420-E80-CH DÉVELOPPEMENT D'APPLICATIONS (EXPERT)

Mini-Projet #1 - BattleShipLike

Groupe 00002

Boualem El Guendouz & Guillaume Barriault &
Alexandra Gérard
2025-09-03

MINI-PROJET #1 - BATTLESHIPLIKE

Présentation du projet

Contexte

Dans le cadre de notre cours de développement d'application (expert), nous avons la demande de réaliser un projet qui permettra de jouer au jeu de BattleShipLike en réseau. Cette application devra être joué par 2 joueurs, à tour de rôle.

Besoin

Cette application comporte différents besoins :

- Pouvoir jouer sur 2 ordinateurs différents (serveur/client)
- Utiliser les mécanismes de socket dans le but de réaliser les communications entre le client et le serveur
- Permettre la saisie des informations
- Une solution qui compile
- Gérer adéquatement les exceptions et effectuer les validations nécessaires au bon fonctionnement des traitements du programme.
- Les 2 joueurs devraient pouvoir effectuer autant de partie qu'ils le désirent et ce sans rétablir une nouvelle connexion.
- Aucun des 2 joueurs (client ou serveur) ne peut prendre unilatéralement une décision
- Lorsque le client décide de cesser de jouer (après une partie) le serveur se remet en mode attente
- Il ne peut y avoir qu'une seule connexion client concourante sur le serveur.
- Minimiser et standardiser les informations transmises entre le client et le serveur.

Exemple d'utilisation simple

Simon et Vincent veulent se faire une soirée "BattleShip" malgré la distance qui les séparent. Ils décident donc de faire quelques parties en utilisant l'application BattleShipLike.

Description de la demande

Objectifs

L'objectif principal du projet est de développer une application console de jeu Bataille navale ou plus communément appelé "BattleShipLike" qui devra répondre à plusieurs besoins. Parmi ceux-ci, il y a le fait que l'application console devra faire en sorte de pouvoir crée une connexion entre deux ordinateurs et leurs permettra de jouer à BattleShip entre eux.

Fonctionnalités du système

- La grille de jeu est de 4x4 (taille fixe pour le projet)
- Chaque joueur a un (1) seul bateau de taille 1x2
    - Les 2 « cases » occupées par le bateau doivent être contiguës (collées) (Pas de diagonale)
- Chaque joueur va jouer à tour de rôle et pourra attaquer 1x par tour
- Que le coup ait touché un BatlleShip ou non, le tour passera à l'autre joueur
- L'application sera mise à jour votre interface visuelle immédiatement après avoir envoyé votre « attaque »
    - Vous ne pouvez pas attendre « l'attaque » de l'autre joueur pour mettre à jour votre affichage
    - Ceci aura donc des impacts sur la mécanique
- La partie termine lorsqu'un joueur a touché les 2 emplacements du BattleShip de l'adversaire (bateau coulé)
- Le client peut décider de jouer une 2ieme partie ou de quitter le jeu

Critère d'acceptabilité

À la suite d'une entente avec le client, le livrable final respectera les points suivants :

- L'application permet de réaliser adéquatement les objectifs nommes ci-haut
- La conception de l'interface est ergonomique, professionnelle et simple a utilisé.

Solution proposée

Solution A – Application console

Avantages

- Ne nécessite pas une gestion de droits ou d'authentification
- Portabilité et distribution facile de l'application
- Développement plus rapide

Inconvénients

- Moins esthétique
- Requière un ordinateur par client

Solution B – Application Web

Avantages :

- Accessible de partout dans le monde
- Accessible à l'aide de n'importe quel dispositif électronique ayant accès à internet et d'un navigateur web

Inconvénients :

- Requiert d'être hébergé avec des frais associés
- Gestion de comptes et d'authentification.
- Besoin d'un serveur et de 2 clients qui communique avec le serveur afin de valider des requêtes.
- Peut ne pas être bien supporté par certains navigateurs.

Solution sélectionnée

Solution sélectionnée

Nous avons sélectionné la solution A (l'application console)

Argumentation

Cette solution est particulièrement intéressante pour le client puisqu'on a uniquement besoin de deux ordinateurs (un hébergeur et un client) pour jouer contrairement à la solution B qui requière un serveur web. De plus, la solution A coûte moins cher que la solution B.

