



Bienvenue dans ce défi collaboratif ! Vous allez recréer une version numérique et en réseau inspiré du jeu de société *Fiesta de los Muertos*. Ce jeu coopératif met au défi la mémoire et l'association d'idées des participants pour retrouver des personnages.

Votre mission : développer un jeu multi-joueurs en réseau où chaque participant pourra jouer à distance tout en respectant les règles du jeu original.



Objectifs

1. Concevoir une application en C++ avec Qt, utilisant des sockets TCP ou des WebSockets pour la communication entre les joueurs.
2. Implémenter une base de données SQLite pour stocker :
 - Les personnages à découvrir.
 - Les scores des joueurs.
 - L'historique des parties.
3. Utiliser un fichier de configuration pour ne pas toujours saisir les paramètres.
4. Créer une interface utilisateur ergonomique et engageante en C++ sous Qt et/ou sous la forme d'une page Web.

A la fin de la semaine, vous ferez une présentation de votre travail à l'ensemble de la classe.

Spécifications techniques

Fonctionnalités réseau :

- **Serveur TCP :**
 - Gérer les connexions des joueurs.
 - Distribuer les personnages aléatoirement au début de chaque partie.
 - Synchroniser les étapes du jeu (écriture de mots, déductions, résultats).
- **Client TCP :**
 - Connecter chaque joueur au serveur.
 - Permettre à chaque joueur de recevoir et envoyer les données nécessaires pour jouer (mots, personnages...).

Base de données :

- Table Personnages :
 - **Colonnes** : id, nom, description.
- Table Scores :
 - **Colonnes** : id_partie, nom_joueur, score.
- Table Parties :
 - **Colonnes** : id_partie, date, nombre_joueurs.

Interface utilisateur :

1. **Écran d'accueil :**
 - Connexion au serveur.
 - Choix ou création d'un pseudo.
2. **Écran de jeu :**
 - Écriture des mots.
 - Consultation des déductions.
3. **Écran des résultats :**
 - Affichage des scores de la partie.
 - Classement général.
4. **Mode administrateur :**
 - Gestion des données des personnages via l'interface.

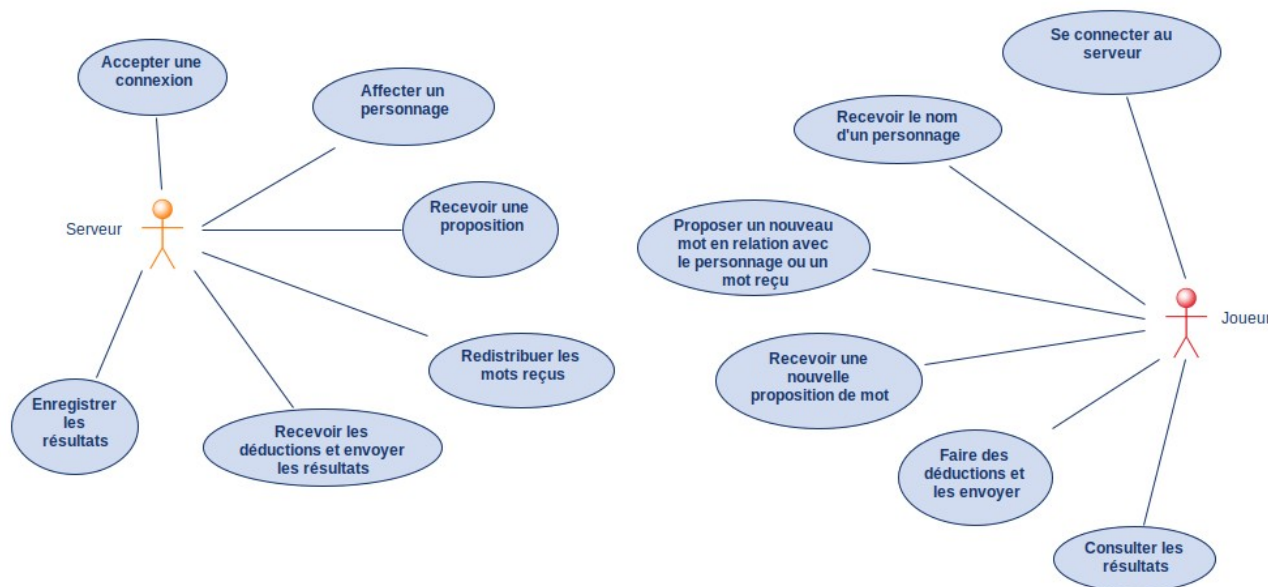
Analyse

Catalogue des acteurs

Joueur	Participant humain utilisant un client pour interagir avec le jeu.
Serveur	Composant central qui gère les connexions, les phases du jeu, et les communications entre les joueurs.

