

## [M2 GIL] Rapport .NET

---

**Formation :** Master 2 Génie Informatique Logiciel

**Université :** Université de Rouen Normandie

**Étudiants :**

Abderrahmane EL HATHOUT

Ibrahima KOUNDIA

Mahad MOUSSA ABDILLAHI

**Encadrants :** Chris Farin & Gauthier Cabot

**Année universitaire :** 2025–2026

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Présentation générale de la plateforme</b>	<b>2</b>
<b>3</b>	<b>Architecture globale de l'application</b>	<b>2</b>
<b>4</b>	<b>Technologies utilisées</b>	<b>2</b>
<b>5</b>	<b>Fonctionnalités générales</b>	<b>3</b>
<b>6</b>	<b>Jeu Morpion</b>	<b>3</b>
6.1	Principe du jeu . . . . .	3
6.2	Gestion des lobbies . . . . .	3
6.3	Communication temps réel . . . . .	3
6.4	Déroulement d'une partie . . . . .	3
6.5	Gestion des rôles . . . . .	4
6.6	Chat en temps réel . . . . .	4
<b>7</b>	<b>Jeu Puissance 4</b>	<b>4</b>
7.1	Principe du jeu . . . . .	4
7.2	Création et accès aux lobbies . . . . .	4
7.3	Gestion des joueurs et des rôles . . . . .	5
7.4	Architecture et logique serveur . . . . .	5
7.5	Communication temps réel avec SignalR . . . . .	5
7.6	Interface utilisateur . . . . .	5
7.7	Chat intégré . . . . .	6
7.8	Fin de partie et relance . . . . .	6
<b>8</b>	<b>Jeu SpeedTyping</b>	<b>6</b>
8.1	Principe du jeu . . . . .	6
8.2	Gestion des lobbies . . . . .	6
8.3	Communication temps réel . . . . .	7
8.4	Déroulement d'une partie . . . . .	7
8.5	Gestion des rôles . . . . .	7
8.6	Chat en temps réel . . . . .	7
<b>9</b>	<b>Difficultés rencontrées</b>	<b>8</b>
<b>10</b>	<b>Conclusion</b>	<b>9</b>

## 1 Introduction

Dans le cadre du Master 2 Génie Informatique Logiciel, ce projet a pour objectif la conception et le développement d'une plateforme de jeux multijoueurs en temps réel, basée sur les technologies .NET.

L'application, nommée **PlayRoom**, permet à plusieurs utilisateurs de créer ou rejoindre des lobbies afin de jouer à différents jeux multijoueurs directement depuis un navigateur web. Chaque jeu est conçu pour fonctionner en temps réel, avec une synchronisation immédiate entre les joueurs.

Ce rapport présente l'architecture générale de la plateforme ainsi que l'implémentation détaillée du jeu **Morpion**, développé individuellement dans le cadre de ce projet.

## 2 Présentation générale de la plateforme

PlayRoom est une plateforme web de jeux multijoueurs orientée temps réel. Elle repose sur un système de lobbies permettant aux joueurs de se regrouper avant de lancer une partie.

Les principales fonctionnalités de la plateforme sont :

- Création et partage de lobbies via un lien public
- Gestion des rôles (créateur du lobby / joueurs)
- Synchronisation en temps réel des actions des joueurs
- Interface web responsive accessible depuis un navigateur

Chaque jeu est implémenté comme un module indépendant, partageant une architecture commune pour la gestion des connexions et des lobbies.

## 3 Architecture globale de l'application

L'application repose sur une architecture client-serveur :

- Le **serveur** est développé en ASP.NET Core MVC
- Le **client** utilise HTML, CSS, JavaScript et Bootstrap
- La communication temps réel est assurée par **SignalR**

Les données liées aux parties et aux lobbies sont gérées **en mémoire** au sein du serveur, via des services applicatifs. Cette approche est adaptée au contexte des jeux multijoueurs, où les parties sont temporaires et ne nécessitent pas de persistance longue durée.

La logique principale de gestion des lobbies est centralisée dans un service singleton partagé par l'ensemble de l'application.

