

Projet – Jeu Simon avec une architecture client-serveur

Rendu et évaluation

Le projet doit être réalisé seul, sans collaboration avec vos camarades. En cas de triche la note sera divisée par le nombre de protagonistes. De même, tout plagiat sur un code existant sur internet sera fortement sanctionné.

Après la dernière séance, vous disposez de deux semaines pour terminer le projet et le soumettre sous forme d'une archive zip sur Celene. Le projet doit être rendu au plus tard le **vendredi 19 novembre à 23h59**. Tout retard sera sanctionné.

Les critères d'évaluation du projet sont les suivants :

- Fonctionnalités implémentées, respect du cahier des charges.
- Qualité du code. Respect des bonnes pratiques de la programmation objet. Les exceptions doivent être gérées pour envoyer un message à l'utilisateur ou à défaut afficher un message dans la console.
- Qualité de la modélisation. Le projet doit être bien structuré pour être facile à maintenir.
- Propreté du code. Le code doit être indenté et commenté. N'hésitez pas à détailler clairement les fonctionnalités de chaque classe et à expliquer vos choix si vous pensez que cela est nécessaire.
- Exécution. Le code rendu ne doit pas générer d'erreurs ni à la compilation ni à l'exécution. Le rendu d'un code non fonctionnel sera fortement sanctionné. Si vous n'arrivez pas à résoudre des erreurs, commentez la partie du code et expliquez l'erreur.

Sujet



Figure 1 Jeu électronique Simon (source : Wikipedia)

L'objectif du projet est de réaliser un jeu Simon en utilisant une architecture client-serveur exploitant une communication TCP/IP. Selon Wikipedia :

« Simon est un jeu de société électronique de forme circulaire comportant quatre grosses touches de couleurs différentes : rouge, vert, bleu et jaune. Le jeu, électronique, éclaire une des quatre couleurs et produit un son toujours associé à cette couleur. Le joueur doit alors appuyer sur la touche de la couleur qui vient de s'allumer dans un délai assez court.

Le jeu répète la même couleur et le même son, puis ajoute au hasard une nouvelle couleur. Le joueur doit reproduire cette nouvelle séquence. Chaque fois que le joueur reproduit correctement la séquence, le jeu ajoute une nouvelle couleur. »

Ce mode de jeu correspond au mode solo du jeu par défaut. Il existe aussi d'autres variantes.

Pour ce projet, on souhaite implémenter ce jeu afin qu'il soit jouable seul, mais aussi à plusieurs joueurs à distance. Pour cette raison, l'application sera développée suivant une architecture client-serveur.

Côté client, l'architecture s'appuiera sur une interface homme machine permettant à l'utilisateur de presser les boutons pour reproduire la séquence de couleurs. Pour simplifier, on pourra utiliser des boutons classiques sans chercher à reproduire le visuel du jeu d'origine. De même, il n'est pas demandé que le jeu fasse du son.

Côté serveur, l'objectif est de vérifier les séquences transmises, de les complexifier et de gérer le tour de jeu en mode multi-joueur. Plus précisément, en mode solo, le serveur vérifiera la séquence de couleurs transmise. Dès qu'une erreur est commise, il prévient qu'une erreur est commise et il recommence. Si le joueur est parvenu à reproduire la séquence entièrement, le serveur renvoie la séquence en ajoutant une nouvelle couleur.

En mode multi, plusieurs modes de jeux sont possibles :

- **Mode parallèle** : tous les joueurs jouent avec le même serveur mais chacun dans une partie différente.
- **Mode tour à tour** : tous les joueurs jouent avec le même serveur et la même séquence, mais chacun leur tour. Une fois que le premier joueur est parvenu à reproduire la séquence, le serveur retourne à tout le monde la séquence avec une couleur supplémentaire. C'est alors au joueur suivant de reproduire la séquence. Un joueur est éliminé lorsqu'il se trompe.

Il faudra réaliser en priorité une application implémentant un mode solo fonctionnel avant de proposer un mode multijoueur.

Spécifications côté client

L'interface graphique côté client sera constituée des composants suivants :

- **Menu** : Un menu permettant au moins d'accéder à la configuration du serveur.
- **Afficheur** : Un label affichant les informations transmises par le serveur et exécutées côté client.
- **Boutons** : 4 boutons correspondant chacun à une des couleurs. Ils seront disposés sous la forme d'une grille 2x2. Lorsqu'on appuiera sur un bouton, il faudra afficher l'information associée dans l'afficheur (par exemple, « Vous : Rouge » pour le bouton Rouge).
- La fermeture de la fenêtre devra proprement fermer l'application.

Des composants supplémentaires pourront être ajoutés côté client pour améliorer l'interface. Par exemple, un composant pour afficher une barre d'état pour indiquer la longueur de la séquence actuelle et le record ; un composant pour recommencer la partie ; etc.

Protocole d'échange en mode solo

Voici la trame principale du protocole à implémenter entre le client et le serveur :

1. Connexion du client
2. Serveur au client : Transmission du début de partie
3. Transmission de la séquence à reproduire
 - a. Serveur au client : Transmission du « A moi »
 - b. Serveur au client : Transmission des couleurs
 - c. Serveur au client : Transmission du « A toi »
4. Transmission de la séquence reproduite
 - a. Client au serveur : Transmission des couleurs (une à une avec « A toi » si c'est OK)
 - b. On recommence au 2 « Début de partie » si une erreur est commise) ou au 3 (« A moi » si la séquence est complète)

Références

[https://fr.wikipedia.org/wiki/Simon_\(jeu\)](https://fr.wikipedia.org/wiki/Simon_(jeu)) Page Wikipedia du jeu Simon

https://www.youtube.com/watch?v=8fRn5nMsbUg&ab_channel=MisterPub Publicité française pour le jeu Simon