

---

**<Équipe 101>**

---

**<Erratum>**  
**Plan de projet**

**Version 1.03**

## Historique des révisions

Date	Version	Description	Auteur
2023-09-05	1.0	Première itération	Etienne
2023-09-07	1.01	Contrôle de qualité	Nina
2023-09-08	1.02	Échéancier et entente contractuelle	Nina
2023-09-29	1.03	Complétion de certaines parties et révision finale	Mohamed Reda Rhanmouni, Nina Lounici, Etienne Desclaux, Nassour Nassour, Ghali Chraibi, Amirmasood Dadkhah

# Table des matières

<b>1. Introduction</b>	<b>4</b>
<b>2. Énoncé des travaux</b>	<b>4</b>
2.1. Solution proposée	4
2.2. Hypothèses et contraintes	4
2.3. Biens livrables du projet	5
<b>3. Gestion et suivi de l'avancement</b>	<b>5</b>
3.1. Gestion des exigences	5
3.2. Contrôle de la qualité	6
3.3. Gestion de risque	7
3.4. Gestion de configuration	10
<b>4. Échéancier du projet</b>	<b>10</b>
<b>5. Équipe de développement</b>	<b>13</b>
<b>6. Entente contractuelle proposée</b>	<b>15</b>

# Plan de projet

## 1. Introduction

L'objectif de ce document est de définir le cadre de gestion du projet en décrivant les tâches, les livrables, les méthodes utilisées ainsi que les délais. Nous débuterons par une description des travaux réalisés. Par la suite, nous présenterons le détail de l'organisation choisie par notre équipe pour gérer les exigences et leur qualité, le temps pris, les risques envisagés et les tâches associées au projet. En troisième partie, nous exposerons l'échéancier, suivi d'une présentation de notre équipe. Finalement, nous suggérons une entente contractuelle suite à l'appel d'offres.

## 2. Énoncé des travaux

### 2.1. Solution proposée

Nous proposons de faire une transformation d'une plateforme de jeu des différences en ligne développée dans le cadre du cours **LOG2990** en une plateforme de jeu des différences multiplateforme et multi-mode. La version originale web sera utilisée pour développer une version bureau ainsi qu'une version mobile avec de nouvelles fonctionnalités. La version déjà existante consiste en un mode classique et un mode temps limité, et dispose également d'une interface de création de jeu. Nous suggérons une base de jeu similaire, comprenant cependant diverses options supplémentaires : de nouveaux modes de jeu, un système de récompenses, des canaux de messagerie, de nouvelles fonctionnalités pour la création de parties. Nous garantissons également une excellente expérience utilisateur sur ordinateur comme sur tablette. Les fonctionnalités exactes de notre application sont détaillées dans le document de spécifications des requis du système.

### 2.2. Hypothèses et contraintes

- Notre équipe est composée de 6 étudiants engagés à fournir environ 12 heures de travail par semaine.
- Le projet doit être livré avant le 1er décembre 2023.
- Au niveau de l'équipement, nous supposons que le client lourd doit fonctionner sur Windows 10 et le client léger doit fonctionner sur Android 9.
- Les utilisateurs mobiles devront être sur Android.

- Nous supposons que les utilisateurs mobiles utilisent une tablette.
- Une sélection de technologies distinctes doit être faite entre les différents clients de l'application.
- L'apprentissage de nouvelles technologies doit être réalisé.
- Le serveur doit être hébergé sur un service (de préférence gratuit).
- Le langage de programmation du client lourd doit être différent de celui du client léger.
- Nous disposons d'un budget limité: Aucune ressource financière supplémentaire ne sera allouée en dehors du budget initial.

### **2.3. Biens livrables du projet**

Le 29 septembre 2023 à 23h59 au plus tard, nous devons remettre :

- Le plan de projet
- Le SRS
- Le protocole de communication
- Le document d'architecture logicielle
- Le prototype de communication du client lourd
- Le prototype de communication du client léger

Le 1er décembre 2023 à 23h59 au plus tard, nous devons remettre :

- Le plan de tests
- Les résultats de tests
- Les artefacts de la remise du 29 septembre, mis à jour
- Le code source
- Exécutable du produit final

## **3. Gestion et suivi de l'avancement**

### **3.1. Gestion des exigences**

Jira sera utilisé pour le suivi des exigences et de leurs modifications. Des tickets seront créés pour chaque tâche et assignés à différents membres de l'équipe. Pour chacun d'entre eux, un niveau de priorité est accordé.