## 4 Technologies utilisées

Les technologies suivantes ont été utilisées pour le développement de la plateforme :

- **ASP.NET Core MVC** : structure de l'application web
- **SignalR** : communication temps réel entre clients et serveur
- **JavaScript** : gestion des interactions côté client
- **Bootstrap** : interface utilisateur responsive
- **C#** : logique métier et gestion des lobbies

Aucune base de données persistante n'a été utilisée dans l'implémentation actuelle. Les états de jeu sont stockés en mémoire et supprimés à la fin des parties.

## 5 Fonctionnalités générales

La plateforme offre plusieurs fonctionnalités communes à l'ensemble des jeux :

- Création d'un lobby par un utilisateur
- Génération d'un lien public permettant de rejoindre le lobby
- Attribution automatique du rôle d'hôte au créateur
- Lancement de la partie par l'hôte
- Synchronisation en temps réel des actions

Ces fonctionnalités constituent une base commune réutilisable pour l'ensemble des jeux proposés sur la plateforme.

## 6 Jeu Morpion

Le Morpion est un jeu de stratégie classique à deux joueurs. Chaque joueur dispose d'un symbole et joue à tour de rôle sur une grille de 3x3 afin d'aligner trois symboles.

Ce jeu a été implémenté comme premier module fonctionnel de la plateforme.

### 6.1 Principe du jeu

Deux joueurs s'affrontent en plaçant successivement leur symbole sur la grille. La partie se termine lorsqu'un joueur aligne trois symboles ou lorsque la grille est entièrement remplie, entraînant un match nul.

### 6.2 Gestion des lobbies

Lors de la création d'un lobby, un identifiant unique est généré. Un lien public est alors fourni au créateur, qu'il peut partager avec un autre joueur.

Le lobby accepte exactement deux joueurs :

- Le créateur du lobby devient l'hôte
- Le second joueur rejoint via le lien partagé

Une fois les deux joueurs connectés, l'hôte peut lancer la partie.

### 6.3 Communication temps réel

La synchronisation entre les joueurs est assurée par SignalR. Chaque action effectuée par un joueur est immédiatement transmise au serveur, puis diffusée à l'autre joueur connecté au lobby.

Cela inclut :

- Les mouvements de jeu
- Les changements d'état de la partie
- Les messages du chat

### 6.4 Déroulement d'une partie

Lors du lancement de la partie :

- Les symboles sont attribués aux joueurs
- Le joueur commençant est désigné
- La grille est initialisée

À chaque coup joué, le serveur vérifie la validité du mouvement, met à jour l'état du jeu et informe les deux joueurs du nouveau tour.

## 6.5 Gestion des rôles

Le jeu distingue deux rôles :

- **Hôte** : peut lancer et relancer la partie
- **Joueur** : participe à la partie sans contrôle du lobby

Ces rôles sont automatiquement déterminés lors de la création et de la connexion au lobby.

## 6.6 Chat en temps réel

Un système de messagerie instantanée est intégré au lobby du Morpion. Il permet aux joueurs d'échanger des messages pendant la partie.

Les messages sont transmis en temps réel via SignalR et affichés instantanément côté client.

# 7 Jeu Puissance 4

Le jeu **Puissance 4** constitue le second module de jeu multijoueur développé sur la plateforme **PlayRoom**. Il s'agit d'un jeu de stratégie à deux joueurs, reposant sur une grille verticale de 6 lignes et 7 colonnes.

Contrairement au Morpion, le Puissance 4 introduit une logique de jeu plus avancée, notamment en raison de la gestion de la gravité des jetons et de la multiplicité des conditions de victoire. Son implémentation a permis de valider la robustesse de l'architecture mise en place pour la plateforme.

## 7.1 Principe du jeu

Deux joueurs s'affrontent en insérant à tour de rôle des jetons de couleur dans une grille verticale. Lorsqu'un joueur choisit une colonne, le jeton tombe automatiquement dans la case la plus basse disponible de cette colonne, simulant un effet de gravité.

