

Projet PSE - SLAM RELOADED - Manuel

Lucas VINCENT

lucas.vincent@etu.emse.fr

Promotion EI23 - Groupe IV

Mai-Juin 2024

Baptiste RIQUIER

baptiste.riquier@etu.emse.fr



Fig. 1. – Logo du jeu télévisé SLAM depuis 2017

Table des matières

1. Installation de l'archive	3
2. Exécution des programmes	4
3. Déroulement d'une partie	5

1. Installation de l'archive

De manière à préparer le jeu sur votre machine, veuillez suivre les instructions suivantes :

[1] Commencer par extraire l'archive *RIQUIER_VINCENT.tar* dans le répertoire de votre choix.

```
user:~$ tar xvf RIQUIER_VINCENT.tar
```

[2] Ensuite, se placer dans ce répertoire.

```
user:~$ cd ./RIQUIER_VINCENT
```

[3] À partir de là, en tapant la commande **ls**, vous devriez obtenir le résultat suivant :

```
CLIENT  LIB  README.md  SERVEUR  SourceMe.txt
```

[4] Pour compiler le projet, il suffit d'exécuter les deux makefiles à l'aide du fichier source.

```
user:~$ source SourceMe.txt
```

[4bis] Si tout se passe bien vous devriez avoir le résultat suivant :

```
user:~$ source SourceMe.txt
- - - Initialisation du projet - - -
- - Makefiles - -
→ Serveur
gcc -c serveur.c
gcc -c jeu_serveur.c
gcc -c ../LIB/utilities.c
gcc -Wall -Wextra -g serveur.o jeu_serveur.o utilities.o -o Serveur_start
→ Client
gcc -c client.c
gcc -c jeu_client.c
gcc -c ../LIB/utilities.c
gcc -Wall -Wextra -g client.o jeu_client.o utilities.o -o Client_start
- - Nettoyage du projet - -
```

2. Exécution des programmes

SERVEUR :

Pour lancer le serveur, il suffit de taper la commande suivante dans un terminal :

```
user:~$ ./SERVEUR/Serveur_start [PORT]
```

Le serveur est désormais actif et va commencer à poser des questions à son administrateur.

REMARQUE : Vous pouvez remplacer [PORT], par n'importe quel port acceptable. Par exemple, 2500.

CLIENTS :

Ouvrez autant d'instances de terminal, que de clients souhaités. Dans chacun des terminaux, taper la commande suivante :

```
user:~$ ./CLIENT/Client_start localhost [PORT]
```

REMARQUE : Le port, ici [PORT], doit être identique à celui du serveur.

Félicitation ! Vous êtes prêt à jouer !

3. Déroulement d'une partie

Pour observer le fonctionnement du serveur et des clients, nous étudierons l'exemple d'une partie nommée : **PSE**, composée de **2 joueurs**. Dans un premier temps, l'administrateur configure le serveur :

```
- - - SLAM Reloaded - - -  
Choisissez le nombre de joueurs : 2  
Choisissez le nom de la partie : PSE
```

S'ouvre alors dans le terminal, la fenêtre suivante :

```
Every 1.0s: cat ./SERVEUR/salle_attente.txt  
  
< - - -[Salon PSE]- - - >  
Joueur 1 : Pas de joueur sur le canal -1  
Joueur 2 : Pas de joueur sur le canal -1
```

Pour l'instant aucun joueur n'est connecté. Le nom des joueurs est donc celui par défaut : *Pas de joueur* et les variables *canal* sont à leurs valeurs par défaut : *-1*.

Un premier joueur nommé : *Basile* se connecte :

```
Quel est votre nom ?  
Nom : Basile
```

Il est désormais connecté au serveur, en attendant les autres joueurs, il peut lire les règles du jeu :

```
Phase 1: Questions  
1. Une question sera posée, la réponse est une lettre.  
2. Celui qui trouve la réponse le plus vite remporte la phase 1 et passe en phase 2.  
Phase 2: Mots croisés  
3. La lettre devinée est révélée dans la grille de mots croisés.  
4. Si la lettre est présente dans plusieurs mots, le gagnant de la phase précédente choisit le mot à deviner.  
5. Le gagnant de la phase 1 doit deviner le mot à l'aide des lettres présentes et d'un court indice.  
6. S'il devine le mot, il gagne autant de points qu'il y a de lettres dans le mot deviné.  
7. A ce moment là, n'importe quel joueur peut Slammer, pour rentrer dans la phase 3, sinon retour à la phase 1 ou fin de la manche si tous les mots sont devinés.  
Phase 3: SLAM  
8. Lorsqu'un joueur déclenche le SLAM, il doit tenter de deviner tous les mots incomplets à la suite.  
9. Il n'a pas accès à la définition des mots.  
10. S'il se trompe sur l'un des mots, il perd la partie, retour à la phase 1 pour les autres joueurs.  
11. Comme à la phase 2, il gagne autant de points qu'il y a de lettres dans les mots devinés.  
Autre :
```

11. Le gagnant est celui qui a le plus de points une fois la grille complétée.

Dans la salle d'attente du serveur se trouve désormais Basile, qui a automatiquement été déclaré joueur 1.

Du côté du serveur, on observe que Basile est bien connecté :

```
Every 1.0s: cat ./SERVEUR/salle_attente.txt  
< - - -[Salon Programmation]- - - >  
Joueur 1 : Basile sur le canal 6  
Joueur 2 : Pas de joueur sur le canal -1
```

Pendant, que Basile lisait les règles, un deuxième joueur nommé *Thomas* s'est connecté :

```
Every 1.0s: cat ./SERVEUR/salle_attente.txt  
< - - -[Salon PSE]- - - >  
Joueur 1 : Basile sur le canal 6  
Joueur 2 : Thomas sur le canal 7
```

Après 5 secondes d'attente, la partie se lance :

```
serveur: Début de la partie dans 3  
serveur: Début de la partie dans 2  
serveur: Début de la partie dans 1  
serveur: → Début de la partie
```

Nous n'avons pas réussi à atteindre nos objectifs dans le temps imparti pour ce projet. Ainsi, au lieu de la suite du jeu, les clients et le serveur seront confrontés au message suivant :

```
- - - En construction - - -
```

Après cette limite, les fonctions et les interactions serveur/clients ne fonctionnent pas correctement ou entraînent des *Segmentation Fault*.