Déroulement du projet

Planification

Tâches	Echéance
Création du cahier des charges	Mercredi 3 septembre 2025
Envoie du projet final – toute l'équipe	Vendredi12 septembre 2025

Répartition des tâches

Les tâches vont être réparties de la manière suivante :

- Une personne va s'occuper des modèles et du jeux
- Les deux autres personnes vont s'occuper des sockets en faisant du « pair programming »

Diagrammes

Diagramme de séquence

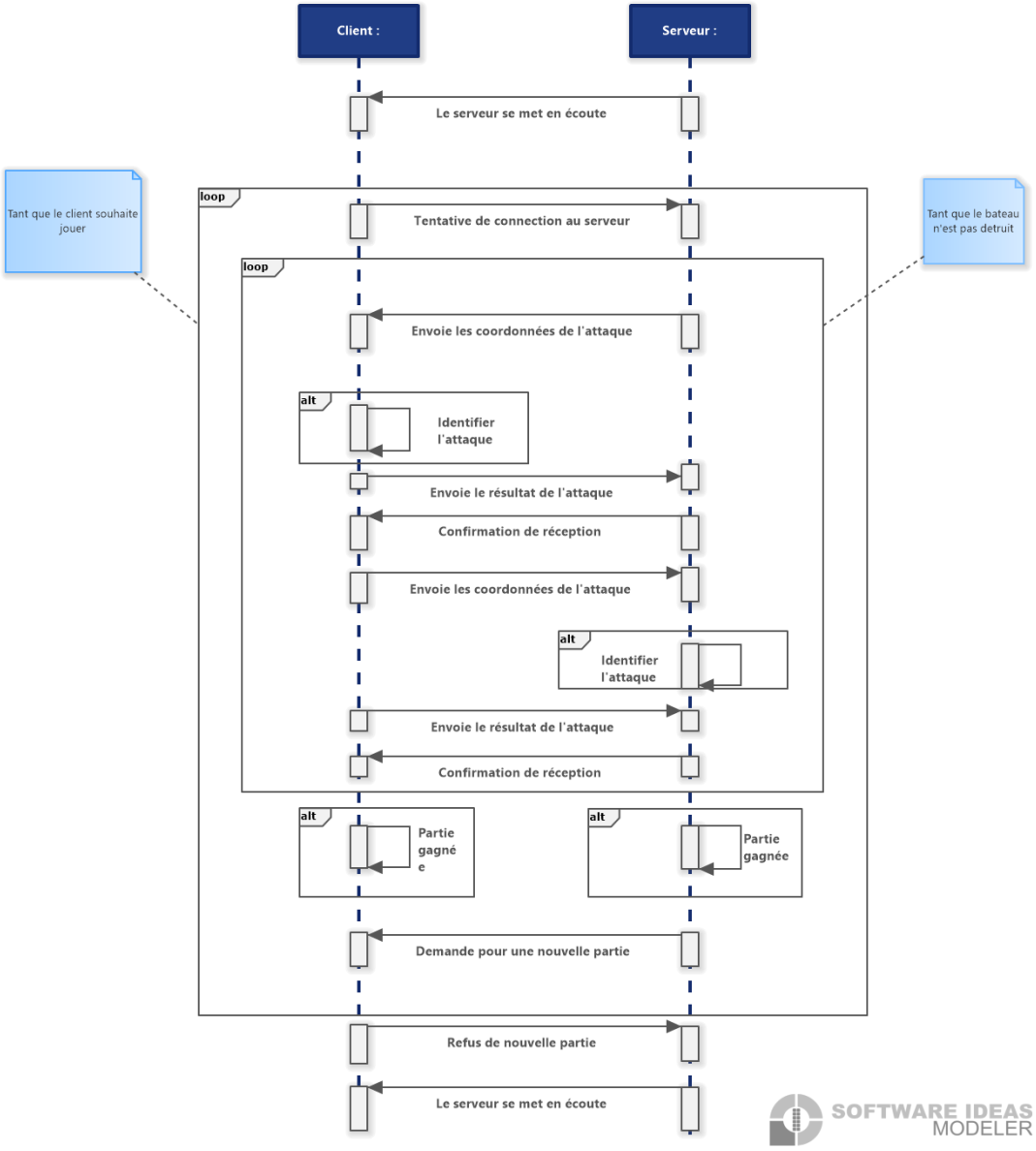
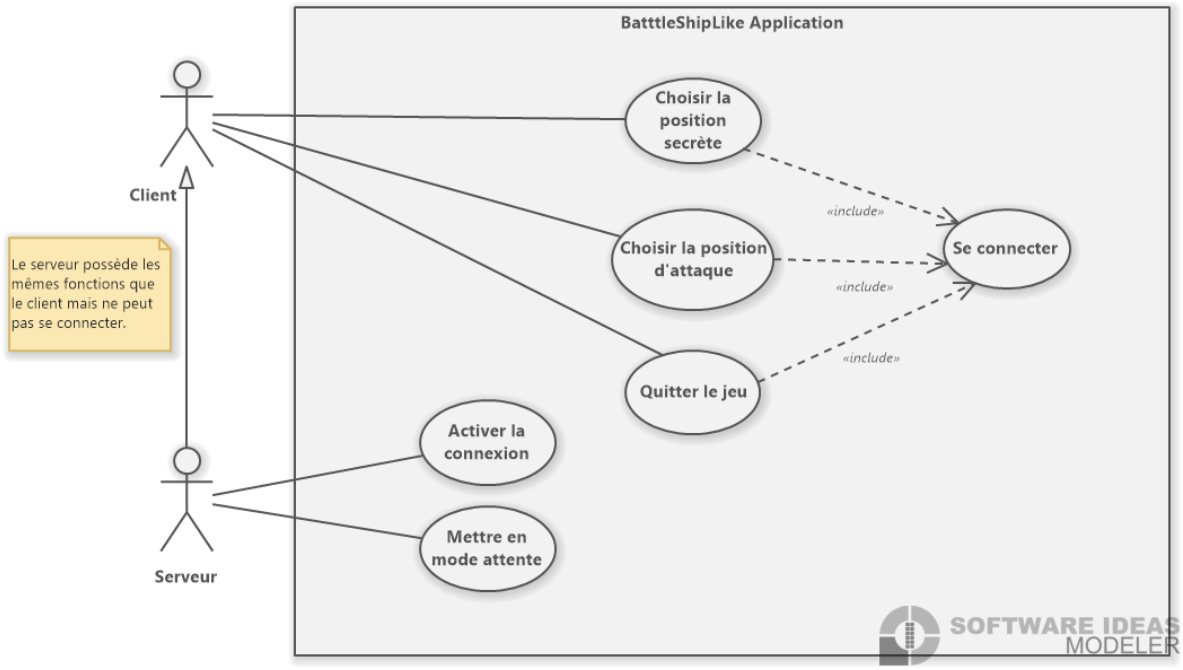


