

Système et réseau

Projet de fin de semestre

Rémi Nollet

avril 2021

(dernière correction le 16 avril 2021)

Dans ce projet, vous allez implémenter un protocole au dessus de TCP permettant à deux personnes de jouer au Puissance 4¹ en se connectant à travers le réseau.

1 Conventions sur le jeu

Il y a deux joueurs, Rouge et Jaune, qui jouent à tour de rôle. Chacun place dans la grille des jetons de sa couleur. Rouge commence.

Le joueur rouge (le premier joueur) est codé par l'entier 0. Le joueur jaune (le second joueur) est codé par l'entier 1.

	Numéro du joueur	Couleur habituelle
Premier joueur	0	rouge
Second joueur	1	jaune

La grille de jeu a 7 colonnes et 6 lignes. Les colonnes de la grille sont numérotées à partir de 0, en partant de la gauche. Les lignes de la grille sont numérotées à partir de 0, en partant du bas. Quand on aura besoin de numéroté les cases de la grille, on les numéroté comme indiqué sur la figure 1.

De cette manière, le numéro de la case en colonne `col` et en ligne `lin` vaut `col * 6 + lin`. Réciproquement, la case numéro `cas` est en ligne `cas % 6` et en colonne `cas / 6`.

¹https://fr.wikipedia.org/wiki/Puissance_4

5	5	11	17	23	29	35	41
4	4	10	16	22	28	34	40
3	3	9	15	21	27	33	39
2	2	8	14	20	26	32	38
1	1	7	13	19	25	31	37
0	0	6	12	18	24	30	36
	0	1	2	3	4	5	6

FIG. 1 : Numérotation des lignes, colonnes et cases de la grille

2 Format des TLV

Vos programmes échangeront des messages à travers une connexion TCP. Chaque message est un TLV : un triplet (type, longueur, valeur). Chaque TLV a la forme suivante :

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Type   | Length | Body...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Les champs sont définis comme suit :

Type : le type du TLV, un entier non signé.

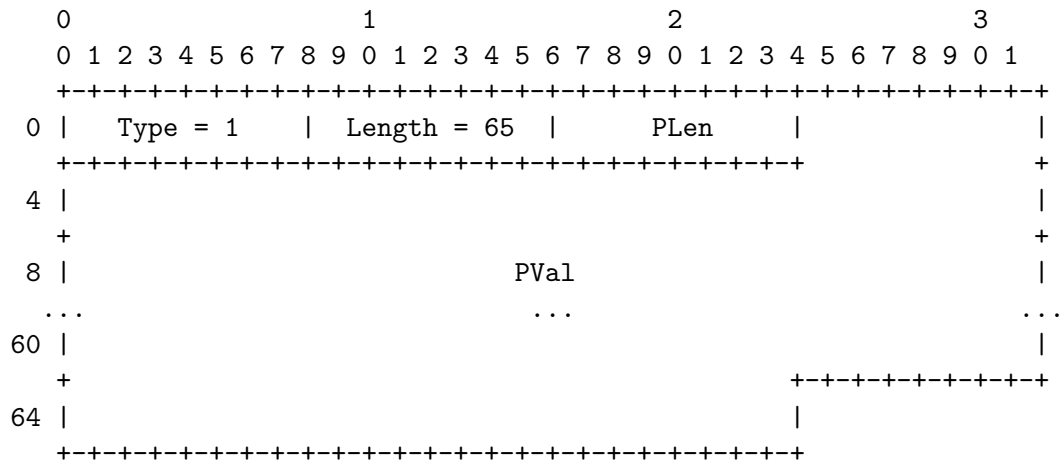
Length : la longueur du TLV, sans compter les champs **Type** et **Length** ; un entier non signé.

Body : une suite d'octets de longueur **Length**.

Tous les entiers qui apparaissent dans les TLV sont représentés de manière non signée.

3 Détails des TLV

PSEUDO



(Notez que, tandis que les nombres au dessus du TLV comptent en bits, comme pour les autres TLV, les nombres à gauches comptent en octets.)