Diagramme des cas d'utilisation

Le diagramme ci-dessous présente une proposition de décomposition en cas d'utilisation pour de la version réseau du jeu. La modélisation inclut les acteurs principaux et les interactions attendues. Il doit permettre de vous répartir les tâches.



Descriptions des cas d'utilisation côté joueur

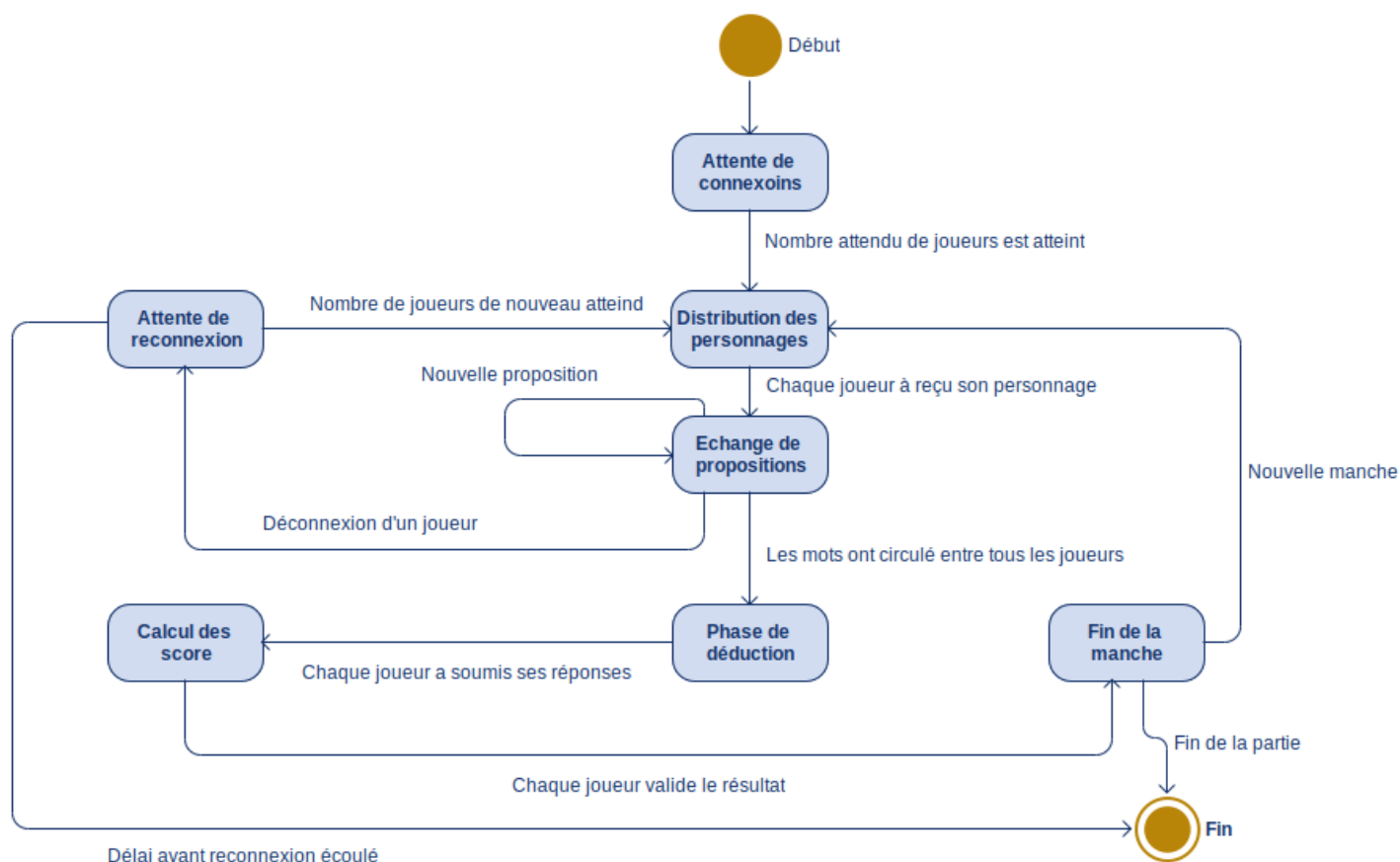
Cas d'utilisation	Description
Se connecter au serveur	Permet au joueur de se connecter au serveur avec un pseudo unique.
Recevoir le nom d'un personnage	Le joueur reçoit le nom d'un personnage unique pour la partie depuis le serveur.
Proposer un nouveau mot en relation avec le personnage ou un mot reçu	Le joueur propose un nouveau mot en fonction du mot qu'il a reçu d'un autre joueur. Ce mot est ensuite transmis au serveur.
Recevoir une nouvelle proposition d'un mot	Le joueur reçoit un mot modifié par un autre joueur.
Faire des déductions et les envoyer	Le joueur tente d'associer les mots finaux aux personnages proposés. Le joueur soumet ses choix pour les correspondances personnage/mot.
Consulter les résultats	Le joueur visualise les bonnes réponses et son score.