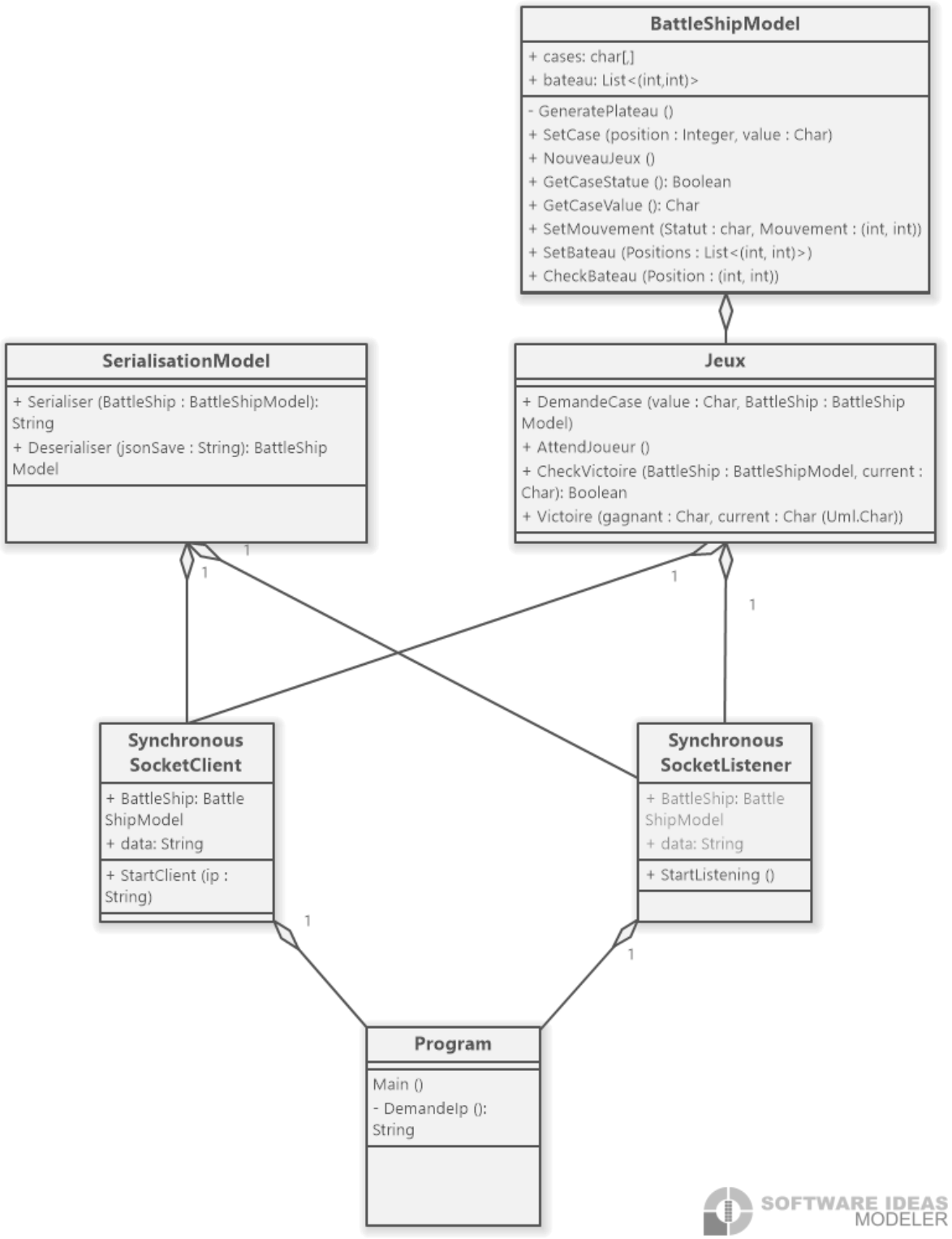
Diagramme de cas d'utilisation



Maquettes


Serveur (en attente)
Adresse ip du serveur : xxx.xxx.xxx
Serveur en attente ....
Client (en attente)
Entrez l'adresse ip du serveur : xxx.xxx.xxx
Serveur trouvé, mise en relation pour la partie !
Server (en jeu)
Choisissez ou mettre votre bateau a l'aide des flèches directionnelle :
_	_	_	_
_	_	_	_
_	_	_	_
_	_	_	_
Appuyer sur entrée lorsque vous avez choisi
Il sera possible de choisir a l'aide des fleches directionnelle l'emplacement du bateau
Client (en jeu)
Choisissez ou mettre votre bateau l'aide des flèches directionnelle :
_	_	_	_
_	_	_	_
_	_	_	_
_	_	_	_
Appuyer sur entrée lorsque vous avez choisi
Il sera possible de choisir a l'aide des fleches directionnelle l'emplacement du bateau
Server (en jeu)
Votre tour :
_	_	x	_
_	_	_	_
_	_	_	_
_	0	0	x
Appuyer sur entrée lorsque vous avez choisi
ou x equivaut a une case marqué comme "manqué"
et ou le carre est le bateau qui sera juste un fond de couleur differente
Client (en jeu)
Votre tour :
_	_	_	x
_	_	_	_
_	x	0	_
_	_	0	_
Appuyer sur entrée lorsque vous avez choisi
ou x equivaut a une case marqué comme "manqué"
et ou le carre est le bateau qui sera juste un fond de couleur differente
Client (fin de partie)
Vous avez gagné ! Félicitation !!!!!!
Voulez vous rejouer ?
SOFTWARE IDEAS MODELER

Diagramme de classes



Plan de test

#	Description	Précondition	Scénario	Résultats attendus
1	Sélectionner entre être client ou être serveur	L'application doit être lancée;	1. Sélectionner entre client et serveur;	L'ordinateur continue l'exécution du script selon le choix de l'utilisateur.