Ce TLV est envoyé par un client au début de la partie pour annoncer le pseudo du joueur. Le champ **PLen** est un entier qui indique la longueur, en octets, du pseudo. Cette longueur doit être ≤ 64 . Les **PLen** premiers octets du champ **PVal** contiennent le pseudo. Les 64—**PLen** derniers octets du champ **PVal** doivent être nuls à l'écriture, et sont ignorés à la lecture.

	0										1										2										3									
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
0	Type = 2										Length = 131										PCol										PPLen									
4																																								
8																					PPVal																			
...																													
60																																								
64																																								
68	OPLen																																							
72																																								
76																					OPVal																			
...																													
124																																								
128																																								
132																																								

4

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Type = 3       | Length = 44 |                               STATE                                |               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                                           |
+                                                           +
|                                                           |
+                                                           +
|                                                           |
+                                                           +
|                                                           |
+                                                           +
|                                     CONT                    |
+                                                           +
|                                                           |
+                                                           +
|                                                           |
+                                                           +
|                                                           |
+                                                           +
|                                                           |
+                                                           +
|                                                           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                                           |
```

- Si son premier octet vaut 0, son second octet indique quel joueur doit jouer le prochain coup : 0 pour Rouge, 1 pour Jaune.
- Si son premier octet vaut 1, la partie a été gagnée par un joueur. Son second octet indique alors quel joueur a gagné.
- Si son premier octet vaut 2, la grille est complètement remplie sans qu'aucun joueur ait gagné et la partie est nulle. Dans ce cas le deuxième octet vaut 0 à l'écriture et est ignoré à la lecture.

5

MOVE

```
0                               1                               2
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-----+-----+-----+-----+-----+-----+-----+
|  Type = 4    |  Length = 1    |      COL      |
+-----+-----+-----+-----+-----+-----+-----+
```

Ce TLV est envoyé par le client pour indiquer quel coup il veut jouer, ou par le serveur pour indiquer quel coup a joué l'adversaire. Le champs COL indique la colonne dans laquelle le coup est joué.

MOVEACK

```
0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|  Type = 5    |  Length = 2    |      COL      |      OK      |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Ce TLV est envoyé par le serveur en réponse à un MOVE. Le champ COL redonne la colonne dans laquelle le coup a été joué. Le champs OK vaut 1 si ce coup est accepté et 0 si ce coup est refusé, parce que la colonne a un numéro > 6 ou parce que la colonne est déjà pleine.

CONCEDE

```
0                               1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-----+-----+-----+-----+-----+
|  Type = 6    |  Length = 0    |
+-----+-----+-----+-----+-----+
```

Ce TLV est envoyé par le client pour indiquer qu'il abandonne la partie, ou par le serveur pour indiquer que l'adversaire a abandonné.

DISCON

```
0                               1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-----+-----+-----+-----+-----+
|  Type = 7    |  Length = 0    |
+-----+-----+-----+-----+-----+
```

Ce TLV est envoyé par le serveur pour indiquer que la connection avec l'adversaire a été perdue.

4 Protocole

Une interaction entre un client et le serveur se déroule de la manière suivante, qui est résumée par l'automate de la figure 2 :

