aide de Glade et GTK. Une autre amélioration qui serait nécessaire serait la gestion d'erreur, car en effet actuellement si un joueur quitte la partie, l'autre joueur continue de jouer tout seul et se retrouve donc bloqué. Aussi, si un joueur n'arrive pas à se connecter cela fait bugger le terminal. Nous pourrions aussi implémenter la possibilité de changer les règles notamment en changeant le nombre de joueurs ou le nombre de jetons qui doivent être alignés. Enfin nous pourrions lors de l'affichage, n'afficher "ICI" que pour les cases où le joueur peut réellement jouer soit toute les colonnes non remplies.

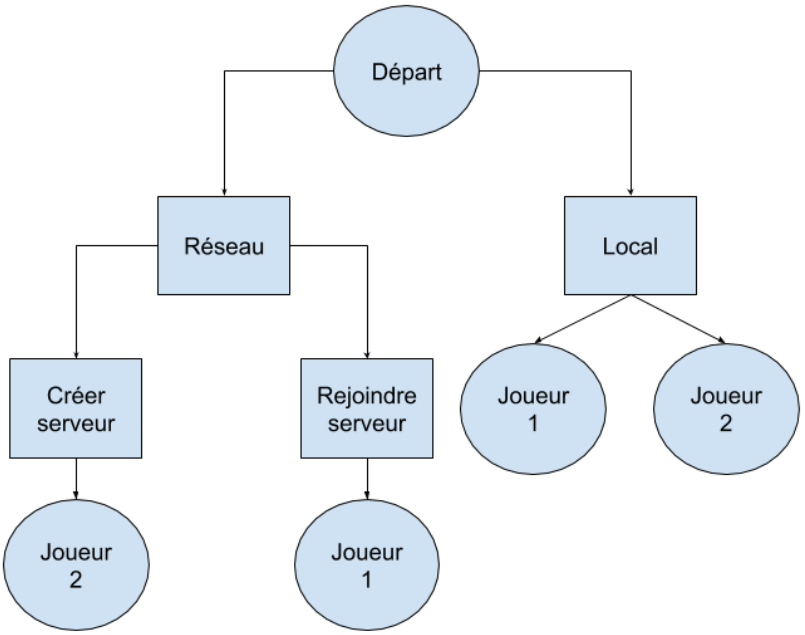
F. Bilan

Le projet maintenant “terminé”, nous pouvons faire un bilan du projet.

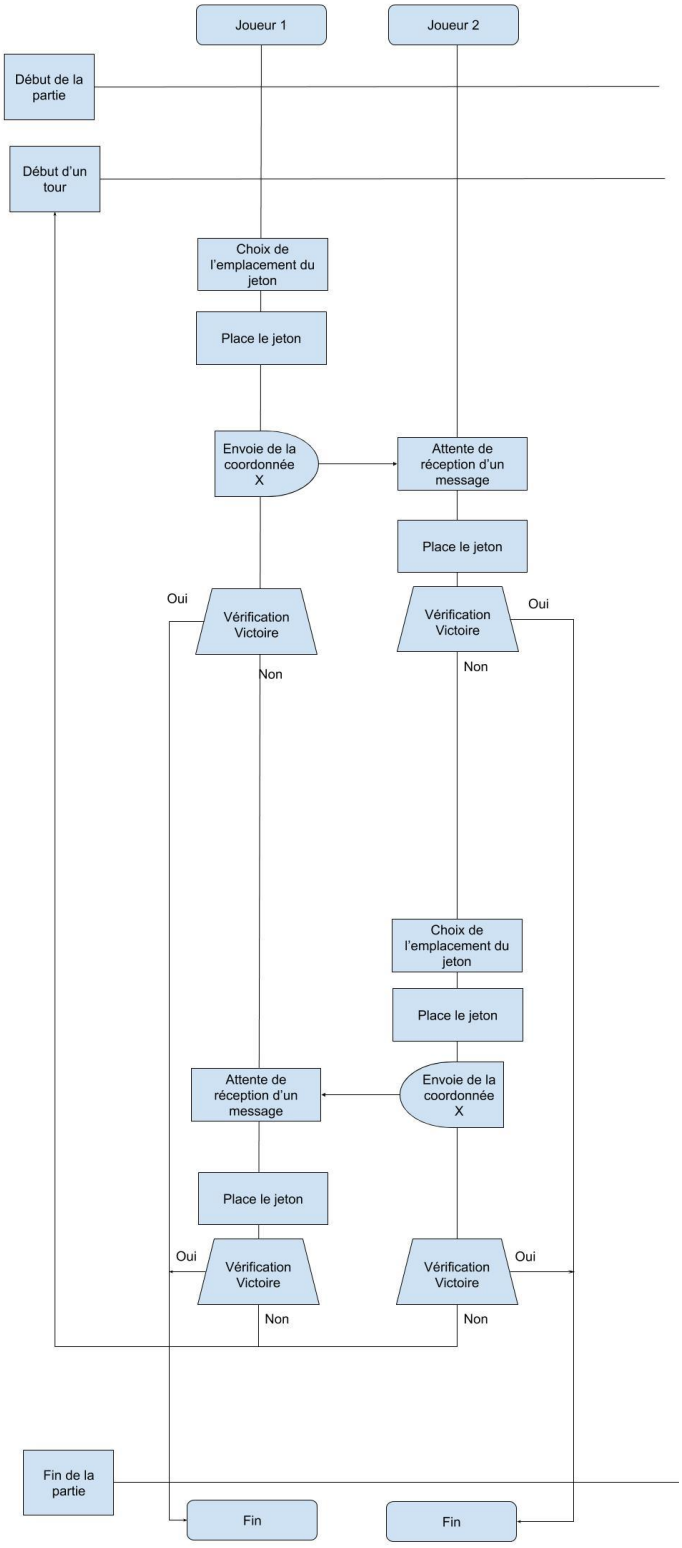
D’abord sur le plan de travail, le projet a été positif pour nous deux, même si Maxime à vraiment piloter et pousser le projet à son terme, alors que Anaël est resté en retrait. La mise en place d’un tel projet en C à permis la progression des 2 développeurs en termes de connaissance et méthode.

