

---

**Équipe 109**

---

***Fais-moi un dessin***  
**Plan de projet**

**Version 1.3**

## Historique des révisions

Date	Version	Description	Auteur
2020-01-28	1.0	Première rédaction	Geneviève Laroche, Florence Cimon-Paquet, Annie Rochette
2020-02-05	1.1	Ajout de la section gestion de configuration	Florence Cimon-Paquet
2020-02-06	1.2	Rédaction de l'échéancier du projet	Geneviève Laroche, Annie Rochette
2020-02-07	1.3	Révision du plan de projet	Florence Cimon-Paquet

# 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	4
<b>3. Gestion et suivi de l'avancement</b>	<b>5</b>
3.1. Gestion des exigences	5
3.2. Contrôle de la qualité	5
3.3. Gestion de risque	5
3.4. Gestion de configuration	7
<b>4. Échéancier du projet</b>	<b>7</b>
<b>5. Équipe de développement</b>	<b>10</b>
<b>6. Entente contractuelle proposée</b>	<b>10</b>

# Plan de projet

## 1. Introduction

Ce document présente les grandes lignes du projet logiciel *Fais-moi un dessin*. Ces différentes sections porteront sur l'énoncé des travaux, la gestion du suivi et de l'avancement, l'échéancier du projet, une brève présentation de l'équipe de développement et l'entente contractuelle proposée.

## 2. Énoncé des travaux

### 2.1. Solution proposée

La proposition suggérée afin de répondre aux exigences du document de vision du projet *Fais-moi un dessin* est comme suit: un logiciel ayant une architecture client/serveur permettant l'envoi et la réception de données des clients au serveur et vice-versa. L'utilisation de WebSockets sera utilisée comme moyen de communication. Cette dernière permettra aux utilisateurs de communiquer avec le serveur et d'autres utilisateurs connectés au même serveur. Par la suite, ces informations seront stockées dans la base de données MongoDB. La solution logicielle devra être fonctionnelle sur tablette supportant minimalement Android 6 et sur un ordinateur supportant Windows 10, soit le client léger et le client lourd respectivement. Lors du développement logiciel, deux jalons sont établis: le 7 février pour la réponse à l'appel d'offre, et le 23 avril pour la remise du produit final. La gestion et le suivi de l'avancement seront documentés afin d'informer les membres de l'équipe et le client du progrès effectué de façon hebdomadaire.

### 2.2. Hypothèses et contraintes

Le projet sera développé en totalité par une équipe de 5 personnes. Chaque membre de l'équipe devra travailler un minimum de 6 heures par semaine sur le projet et un total de 900 heures-personnes est attendu pour la remise du produit final.

L'équipe devra s'équiper d'une tablette Android afin de pouvoir tester les différentes fonctionnalités qui seront implanté dans le client léger.

Le client lourd sera développé et testé dans un environnement Windows uniquement.

Un premier livrable, la réponse à l'appel d'offre et un prototype, seront remis le 7 février et le produit final sera remis le 13 avril 2020 pour respecter l'appel d'offre.

### 2.3. Biens livrables du projet

#### 1er livrable - 7 février 2020

- Artéfacts: Plan de projet, SRS, listes d'exigences, document d'architecture logicielle, protocole de communication
- Prototype de communication client lourd-serveur et client-léger serveur

#### 2e livrable - 13 avril 2020

- Artefacts: Artefacts du premier livrable mis à jour, plan de tests, résultats de tests
- Code source
- Exécutable du client léger, lourd et du serveur

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

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

Les exigences ont été sélectionnées par l'entière responsabilité de l'équipe et se basent sur les exigences décrites dans le SRS. Cette liste sera évaluée et approuvée par les chargés de laboratoire afin de s'assurer de répondre correctement aux critères de l'appel d'offre. Toute exigence sera mise dans l'outil de management Redmine. Chaque exigence sera subdivisée en sous-tâches dans le but de mieux gérer le suivi de la tâche en question. Un temps de travail estimé, un niveau de priorité et une description de la tâche seront associés à chacune d'entre elles dans le but de mieux prioriser les tâches à effectuer lors de la planification du calendrier. Il sera donc plus facile pour les développeurs de connaître la tâche à effectuer et de documenter son progrès. Toute modification apportée au produit final devra être validée par l'entière responsabilité de l'équipe et approuvée par les superviseurs du projet. Si elle est acceptée, cela entraînerait une modification des artefacts afin de garder une cohérence entre la documentation du logiciel et le logiciel en tant que tel.

#### **3.2. Contrôle de la qualité**

Afin de contrôler la qualité du produit, plusieurs étapes seront mis en place par l'équipe. L'équipe a donc mis en place des règles de programmation à respecter qui ont été mentionnées dans le SRS. Elle s'engage donc à respecter celles-ci afin d'assurer une certaine uniformité au sein du code.