0. À l'établissement de la connexion du client vers le serveur, le client envoie un TLV **PSEUDO** indiquant le pseudo du joueur, qui ne doit pas excéder 64 octets, et reçoit en retour un TLV **START** qui lui redonne son pseudo, indique le pseudo de l'adversaire et indique la couleur du joueur.
1. Après cela, le serveur envoie au client la grille initiale, qui est vide, et dans laquelle c'est à Rouge de commencer. Si c'est au joueur de jouer, c'est-à-dire s'il a reçu la couleur Rouge dans le TLV **START** de l'étape 1, il va à l'étape 2, sinon il va à l'étape 5.
2. Le client arrive à cette étape quand c'est le tour du joueur. Il doit alors envoyer un TLV **MOVE** au serveur, lui indiquant un coup valide à jouer. Il peut aussi envoyer un TLV **CONCEDE** si le joueur veut abandonner la partie. Dans ce dernier cas, il ferme alors la connexion.
3. Si le coup qu'il a envoyé est invalide, le client reçoit du serveur un **MOVEACK** qui le lui indique, suivi d'un **GRID** qui lui redonne l'état de la grille sur laquelle le joueur doit jouer. On revient alors à l'étape 2. Notez qu'en principe cela ne doit pas arriver puisque le client, à l'étape 2, doit envoyer un coup valide au serveur. Dans ce cas, il doit recevoir un **MOVEACK** qui lui confirme la validité de son coup.
4. Ce **MOVEACK** est alors suivi d'un **GRID** qui indique le nouvel état du jeu. Si ce **GRID** indique que le joueur a gagné la partie avec ce coup, ou qu'il a rempli la grille sans qu'il y ait de vainqueur, la connexion est fermée. Si au contraire ce **GRID** indique que c'est à l'adversaire de jouer, alors le client attend le coup de l'adversaire.
5. S'il reçoit un **CONCEDE** indiquant que l'adversaire abandonne la partie, il ferme la connexion. Sinon il reçoit un **MOVE** indiquant un coup valide de l'adversaire.
6. Ce **MOVE** est alors suivi d'un **GRID** qui indique le nouvel état du jeu après le coup de l'adversaire. Si l'adversaire a remporté la partie avec ce coup, ou s'il a rempli la grille sans qu'il y ait de vainqueur, la connexion est fermée. Sinon c'est à nouveau au joueur de jouer, et on retourne à l'étape 2.
7. Quand la partie est finie, la connexion est fermée.

En plus de tout cela, à chaque fois que le client attend un message du serveur, il peut recevoir un TLV **DISCON**, qui indique que le client de l'adversaire a été déconnecté du serveur. Dans ce cas, la connexion est fermée (étape 7).

5 À faire

Implémentez un client et un serveur TCP qui réalisent ce protocole.

Le client doit interagir avec le serveur et avec l'utilisateur. Lorsque le client doit envoyer un **MOVE**, il doit demander à l'utilisateur quel coup jouer. Quand il reçoit un coup de l'adversaire, il doit afficher la grille qui en résulte.

Le serveur doit attendre deux connexions de clients, puis créer un nouveau processus qui les fait jouer et attendre à nouveau, immédiatement, deux nouvelles connexions. Chaque processus fils assigne aléatoirement les couleurs Rouge et Jaune à ses deux clients, puis les fait jouer selon le protocole décrit plus haut. Quand la partie est finie, le fils ferme les connexions et termine. Si un de ses client se déconnecte, un fils envoie un **DISCON** à l'autre client puis ferme ses connexions et termine. Le père ne devra pas oublier de récupérer régulièrement ses fils zombies.

6 Rendu

Le rendu doit s'effectuer sur Moodle, avant le lundi 3 mai, c'est-à-dire au plus tard le dimanche 2 mai au soir. Il doit contenir :

- Le code de vos programmes, que je peux compiler et tester.
- Un **README** expliquant comment compiler et exécuter vos programmes.
- Un court rapport, au format texte ou pdf, indiquant les noms des membres du binôme et expliquant ce que vous avez réussi à faire, comment votre programme est structuré, comment vous vous êtes réparti le travail, ce que vous avez trouvé intéressant, ou difficile, et toute information que vous jugerez utile.

Il n'est bien sûr pas interdit de s'entraider entre binômes ou de chercher de l'aide sur internet si vous êtes bloqués. Je rappelle cependant que vous n'êtes pas censés échanger du code entre binômes ou copier-coller d'importantes parties de votre code depuis l'internet, hormis ce qui est sur le Moodle du cours. Si cela arrivait cependant, il est impératif que vous mentionniez dans votre rapport quelles sont les parties de votre code qui ne sont pas de vous. Intégrer dans vos programmes d'importants morceaux de code qui n'est pas de vous sans en mentionner l'auteur constitue du plagiat et devra être rapporté à une commission disciplinaire.