L'objectif du jeu est d'aligner quatre jetons de sa couleur :

- horizontalement
- verticalement
- ou en diagonale

La partie se termine lorsqu'un joueur remplit une condition de victoire, ou lorsque l'ensemble de la grille est remplie, entraînant un match nul.

## 7.2 Création et accès aux lobbies

La création d'une partie de Puissance 4 s'effectue depuis une page dédiée de la plateforme. Lors de cette étape, l'utilisateur doit choisir un pseudo, qui sera utilisé pour l'identifier tout au long de la partie.

Une fois la partie créée :

- un identifiant de lobby unique est généré côté serveur
- l'utilisateur devient automatiquement l'hôte du lobby
- un lien public est mis à disposition pour inviter un second joueur

Le second joueur accède à la partie via ce lien et choisit son propre pseudo avant de rejoindre le lobby. Cette approche garantit une gestion claire des identités et évite toute ambiguïté entre les joueurs.

### 7.3 Gestion des joueurs et des rôles

Chaque lobby de Puissance 4 accepte exactement deux joueurs. Lors de leur connexion, le serveur attribue automatiquement :

- une couleur de jeton (Rouge ou Jaune)
- un rôle au sein du lobby

Le premier joueur connecté devient l'hôte du lobby. L'hôte dispose de droits supplémentaires, lui permettant notamment :

- de lancer la partie lorsque les deux joueurs sont présents
- de relancer une partie une fois celle-ci terminée

Le second joueur rejoint le lobby en tant que joueur standard et attend le lancement de la partie par l'hôte. Ces rôles sont gérés exclusivement côté serveur et synchronisés en temps réel vers les clients.

### 7.4 Architecture et logique serveur

La logique du jeu Puissance 4 est entièrement centralisée côté serveur, afin de garantir la cohérence de l'état du jeu et d'éviter toute triche côté client.

Chaque partie est représentée par une structure en mémoire contenant :

- la grille de jeu (6x7)
- le joueur dont c'est le tour
- le statut de la partie (en attente, en cours, terminée)
- les informations des joueurs connectés

Lorsqu'un joueur joue un coup, le serveur :

- vérifie que la partie est en cours
- contrôle que le joueur joue bien à son tour
- applique la gravité du jeton dans la colonne choisie
- vérifie les conditions de victoire ou de match nul

L'état mis à jour de la partie est ensuite diffusé à l'ensemble des joueurs connectés au lobby.

### 7.5 Communication temps réel avec SignalR

La communication entre les clients et le serveur repose sur la bibliothèque **SignalR**, permettant une synchronisation en temps réel.

Chaque lobby correspond à un groupe SignalR distinct. Ainsi, les événements relatifs à une partie ne sont transmis qu'aux joueurs concernés.

Les échanges en temps réel incluent :

- la mise à jour de la grille de jeu
- les changements de tour
- les transitions d'état de la partie
- les messages du chat intégré

Cette approche assure une expérience fluide et cohérente pour les deux joueurs, sans rechargement de la page.

### 7.6 Interface utilisateur

L'interface utilisateur du Puissance 4 a été conçue pour être intuitive, moderne et cohérente avec l'ensemble de la plateforme PlayRoom.

Elle comprend notamment :

- une grille interactive représentant le plateau de jeu
- un indicateur dynamique de l'état de la partie

- une liste des joueurs présents dans le lobby
- des boutons contextuels adaptés au rôle du joueur

Les actions disponibles (démarrer, jouer, rejouer) évoluent dynamiquement en fonction de l'état de la partie et du rôle de l'utilisateur. L'interface est entièrement responsive et accessible depuis un navigateur web.

## 7.7 Chat intégré

Un système de messagerie instantanée est intégré directement au lobby du Puissance 4. Il permet aux joueurs d'échanger librement avant et pendant la partie.

Les messages sont transmis via SignalR et affichés instantanément côté client. Des messages système informent également les joueurs des événements importants, tels que l'arrivée d'un joueur ou le lancement de la partie.