Descriptions des cas d'utilisation côté serveur

Cas d'utilisation	Description
Accepter une connexion	Accepte un joueur et initialise son pseudo et son état.
Affecter un personnage	Prends un personnage aléatoire dans la base de données et l'associe à un joueur. Chaque joueur reçoit un personnage unique par partie.
Recevoir une proposition	Reçoit un mot modifié et le stocke dans la partie en cours.
Distribuer les mots reçus	Transfère les mots entre joueurs pour les tours suivants.
Recevoir les déductions et envoyer les résultats	Collecte les réponses des joueurs associant les personnages aux mots finaux. Ensuite, compare les déductions avec les bonnes réponses et transmet les résultats de la manche.
Enregistrer les résultats	Enregistre les scores et l'historique de la partie dans la base de données.

Diagramme état transition du serveur

Le **Serveur** agit comme un médiateur central, garantissant que toutes les étapes du jeu sont synchronisées entre les joueurs. Ce diagramme état transition montre les différentes étapes du jeu.



Protocole de communication entre le serveur et les joueurs

Ce protocole n'est qu'une proposition. Il peut-être, modifié ou adapté selon les besoins de l'équipe de développement.

Structure des trames

Commandes du serveur vers les joueurs			
Lettre	Signification	Données attendues	Exemple de trame
P	Envoi d'un personnage	Un nom (ex : Hercules)	Taille + 'P' + "Hercules"
L	Envoi la liste des personnages	Des noms séparés par ' '	Taille + 'L' + "Hercules Napoléon Henri IV..."
M	Envoi d'un mot pour en déduire une nouvelle proposition	Un nom (ex : Force)	Taille + 'M' + "Force"
D	Envoi des mots pour faire les déductions	Des mots séparés par ' '	Taille + 'D' + "Force Corse Poule au pot..."
A	Annonce de la phase actuelle	Nom de la phase (ex : déduction)	Taille + 'A' + "Déduction"
R	Résultats de la manche pour affichage des résultats	Les résultats sont structurés sous forme d'associations personnage : mot . Chaque association est séparée par le caractère .	Taille + 'R' + "Hercules:Force Napoléon:Corse Henri IV:Poule au pot ..."
S	Nouveaux scores obtenus	Les scores sont structurés sous la forme d'association pseudo:score . Chaque joueur étant séparé par le caractère .	Taille + 'S' + "Robert:15 Albert:10 Alex:20 ..."
E	Message d'erreur	Description du message (ex : déconnexion, Pseudo déjà utilisé...)	Taille + 'E' + "Pseudo invalide"

Commandes d'un joueur vers le serveur			
Lettre	Signification	Données attendues	Exemple de trame
C	Connexion avec pseudo	Pseudo du joueur (ex. "Alex")	Taille + 'C' + "Alex"
P	Proposition d'un mot	Le mot reçu du serveur séparé du mot proposé par ' '	Taille + 'P' + "Force Énergie"
R	Réponse aux déductions	Les réponses sont structurées sous forme d'associations personnage : mot . Chaque association est séparée par le caractère .	Taille + 'R' + "Hercules:Force Napoléon:Corse Henri IV:Poule au pot ..."
Q	Quitter la partie	Aucune donnée	Taille + 'Q'
N	Nouvelle manche	Aucune donnée	Taille + 'N'

Détail d'une trame

Chaque trame sera constituée de trois parties principales, dans l'ordre suivant :

1. **Taille** (2 octets, quint16) : Spécifie la taille totale de la trame (en octets) à partir de la commande.
2. **Commande** (1 caractère, QChar) : Identifie le type d'opération ou d'information.
3. **Données** (QString contenant un mot ou une liste de mots séparés par un ou plusieurs caractères spécifiques) : Contiens les informations nécessaires pour exécuter la commande.