Chaque exigence sera associée à un "epic" dans Jira et les tâches créées seront associées à chaque "epic" correspondant. Par exemple, des "épics" "Plan de projet", "Document d'architecture logicielle" (etc.) seront conçus.

Lorsqu'une exigence n'est pas considérée comme réalisable, les tâches qui lui sont associées seront mises de côté dans Jira, et ajoutées dans la colonne 'non-implémenté'.

Afin d'assurer la clarté et la pertinence des exigences, une revue de celles-ci sera effectuée régulièrement. Les membres de l'équipe discuteront des exigences au fur et à mesure de l'avancement du projet. Si des modifications sont nécessaires, elles doivent être approuvées par toute l'équipe avant d'être intégrées. Des réunions hebdomadaires avec toute l'équipe permettent de contrôler l'avancement de la réalisation des exigences.

Avant l'implémentation d'une exigence, une phase de validation est nécessaire pour s'assurer qu'elle est correctement implémentée. Cette validation sera effectuée par les membres de l'équipe responsables de l'exigence et par les autres membres afin d'avoir différentes perspectives.

### 3.2. Contrôle de la qualité

Moment	Méthodes utilisées
Début du projet : rencontre du 3 septembre	Dresser une liste complète de tous les biens livrables que l'on devra soumettre Survoler l'ensemble des documents pour que toute l'équipe ait une vision d'ensemble du travail à effectuer
1er point de contrôle après que les chargés aient validé toutes nos exigences	Compléter les documents maintenant que nous avons une liste d'exigences valides Faire un premier point sur notre avancée pour être sûrs de ne pas prendre de retard
2ème point de contrôle après la soumission du TP1 de LOG3000 (après le 14 septembre)	Apporter les modifications nécessaires aux documents en tenant compte de la correction de notre TP dans le but d'améliorer la qualité du SRS
3ème point de contrôle : 25 septembre, une semaine avant la remise	Revue de code par les membres de l'équipe sur la partie logicielle de la remise Test du bon fonctionnement

4ème point de contrôle: avant la première remise du 29 septembre	Contrôle de la qualité des livrables à remettre : absence de fautes d'orthographe, concision et clarté du texte, qualité du code, forme appropriée (respect de l'UML pour les diagrammes etc.)
Suite à la démo : 7ème semaine du projet	Apporter les corrections nécessaires en fonction du retour reçu
15 novembre au plus tard	Tests d'interfaces utilisateurs : faire jouer des utilisateurs externes au projet, sur mobile et ordinateur, prendre note de leurs critiques et améliorer le produit Tests des fonctionnalités et recherche de bogues

De plus, nous profitons des rencontres chaque semaine pour faire un contrôle de notre avancée, et corriger nos méthodes de travail si nécessaire : notamment en termes d'organisation et de gestion du temps.

Des tests seront effectués lors de la complétion de tâches et des tâches supplémentaires seront créées pour corriger les défauts.

### 3.3. Gestion de risque

La description des risques suit la convention suivante :

- Ampleur : sur une échelle de 1 à 10, 10 étant le risque le plus élevé. Cette analyse est basée sur la probabilité d'occurrence du risque, ainsi que ses impacts.
- Description : une description textuelle du risque ainsi que les problèmes attendus.
- Impact : échelle définissant la portée du risque
  - o C – critique (affecte le projet en entier)
  - o E – élevé (affecte les fonctionnalités principales du système)
  - o M – moyen (devrait être maîtrisable en appliquant une stratégie d'atténuation adéquate)
  - o F – faible (l'acceptation du risque est une stratégie envisageable)
- Facteurs : aspects (**métriques**) du système pouvant être compromis.
- Stratégie de gestion : mesures à prendre afin de gérer le risque.

1 - Sous-estimation d'une exigence souhaitable				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
3	Une exigence souhaitable est sous-estimée et prend plus de temps et de ressources à implémenter que ce qui était prévu.	F	Temps, ressources	Implémenter une autre exigence souhaitable.