## 7.8 Fin de partie et relance

À l'issue d'une partie, le résultat est affiché à l'ensemble des joueurs (victoire d'un joueur ou match nul).

L'hôte dispose alors d'un bouton **Rejouer**, permettant de réinitialiser l'état du jeu et de relancer immédiatement une nouvelle partie avec les mêmes joueurs, sans recréer de lobby.

Cette fonctionnalité améliore l'expérience utilisateur en favorisant l'enchaînement rapide de plusieurs parties.

# 8 Jeu SpeedTyping

Le jeu **SpeedTyping** est une course de dactylographie multijoueur dont l'objectif est de taper le plus rapidement et le plus précisément possible un texte imposé. Plusieurs joueurs rejoignent un même lobby et saisissent simultanément le texte, tandis que la plateforme calcule en temps réel leurs statistiques (vitesse, précision, score, progression).

Ce jeu met particulièrement en valeur la communication temps réel assurée par SignalR ainsi que la visualisation de l'avancée des différents joueurs dans une même interface.

## 8.1 Principe du jeu

Lorsqu'une partie de SpeedTyping commence, tous les joueurs reçoivent le même texte à recopier dans un temps limité (par exemple 60 secondes). Ils saisissent ce texte dans une zone dédiée, tandis que l'application compare chaque caractère saisi au texte de référence.

Pour chaque joueur, la plateforme calcule en continu :

- la **vitesse de frappe** (mots par minute)
- la **précision** (ratio de caractères corrects)
- un **score** global combinant vitesse et précision
- le **pourcentage de progression** dans le texte

À la fin du compte à rebours, les statistiques finales sont affichées à chaque joueur dans une fenêtre de résultat.

## 8.2 Gestion des lobbies

Comme pour les autres jeux de la plateforme, une partie de SpeedTyping débute par la création d'un lobby. Le joueur qui crée le lobby saisit un pseudo, puis l'application génère un identifiant unique et un lien public associé au lobby.

Les autres joueurs peuvent rejoindre la partie en cliquant sur ce lien et en renseignant à leur tour un pseudo. Le lobby affiche :

- la liste des joueurs connectés avec leur pseudo
- le nombre total de joueurs présents
- les paramètres principaux de la partie (par exemple la difficulté)

Le créateur du lobby conserve le rôle d'hôte et reste responsable du lancement de la partie lorsque tous les joueurs sont prêts.

### 8.3 Communication temps réel

La communication temps réel du SpeedTyping repose sur un hub SignalR dédié (*SpeedTypingHub*). Dès qu'un joueur rejoint un lobby, le serveur met à jour l'état du lobby et diffuse un message *LobbyUpdated* aux clients connectés, leur permettant de rafraîchir instantanément la liste des joueurs.

Lorsque l'hôte lance la partie, le hub :

- sélectionne un texte à taper (en fonction de la difficulté)
- définit la durée de la partie
- envoie à tous les joueurs un message *GameStarted*

Pendant la partie, chaque client envoie régulièrement sa progression (nombre de caractères saisis et corrects). Le hub agrège ces informations et émet un message *ProgressUpdated*, permettant à chaque joueur de visualiser l'avancée des autres participants en temps réel.

### 8.4 Déroulement d'une partie

Au lancement de la partie, chaque client reçoit le texte à taper, initialise un état local de jeu (*game state*) et démarre un compte à rebours. L'interface affiche :

- le texte cible, découpé en caractères et coloré selon la saisie
- un timer indiquant le temps restant
- les indicateurs de vitesse, précision et score
- une barre de progression sur le texte

À chaque frappe, le texte saisi est comparé caractère par caractère au texte cible, ce qui permet de colorer les lettres en fonction de leur exactitude (correct, incorrect, à venir). Les statistiques sont recalculées à partir du nombre de caractères corrects et du temps écoulé.

Lorsque le temps est écoulé, la partie se termine pour tous les joueurs et une fenêtre de résultat affiche les performances individuelles (vitesse, précision, score et nombre de caractères corrects).