Flux de Communication

- **Connexion initiale**
 1. Le client envoie une commande 'C' avec son pseudo.
 2. Le serveur confirme avec 'P' pour envoyer le personnage initial ou 'A' pour indiquer l'attente.
- **Tour de Jeu**
 1. Les joueurs répondent avec 'P' en envoyant leur proposition.
 2. Une fois toutes les propositions reçues, le serveur redistribue les mots et envoie une commande 'M' contenant le mot à modifier.
- **Phase de déduction**
 1. Le serveur envoie une commande 'L' avec tous les personnages.
 2. Le serveur envoie une commande 'D' avec les mots obtenus à la fin de la manche.
 3. Les joueurs répondent avec 'R' en listant leurs choix.
- **Calcul et résultats**
 1. Le serveur compare les réponses et calcule les scores.
 2. Le serveur envoie à chaque joueur 'R' avec les associations personnage/mot de la manche
 3. Une commande 'S' est ensuite envoyée à chaque joueur avec son score.
 4. Le joueur répond avec 'Q' ou 'N' pour quitter le jeu, ou faire une nouvelle manche.
- **Gestion des erreurs :**
 1. Si un pseudo est déjà utilisé : le serveur répond avec 'E'.
 2. Si un joueur se déconnecte : le serveur informe les autres joueurs avec 'E'.

Avant chaque nouvelle phase du jeu, le serveur l'annonce avec la commande 'A'.

Méthode de travail

On vous propose d'utiliser la méthode **Kanban** qui est une approche visuelle de la gestion de projet. Elle vous aidera à structurer votre travail, à suivre vos avancements, et à collaborer efficacement en équipe.

Qu'est-ce que la méthode Kanban ?

La méthode Kanban repose sur l'utilisation d'un tableau divisé en colonnes représentant les différentes étapes d'un flux de travail. Chaque tâche est représentée par une carte (un post-it physique ou une carte virtuelle) qui passe de colonne en colonne jusqu'à être terminée.

L'objectif principal est de :

- **Visualiser le travail** : Identifier rapidement ce qui est en cours et ce qui reste à faire.
- **Limiter le travail en cours** : Ne pas surcharger l'équipe avec trop de tâches à la fois.
- **Améliorer la collaboration** : Partager une vision claire du projet.

Créer un tableau Kanban pour votre projet

Pour le projet *Fiesta de los Muertos*, voici comment structurer votre tableau Kanban :

Colonnes principales :

1. À faire ("To Do") :

- Liste des tâches à réaliser.
- Exemples :
 - Configurer le serveur TCP.
 - Implémenter la commande 'P' (envoi de personnage).
 - Développer l'interface client.

2. En cours ("In Progress") :

- Tâches actuellement travaillées par un membre de l'équipe.
- Exemples :
 - Ajouter la logique pour les déductions.
 - Tester la communication entre le client et le serveur.

3. En revue ("Review") :

- Tâches terminées, mais en attente de vérification ou de tests.
- Exemples :
 - Tester l'envoi de la liste des personnages.
 - Vérifier les résultats des scores.

4. Terminé ("Done") :

- Tâches complètement finalisées.
- Exemples :
 - Configuration initiale du serveur.
 - Intégration de la commande 'R' pour les résultats.

Rôles et organisation à adopter

1. Responsable Kanban :

- Une personne est en charge de mettre à jour le tableau et de s'assurer que les tâches avancent. Il Place les tâches les plus importantes en haut de la colonne "À faire".

2. Assignment des tâches :

- Chaque membre de l'équipe prend une tâche à la fois dans la colonne "À faire". Une personne ne doit pas avoir plus d'une tâche dans "En cours".
- Lorsqu'une tâche est terminée, elle passe dans "En revue".

3. Réunions courtes ("Stand-ups") :

- A la fin de chaque séance, discutez des avancements :
 - Qu'avez-vous terminé ?
 - Sur quoi travaillez-vous actuellement ?
 - Rencontrez-vous des obstacles ?

Quelques outils pour vous organiser

<https://www.beesbusy.com/>

<https://trello.com/home>