Des tests unitaires seront effectués sur les fonctionnalités principales du système et les fonctionnalités avec un niveau de complexité plus élevé. Ceci permettra de vérifier si la fonctionnalité répond entièrement ou non aux exigences établies par l'équipe.

Des revues par les pairs seront effectuées de façon régulière dans le but de s'assurer de la cohérence et de l'uniformité du code au sein de l'équipe. Elles pourront également s'effectuer avant la mise en commun du code sur la branche master. L'équipe s'assurera donc suivre minutieusement la qualité de cette branche.

Lorsqu'une anomalie apparaît, elle doit être signalée aux autres membres de l'équipe le plus rapidement possible. Celle-ci devra être brièvement documentée afin de mieux comprendre la source de cette dernière. L'équipe se rencontrera afin de discuter de cette anomalie afin de la régler. La programmation par les pairs sera l'option privilégiée dans le cas où l'anomalie persiste.

#### **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
  - C – critique (affecte le projet en entier)
  - E – élevé (affecte les fonctionnalités principales du système)
  - M – moyen (devrait être maîtrisable en appliquant une stratégie d'atténuation adéquate)
  - 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 - Utilisation d'une librairie obsolète (SocketIoClientDotNet)				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
2	Nous utiliseront SocketIoClientDotNet pour la connexion du client lourd avec le serveur. Cependant, il s'agit d'une librairie obsolète.	F	Fiabilité Utilisabilité	Si nous rencontrons des problèmes avec cette librairie, il nous serait possible de modifier la façon dont nous gérons les sockets sans que ceci entraîne de très grandes modifications.

2 - Utilisation des sockets				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
3	Le logiciel utilisera les sockets pour la connexion des joueurs au logiciel. Si beaucoup de sockets sont connectées et que chacune effectue une quantité considérable de communication avec le serveur, ce dernier pourrait ralentir et voir planter.	M	Latence Disponibilité	En implantant la gestion de certaines des tâches du côté client, cela diminuera la charge des sockets et avoir beaucoup de sockets connectés ne causera pas de problème.

3 - Client lourd et client léger divergents				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
4	Le client léger et le client lourd seront implémentés par des développeurs différents, il est donc possible que l'interface logiciel de ceux-ci divergent au cours du projet.	M	Interopérabilité Esthétique	Il sera possible de faire une esquisse ou une maquette de l'interface au début du projet pour que les deux clients débutent avec les mêmes bases puis se consulter tout au cours du projet pour assurer une cohérence dans les interfaces.

4 - Diminution de la qualité de logiciel				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
3	En voulant une grande quantité d'exigences et de fonctionnalités dans notre logiciel, il y a un risque que la qualité de celui-ci soit affecté	M	Utilisabilité Fiabilité	Il sera possible d'y remédier en s'assurant d'avoir une base solide et un échéancier précis et adéquat

5 - Sous-estimer le temps de réparation des problèmes				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
6	Ceci pourrait provoquer des défaillances du système plus fréquentes. Cela pourrait également impacter le développement et l'échéancier de l'équipe.	E	Utilisabilité Fiabilité	À chaque sprint, s'allouer un temps afin de régler les problématiques. Ne pas surcharger les sprints et établir un temps dédié pour régler les problèmes.

### 3.4. Gestion de configuration

Afin de gérer la configuration de problème, un membre de l'équipe va ouvrir une nouvelle demande de type anomalie dans l'outil Redmine. Un sujet précis et concis devra lui être assigné dans le but de rapidement comprendre de quoi il s'agit par les autres membres de l'équipe. Une description du problème devra également être ajoutée afin de mieux saisir la problématique et son contexte. Un niveau d'importance devra lui être assigné pour rapidement évaluer l'importance de son impact. Il sera important de documenter les étapes effectuées pour régler l'anomalie.

Les artefacts du projets seront numérotés d'une façon à ce que les modifications apportées au document seront visibles dans le numéro de version. Chaque modification entraînera un changement dans la numérotation de la sous-version: 1.X. Lorsque le document aura été soumis, la numérotation du changement qui surviendra après la remise entraînera un changement dans le numéro de version: X.0.

Au niveau du code, l'équipe utilise un entrepôt GIT et l'outil de logiciel GitHub pour effectuer le contrôle des versions.