Sur le plan du groupe, ce projet a démontré que, malgré la bonne volonté, le groupe n’est pas 100% optimisé. En effet, certe les approches sont différentes ce qui permet d’avoir plusieurs point de vue, mais les méthodes sont trop différentes pour rendre le groupe “viable” sur des projets plus importants. D’un côté, Maxime a pris les devant sans attendre afin d’avancer le projet le plus vite possible, de l’autre, Anaël à plutôt procrastiner et c’est laissé dépasser avant de se remotiver pour les sockets.

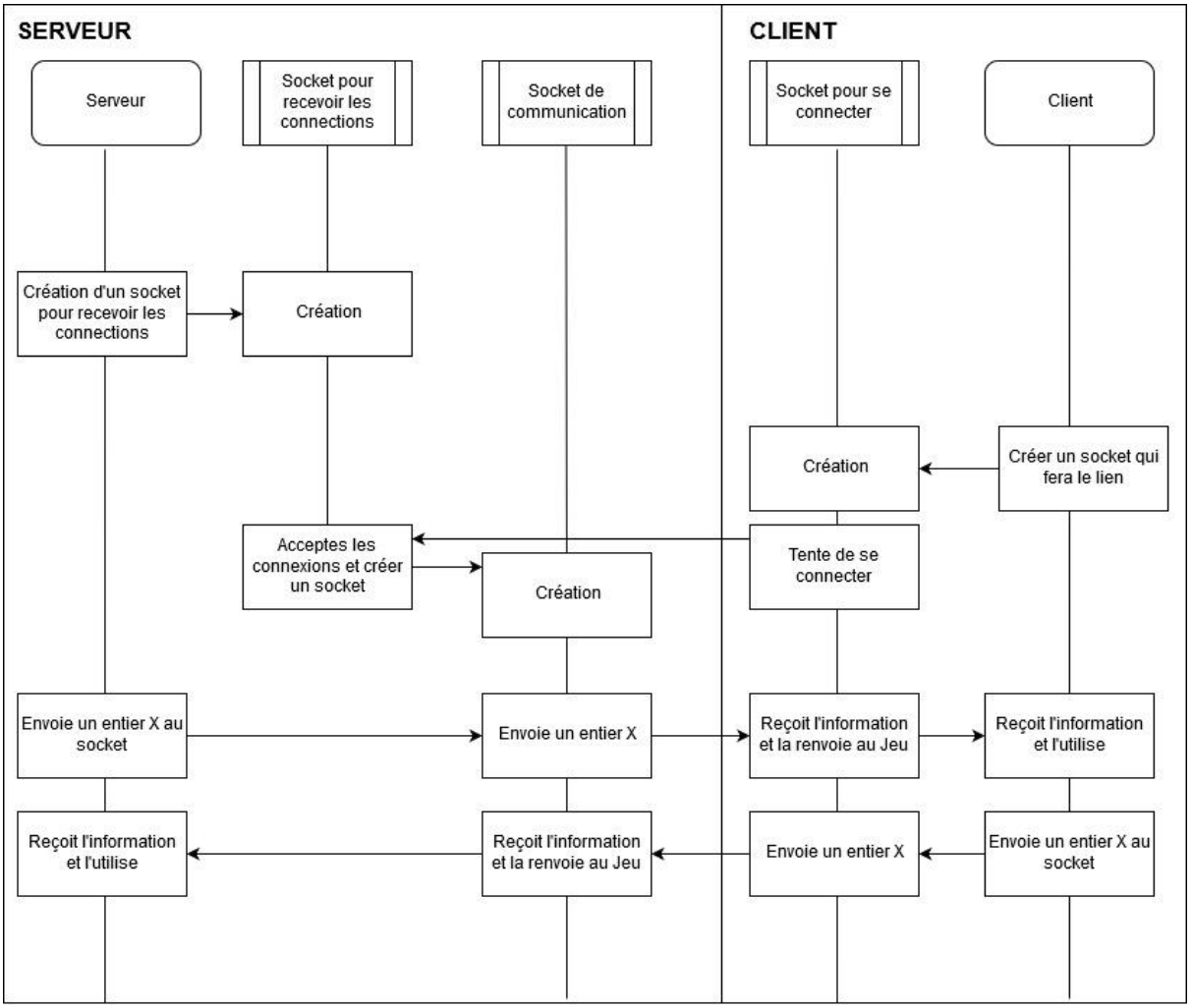
III. Annexe



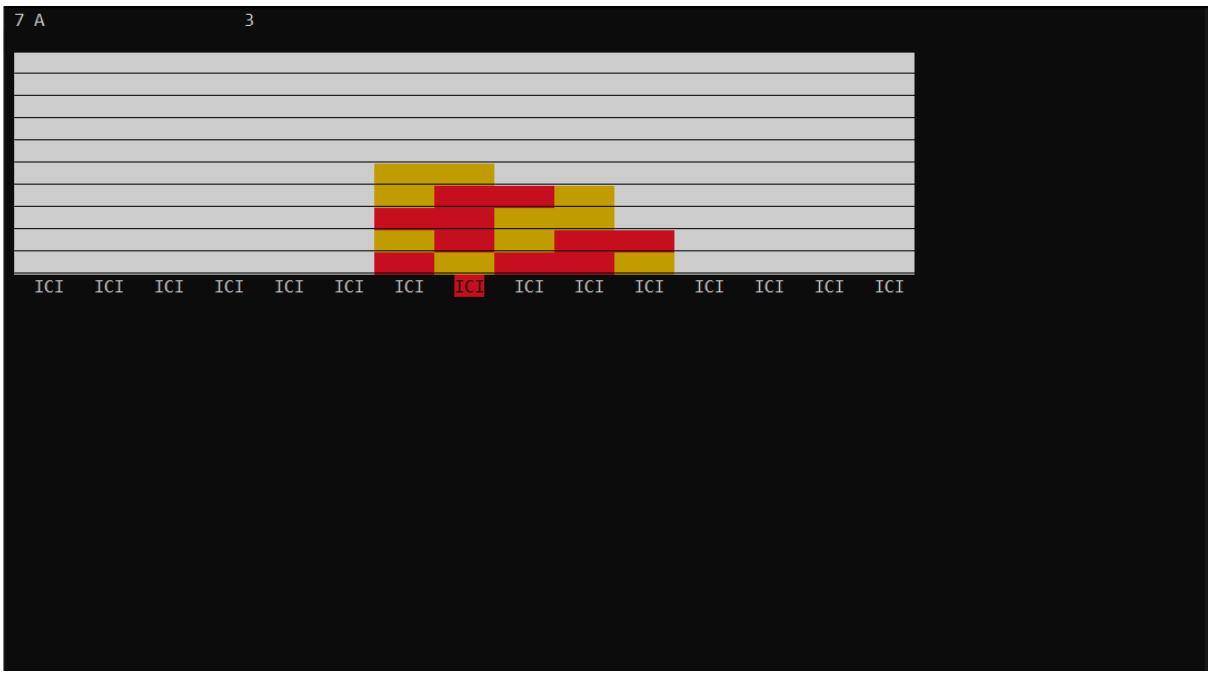
Annexe 1 - Diagramme montrant tous les écrans menant au début de la partie.



