

PROJET S4 à rendre pour le jeudi 31 mars 12H

Programmation réseau

1- Coté Client

Le client

Votre travail consiste à développer un client jouant une partie de Twist-Lock via un serveur de jeu que vous devrez écrire également.

Votre client doit être capable de se connecter au serveur en s'identifiant puis de jouer ses coups lorsque le serveur le demande.

Attention : si votre client n'est plus capable de se connecter au serveur, vous aurez perdu la partie.

Contraintes techniques

Le langage Java n'est pas imposé. La communication vers le serveur se faisant via une connexion socket réseau sur un port dédié, n'importe quel type de client peut être envisagé, utilisant un protocole UDP, à condition qu'il satisfasse aux exigences du serveur en termes d'envoi et de réception des messages.

Phases importantes du jeu

Connexion au serveur

La connexion au serveur se fera en envoyant le nom de l'équipe sur la machine où le serveur est exécuté

Un message réponse du serveur vous signifiera quelle couleur vous êtes dans la partie.

1-Bonjour Equipe the Best

Vous etes le Joueur 1 (ROUGE), attente suite ...

ou bien

2-Bonjour les zouzous

Vous etes le Joueur 2 (VERT), attente suite ...

Début du jeu

Un message est ensuite envoyé aux clients connectés pour indiquer que la partie va commencer, avec la map qui sera le support de cette partie. Attention, d'une partie à l'autre, le nombre de lignes et de colonnes peut changer, et les valeurs des conteneurs également (1 seul envoi).

01-la partie va commencer

MAP=5:7:52:18:10:44:51|52:49:24:46:12:41:25|26:28:22:23:13:38:26|

43:29:33:41:06:52:50|42:45:48:12:16:51:20|37:06:25:26:26:43:33|

5:47:24:5:23:22:18|5:38:38:41:39:17:40|8:19:17:53:29:20:27|

Elle vous permettra si vous le souhaitez d'afficher une version du plateau sur votre écran.

Puis à tour de rôle, Vous indiquerez, après avoir reçu l'invitation à jouer du serveur, les coordonnées de votre Twist-Lock tout attaché (LigneColonneCoin) comme dans l'exemple d'échange ci-dessous. Le serveur enverra votre proposition à l'autre concurrent.

Partie coté VERT

10-A vous de jouer (VERT) :
envoi au serveur chaine 9B1

20-coup adversaire:8B3
10-A vous de jouer (VERT) :
envoi au serveur chaine 8B1

etc

Partie coté ROUGE

20-coup adversaire:9B1
10-A vous de jouer (ROUGE) :
envoi au serveur chaine 8B3

20-coup adversaire:8B1
10-A vous de jouer (ROUGE) :
envoi au serveur chaine 7D1

Si vous envoyez une mauvaise indication, vous recevez alors le message suivant
21-coup joué illégal

Si votre adversaire envoie une position illégale, au lieu de recevoir sa position, vous recevez :
22-coup adversaire illegal

Quand vous n'avez plus de Twist-Lock, le message suivant est envoyé par le serveur
50-Vous ne pouvez plus jouer

On a aussi le message :
91-demande non valide

A la fin de la partie, vous recevrez
88-Partie Terminée, vous avez gagné 391 - 306

Il n'y a pas obligation de développer l'interface graphique pour les clients. Nous pouvons imaginer que chaque client peut visualiser l'écran du serveur, et ainsi pouvoir « jouer » et choisir les positions des twist-lock à la demande

2- Coté Serveur

Le serveur

Votre travail consiste à développer le serveur du jeu pour jouer au TWIST-LOCK à 2 joueurs. Le protocole réseau est imposé : UDP. Les différentes phases sont indiquées ici :

Identification des clients

Le serveur écoute sur un port défini. S'il capte un message alors le message capté est le nom du premier joueur. Il envoie au premier client un seul message :

1-Bonjour Equipe the Best

Vous etes le Joueur 1 (ROUGE), attente suite ...

Le serveur ecoute à nouveau pour attendre le 2^{ème} joueur et envoie au second client un seul message :

2-Bonjour les zouzous

Vous etes le Joueur 2 (VERT), attente suite ...

Initialisation du jeu

une MAP représentant le nombre de lignes et colonnes ainsi que les valeurs des containers est générées puis envoyée aux 2 clients repérés. Le message suivant (par exemple) est envoyé en un seul message :

01-la partie va commencer

MAP=5:7:52:18:10:44:51|52:49:24:46:12:41:25|26:28:22:23:13:38:26|

43:29:33:41:06:52:50|42:45:48:12:16:51:20|37:06:25:26:26:43:33|

5:47:24:5:23:22:18|5:38:38:41:39:17:40|8:19:17:53:29:20:27|

Pour des raisons pratiques, la map générée aléatoirement aura entre 4 et 9 lignes et entre 4 et 9 colonnes. Chaque container aura une valeur aléatoire entre 4 et 54.

le jeu

A tour de rôle, le serveur enverra une invitation à jouer au client voulu et attendra sa réponse (en vérifiant bien que le message reçu provient du joueur attendu) qu'il renverra à l'adversaire (comme dans l'exemple d'échange du sujet sur le client)

Attention, en UDP un autre poste pourrait tenter d'envoyer un message sur le serveur et le port du jeu. Si un tel message arrive, il faudra donc reconnaître que le message provient d'un joueur non valide et renvoyer un message :

91-demande non valide

Dans l'échange présenté plus haut, différents messages en fonction des cas pourront être envoyés.

fin du jeu

Le serveur envoie au joueur un message pour indiquer la fin du jeu et le score.