### 8.5 Gestion des rôles

Dans le contexte du SpeedTyping, deux types de rôles sont distingués au sein d'un lobby :

- **Hôte du lobby** : joueur qui crée le lobby, choisit la difficulté et déclenche le lancement de la partie.
- **Joueurs** : participants qui rejoignent le lobby via le lien partagé et attendent le début de la partie.

Une fois la partie lancée, tous les joueurs sont soumis aux mêmes règles de jeu. Le rôle d'hôte n'influence plus la logique de calcul des scores, mais l'hôte conserve la possibilité de relancer une nouvelle partie dans le même lobby.

### 8.6 Chat en temps réel

Chaque lobby de SpeedTyping intègre un système de chat en temps réel, également basé sur SignalR. Les joueurs peuvent ainsi échanger des messages avant, pendant et après la partie.



Les messages sont envoyés au hub puis redistribués à l'ensemble des clients connectés au lobby. L'interface de chat affiche :

- le pseudo de l'expéditeur
- un avatar simplifié (initiale du pseudo)
- le contenu du message

Des messages système peuvent également être émis par le serveur pour informer les joueurs des événements importants (arrivée d'un joueur, lancement d'une partie, fin de partie, etc.).

## 9 Difficultés rencontrées

Le développement de la plateforme **PlayRoom** a été réalisé par un groupe de trois étudiants, chacun étant responsable de l'implémentation d'un jeu distinct. Cette organisation a permis de couvrir plusieurs jeux multijoueurs, mais a également imposé des contraintes en termes de temps et de coordination.

Une des principales difficultés rencontrées concerne la mise en place de **lobbies privés avec mot de passe**, nécessitant une architecture de persistance plus avancée. Une première réflexion a été engagée autour d'un modèle de base de données permettant de stocker les informations des lobbies, des joueurs et des accès sécurisés. Toutefois, cette approche impliquait un travail conséquent sur la conception du schéma de données, la gestion des accès concurrents et la sécurisation des informations.

En raison du nombre important d'UE à finaliser durant le semestre et des délais imposés, ce choix technique a finalement été abandonné afin de privilégier la stabilité et la complétude des fonctionnalités principales de la plateforme.

À la place, l'équipe a fait le choix d'investir ce temps dans des fonctionnalités améliorant directement l'expérience utilisateur, notamment :

- l'implémentation d'un **chat en temps réel** dans chaque lobby
- le soin apporté aux **interfaces utilisateur**
- la conception d'une application **responsive**, utilisable sur différents formats d'écran

Ce compromis a permis de livrer une plateforme fonctionnelle, cohérente et stable, tout en respectant les contraintes temporelles du projet.

## 10 Conclusion

Ce projet a permis de mettre en pratique les notions abordées durant le Master 2 Génie Informatique Logiciel, en particulier autour du développement d'applications web modernes et de la communication temps réel.

La plateforme **PlayRoom** constitue une base fonctionnelle pour des jeux multijoueurs en ligne, reposant sur une architecture claire et extensible. L'utilisation de **SignalR** s'est révélée particulièrement adaptée pour la synchronisation des actions des joueurs et la gestion des interactions en temps réel.

Chaque jeu développé a permis de valider la robustesse de l'architecture commune, tout en mettant en évidence des problématiques spécifiques selon la complexité des règles de jeu. L'ajout d'un système de chat et d'interfaces responsives a contribué à améliorer significativement l'expérience utilisateur.

Certaines évolutions restent envisageables, comme l'ajout de lobbys privés, la persistance des scores ou la mise en place d'un historique des parties. Ces fonctionnalités avaient été envisagées au départ, mais n'ont pas pu être implémentées dans le temps imparti. Le projet ayant été réalisé par une équipe de trois étudiants, chacun en charge d'un jeu, et mené en parallèle de plusieurs autres projets d'unités d'enseignement, nous avons fait le choix de nous concentrer sur une version fonctionnelle, stable et cohérente de la plateforme.

Dans l'ensemble, ce travail a permis de consolider les compétences techniques du groupe, tout en offrant une vision concrète des enjeux liés au développement d'applications collaboratives en temps réel.