Annexe 2 - Diagramme représentant le déroulement d'une partie en réseau (Image)



Annexe 3 - Explication sur les sockets



Annexe 4 - Illustration d’une partie

```
void afficherGraphique(int player, int selected){
	mvprintw(0,0," %d ",selected);

	//Définit la couleur de la case selon si il y a un joueur, lequel, ou non
	init_pair(3,COLOR_BLACK,COLOR_WHITE);
	init_pair(1,COLOR_BLACK,COLOR_RED);
	init_pair(2,COLOR_BLACK,COLOR_YELLOW);

	//Remplit dans un premier temps tout le tableau de pixel blanc
	//Pour chaque positionY
	for(int positionY = 0 ; positionY < unPlateau.tailleY ; positionY++){
		//Pour chaque positionX
		for(int positionX = 0 ; positionX < unPlateau.tailleX ; positionX++){
			//Pour chaque pixel X que doit prendre un jeton
			for(int epaisseurX = 0; epaisseurX <EPAISSEURX; epaisseurX++){
				//Pour chaque pixel Y que doit prendre un jeton
				for(int epaisseurY = 0; epaisseurY <EPAISSEURY; epaisseurY++){
					//Si le jeton est blanc
					if(unPlateau.zoneDeJeu[positionX][positionY].couleur == 3){

						//Active la couleur blanche
						attron(COLOR_PAIR(unPlateau.zoneDeJeu[positionX][positionY].couleur));
						//Si le pixel est la premiere ligne du jeton
						if((epaisseurX*epaisseurY)%(EPAISSEURX*EPAISSEURY) == 0)
							//Met des tirets dans un but esthetique
							mvprintw(unPlateau.tailleY*EPAISSEURY - positionY*EPAISSEURY+1-epaisseurY ,positionX*EPAISSEURX+1+epaisseurX, "_");
						//Sinon colories juste la case
						else mvprintw(unPlateau.tailleY*EPAISSEURY - positionY*EPAISSEURY+1-epaisseurY ,positionX*EPAISSEURX+1+epaisseurX, " ");
						//Désactive la couleur Blanche
						attroff(COLOR_PAIR(unPlateau.zoneDeJeu[positionX][positionY].couleur));
					}
				}
			}
			//Si le numéro de joueur récu en parametre est égal au numéro de joueur du terminal
			if(numeroJoueur == player){
				//Si le numero de jeton est celui selectionné, désactive la couleur pour l'indiquer
				if(positionX == selected) attron(COLOR_PAIR(player));
				//Indiques sous chaque jeton si celui ci est selectionné ou non
				mvprintw(unPlateau.tailleY*EPAISSEURY +1,EPAISSEURY , positionX*EPAISSEURX+1+EPAISSEURX/3, "ICI");
				//Si le numero de jeton est celui selectionné, désactives la couleur pour l'indiquer
				if(positionX == selected) attroff(COLOR_PAIR(player));
			}
		}
	}
	//Si le numéro de joueur récu en parametre n'est pas égal au numéro de joueur du terminal
	if(numeroJoueur != player){
		//Affiche un message d'attente
		mvprintw(unPlateau.tailleY*EPAISSEURY+EPAISSEURY,0,"En attente de l'autre Joueur");
	}
```

```
	//Ajoutes les jetons
	////Pour chaque positionY
	for(int positionY = 0 ; positionY < unPlateau.tailleY ; positionY++){
		//Pour Chaque positionX
		for(int positionX = 0 ; positionX < unPlateau.tailleX ; positionX++){
			//Pour chaque pixel Y que doit prendre le jeton
			for(int epaisseurX = 0; epaisseurX <EPAISSEURX; epaisseurX++){
				//Pour chaque pixel X que doit prendre le jeton
				for(int epaisseurY = 0; epaisseurY <EPAISSEURY; epaisseurY++){
					//Si le jeton n'est pas blanc
					if(unPlateau.zoneDeJeu[positionX][positionY].couleur != 3){
						//Actives la couleur correspondant au jeton
						attron(COLOR_PAIR(unPlateau.zoneDeJeu[positionX][positionY].couleur));
						//Si le pixel est le premier du jeton
						if((epaisseurX*epaisseurY)%(EPAISSEURX*EPAISSEURY) == 0)
							//Met des tirets dans un but esthetique
							mvprintw(unPlateau.tailleY*EPAISSEURY - positionY*EPAISSEURY+1-epaisseurY ,positionX*EPAISSEURX+1+epaisseurX, "_");
						//Sinon colories juste des cases
						else	mvprintw(unPlateau.tailleY*EPAISSEURY - positionY*EPAISSEURY+1-epaisseurY ,positionX*EPAISSEURX+1+epaisseurX, " ");
						//Désactives la couleur
						attroff(COLOR_PAIR(unPlateau.zoneDeJeu[positionX][positionY].couleur));

					}
				}

			}
		}

	}

	move(0,0);

	//clrtoeol();
	curs_set(0);
	// Rafraichit la fenêtre courante afin de voir le message apparaître
	refresh();
}
```

Annexe 5 - Code de la fonction afficherGraphique() qui affiche le plateau

Documentation utilisé :