2 - Délaissement de l'expérience utilisateur				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
4	L'expérience utilisateur est délaissée au profit de l'accomplissement des exigences.	M	Temps	Prendre en compte l'expérience utilisateur tout au long du projet. Définir une date maximum à laquelle le projet sera présenté à des utilisateurs externes pour tester l'expérience utilisateurs, assez éloignée de la remise pour pouvoir apporter des modifications.

3 - Sous-estimation d'une exigence nécessaire				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
8	Une exigence nécessaire est sous-estimée et prend plus de temps et de ressources à implémenter que ce qui était prévu.	E	Temps, ressources	Mettre cette exigence en priorité, assigner la tâche à plus de personnes



4 - Instabilité du serveur				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
5	Instabilité du serveur lorsqu'il y a beaucoup d'utilisateurs connectés en même temps, donc beaucoup de trafic.	E	Disponibilité, performance	Révision de la performance du serveur, réviser la qualité du code côté serveur. Attribuer une instance cloud plus puissante.

5 - Problèmes d'interface utilisateur				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
4	Des problèmes d'interface utilisateur qui mènent à une mauvaise expérience utilisateur	M	Temps pris par un usager pour compléter une tâche	Faire des tests utilisateurs pour avoir du feedback, mettre des développeurs sur l'expérience utilisateur, attribuer du temps pour s'assurer de la qualité de notre expérience utilisateur.

6 - Indisponibilité des membres de l'équipe				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
6	Un ou plusieurs membres de l'équipe sont indisponibles pendant une période prolongée (maladie, urgence personnelle, etc.).	M	Temps, efficacité, ressources	Mise en place d'un plan de redistribution des tâches du membre indisponible.

7 - Conflits dans l'équipe				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
6	Des désaccords ou des malentendus entre les membres de l'équipe qui peuvent ralentir la progression du projet.	M	Productivité, efficacité, ressources	Avoir une communication ouverte, avoir un médiateur pour aider à résoudre les conflits.

### 3.4. Gestion de configuration

Nous utiliserons Gitlab pour la gestion de versions du code source. Nous utiliserons différentes branches pour implémenter de nouvelles fonctionnalités et nous les fusionnerons avec la branche principale lorsqu' une nouvelle implémentation sera complète.

La branche master contient une version de l'application prête pour la production. La branche prototype est celle de développement. Nous utilisons des branches de types features pour implémenter les nouvelles fonctionnalités, et des branches de types bug pour régler des bogues. Ces branches seront nommées selon cette nomenclature:

- mobile/feature-nom-de-la-feature
- desktop/feature-nom-de-la-feature
- desktop/bug-nom-de-la-feature
- server/feature-nom-de-la-feature

La nomenclature des commits du projet sera comme ci-dessous:

[FIX/ADD/WIP] description-du-commit

### 4. Échéancier du projet

Nous avons 14 semaines complètes pour réaliser le projet. Si on se base sur un estimé de 4 heures de travail par crédit et par personne, nous arrivons à 12 heures par semaine par personne, pour un total d'environ 1008 heures.

Au total, nous avons 135 points de fonctionnalités à implémenter, ce qui correspond à environ 7.5 heures par point.

Sprints	Date de début	Date de fin	Principaux lots de travaux associés	Nombre d'heure estimé	
Sprint 1	28-08-23	29-09-23	Réponse à l'appel d'offres incluant : le SRS, le document d'architecture logicielle, le protocole	70h	200h

			de communication et le plan de projet			
			Prototype du client léger Prototype du client lourd		80h	
			Client léger : implémenter l'interface de base et certaines premières fonctionnalités afin de rattraper le retard sur le client lourd (qui contient une base grâce à LOG2990).		50h	
Sprint 2	30-09-23	27-10-23	Client lourd <b>Exigences essentielles</b>	Clavardage (canaux de discussions et intégration)	35h	195h
				Modes de jeu	60h	
				Vue administration et vue de création	38h	
				Reprise vidéo	30h	
				Compte utilisateur et historique	25h	
				Avatar	7h	
			Client léger : <b>Exigences essentielles</b>	Clavardage	45h	179h
				Modes de jeu	64h	
				Reprise vidéo	25h	
				Compte historique et utilisateurs	15h	
				Avatar	10h	
				Personnalisation de l'application	10h	
				Gif	10h	
Sprint 3	28-10-23	24-11-23	Client lourd :	Mode observateur	40h	194h