2	Se connecter à un serveur	L'utilisateur doit s'être connecté en tant que client;	1. Entrer l'adresse IP du serveur;	L'ordinateur client se connecte au serveur et le serveur lance la partie.
3	Sélectionner la position du bateau	Une partie doit être en cours;	1. Sélectionner la première case du bateau; 2. Sélectionner la deuxième case du bateau selon les cases disponibles;	La position du bateau est stockée
4	Sélectionner la case à jouer	Une partie doit être en cours; C'est à ton tour de jouer;	1. Sélectionner une case dans les cases disponibles;	Un message est envoyé à l'adversaire avec le statut et la position de l'attaque
5	Répondre à une action adverse	L'adversaire doit avoir envoyé une action avec le statut "attaque";	1. Vérifier si la case de "l'attaque" est une case où il y a ton bateau; 2. Vérifier si le joueur adverse a gagné; 3. Envoyer une réponse au joueur adverse	Un message est envoyé à l'adversaire avec la réponse de l'attaque
6	Afficher une victoire	L'adversaire doit avoir envoyé une action avec le statut "victoire";	1. Afficher un message de victoire; 2. Proposer de lancer une nouvelle partie;	Un message est affiché sur l'écran pour notifier le joueur de sa victoire
7	Afficher une défaite	L'adversaire doit avoir gagné la partie;	1. Afficher un message de défaite;	Un message est affiché sur l'écran pour notifier le joueur de la victoire de l'adversaire
8	Lancer une nouvelle partie	La partie précédente doit être terminée;	1. Lance une nouvelle partie;	Une nouvelle partie est lancé (la position du bateau et le plateau sont réinitialisés)
9	Quitter	Une connexion entre le client et le serveur doit être établie;	1. Le client quitte l'application	La partie s'arrête et le serveur se remet en attente de joueur.