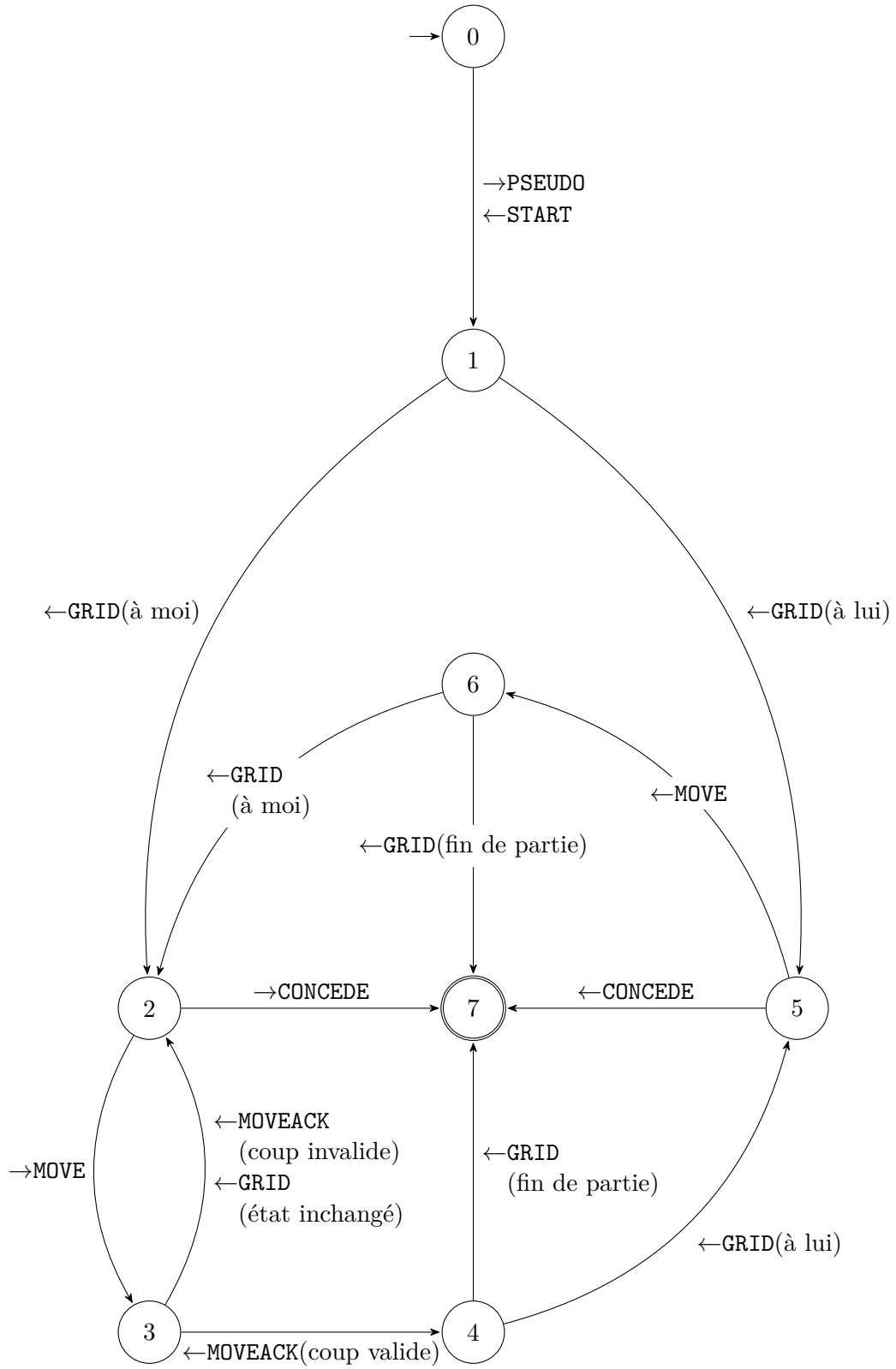


FIG. 2 : Un automate décrivant le protocole