			<b>Exigences essentielles</b>	Personnalisation de l'application	10h	
			<b>Client lourd : Exigences souhaitables</b>	Persistance des configurations	7h	
				Mot de passe oublié	10h	
				Ajouter ses propres sons	10h	
				Clavardage-intégré	7h	
				Mode Free for All	34h	
				Noter les parties	14h	
				Chat vocal	40h	
				GIF dans les discussions	10h	
				Modération dans le chat	6h	
				Choisir les sons d'une liste de sons	6h	
			<b>Client léger : Exigences essentielles</b>	Mode observateur	20h	170h
			<b>Client léger : Exigences souhaitables</b>	Envoyer des photos dans le chat	15h	
				Choisir les sons d'une liste	10h	
				Personnaliser sons avec micro	20h	
				Noter les parties avec triages	10h	
				Modération dans le chat	10h	
				Chat vocal	28h	

				Notifications	21h	
				Partager sur les réseaux	14h	
				Mode Free For All	22h	
Sprint 4 : remise	25-11-23	01-12-23	Dernière révision du produit final		50h	70h
			Présentation orale		20h	

## 5. Équipe de développement

Notre équipe se compose de 6 développeurs, étudiants à Polytechnique Montréal en génie logiciel.

Etienne Desclaux - Développement web et serveur: Étudiant en troisième année d'ingénierie logicielle à Polytechnique, doté d'une solide expérience en développement et déploiement de logiciels industriels grâce à la réalisation de deux stages pertinents dans le domaine.

Ghali Chraibi - Développement web et serveur: Étudiant en quatrième année d'ingénierie logicielle à Polytechnique, Ghali apportera son expérience en programmation de solution efficace et performante, acquises durant ses expériences en industrie dans des contextes de programmation de logiciels à basse latence.

Mohamed-Reda Rhanmouni - Développement web et serveur: Étudiant en ingénierie logicielle ayant une expérience approfondie en développement full-stack de part ses stages en industrie. Il aidera pour l'application desktop et certaines exigences fonctionnelles tout en s'assurant d'avoir une solution robuste au niveau du serveur.

Nina Lounici - Développement mobile : Étudiante en ingénierie logicielle à Polytechnique, possédant une expérience en développement de logiciel acquise lors d'un stage spécialisé sur le développement front-end et l'optimisation de l'expérience utilisateur. Elle fera partie de l'équipe de développement mobile.

Amirmasood Dadkhah - Développement mobile : Étudiant en génie logiciel à polytechnique montréal, Amirmasood possède une riche expérience grâce à la réalisation de trois stages variés dans le domaine du développement logiciel. Il se spécialise en développement mobile et il aura un rôle dans la conception et l'optimisation de l'application mobile afin d'avoir une interface utilisateur fluide et intuitive.

Nassour Nassour - Développement mobile et serveur: Étudiant en ingénierie logicielle à Polytechnique, Nassour apportera son expertise en développement mobile acquise dans ses projets personnels précédents. De plus, Nassour pourra contribuer au développement du serveur à l'aide de ses connaissances en NodeJS acquises lors du projet 2.

## **6. Entente contractuelle proposée**

Conformément à l'appel d'offres, nous nous engageons à remettre une application de jeux des différences. Nous produirons une application de bureau ainsi qu'une application sur tablette, qui comprendront toutes les caractéristiques et fonctionnalités explicitées dans le SRS. Le client doit accepter que l'équipe peut choisir d'implémenter des exigences souhaitables différentes de celles présentées dans l'échéancier de ce plan de projet. Cependant, ces exigences doivent être décrites dans le SRS. L'équipe propose un contrat fixe pour ce projet.

Notre équipe doit respecter l'échéancier décrit et doit livrer le produit final le 1 décembre 2023 à 23h59 au plus tard. A cette date, nous remettrons également une version mise à jour du document d'architecture logicielle, du protocole de communication, du plan de projet ainsi que du SRS, un plan de test et les résultats des tests.

L'équipe est composée de six personnes. Pour chacun des développeurs, la charge de travail est estimée à 168 heures pour l'entièreté du projet, ce qui revient à un total de 1080 heures. L'équipe demande 100\$/heure. Le client s'engage donc à payer le montant de 100800\$ pour couvrir les frais du projet.

En cas de modification quelconque sur le contrat, le client doit avertir l'équipe et accepter un temps additionnel nécessaire aux changements de requis.