#### 4. Échéancier du projet

	Lot de travail	Effort estimé (heures-personnes)	Date de début	Date de fin
<b>Sprint 1 (14 janvier au 7 février)</b>				
Réponse à l'appel d'offre	Déterminer la liste d'exigences	20	14/01	06/02
	SRS	20	14/01	06/02
	Prototypes client léger/lourd	20	14/01	06/02
	Plan de projet	15	28/01	06/02
	Document d'architecture logicielle	15	28/01	06/02
	Protocole de communication	10	28/01	06/02
	Révision finale	5	06/02	07/02
<b>Sprint 2 (8 février au 28 février)</b>				
	<b>Client léger</b>			
	Dessin libre	40	16/02	26/02
	Ajout de canaux de discussion	20	08/02	16/02
	Affichage du profil et historique de l'utilisateur	15	08/02	16/02
	Révision	15	26/02	28/02
	<b>Client lourd</b>			
	Ajout de canaux de discussion	15	08/02	16/02
	Affichage du profil et historique de l'utilisateur	15	08/02	16/02
	Création du mode de jeu sprint solo	40	16/02	26/02
	Création mode de jeu (Manuelle 1)	40	16/02	26/02
	Révision	10	26/02	28/02
<b>Sprint 3 (29 février au 20 mars)</b>				
	<b>Client léger</b>			
	Mode de jeu sprint solo	40	29/02	07/03
	Liste d'amis	15	29/02	07/03
	Mode de jeu mêlée générale	40	07/03	18/03



	Plan de test	10	29/02	18/03
	<b>Client lourd</b>			
	Plan de test	10	29/02	18/03
	Liste d'amis	15	29/02	07/03
	Création mode de jeu (Manuelle 2)	20	29/02	14/03
	Création de jeu (Assisté 1&2)	40	07/03	18/03
	Mode de jeu mêlée générale	30	07/03	18/03
	Révision	10	18/03	20/03
<b>Sprint 4 (21 mars au 13 avril)</b>				
	<b>Client léger</b>			
	Mode de jeu Challenge	25	21/03	04/04
	Accomplissements	10	28/03	04/04
	Tutoriel	10	05/04	10/04
	Groupe d'effets	10	21/03	28/03
	Modification d'avatar	10	21/03	28/04
	Plan de tests	10	21/03	06/04
	Révision finale	15	06/04	12/04
	<b>Client lourd</b>			
	Plan de test	10	21/03	06/04
	Mode de jeu Challenge	20	21/03	04/04
	Groupes d'effets	10	21/03	28/03
	Modification d'avatar	10	21/03	28/04
	Personnalité des joueurs virtuels	10	28/03	04/04
	Accomplissements	10	28/03	04/04
	Tutoriel	10	05/04	10/04
	Révision finale	15	06/04	12/04
	Préparation pour la présentation du projet	15	06/04	12/04

## **5. Équipe de développement**

### **Maxime Bernier**

Maxime Bernier est un étudiant de troisième année en génie logiciel qui possède de l'expérience en développement de serveur. Sa principale responsabilité sera de développer le serveur du projet.

### **Geneviève Laroche**

Geneviève Laroche est une étudiante de troisième année en génie logiciel qui possède de l'expérience en développement front-end. Sa principale responsabilité sera de développer le front-end du client lourd.

### **Manyl Telmat**

Manyl Telmat est un étudiant de troisième année en génie logiciel avec de l'expérience en développement front-end et en développement de serveur. Il aidera à part égale à développer le serveur et le front-end du client lourd.

### **Florence Cimon-Paquet**

Florence Cimon-Paquet est une étudiante de troisième année en génie logiciel avec de l'expérience en développement front-end et possède de l'intérêt pour apprendre le développement mobile. Elle s'occupera principalement de développer le client léger.

### **Annie Rochette**

Annie Rochette est une étudiante de troisième année en génie logiciel avec de l'expérience en développement front-end et possède de l'intérêt pour apprendre le développement mobile. Elle s'occupera principalement de développer le client léger.

## **6. Entente contractuelle proposée**

L'entente contractuelle proposée est de type clé-en-main (prix ferme) et la date finale de remise du produit final est le 13 avril 2020. Ce type de contrat a été sélectionné puisque les caractéristiques du projet sont bien définies et que les clients connaissent les capacités des développeurs, soit des étudiants de 3e année en génie logiciel.

L'équipe s'engage à livrer le produit final le 13 avril 2020 avec l'intégralité des exigences essentielles et un certain nombre d'exigences souhaitables. Elle s'engage également à respecter les clauses et les modalités suivantes:

- Polytechnique Montréal se réserve le droit de rejeter la soumission si l'une des conditions suivantes n'est pas respectée:
  - La charge de travail ne respecte pas les contraintes spécifiées dans la charge de travail;
  - Le soumissionnaire ou l'un des employés est reconnu coupable en vertu de l'article 8 du document *Règlements des études du baccalauréat en ingénierie pour l'année 2019-2020*;
- L'équipe s'engage à respecter les droits de propriété intellectuelle;

Pour une équipe de 5 membres, un lot de travail de 900 heures-personnes est prescrit pour la réalisation du logiciel. Les différents taux horaires établis sont les suivants:

- 125\$/h pour le gestionnaire de projet
- 100\$/h pour un développeur

Le coût total du projet sera d'une valeur totale de 94 500\$.