



**Plan de projet**

**Version 1.1**

## Historique des révisions

Date	Version	Description	Auteur
2020-01-30	0.1	La rédaction initiale du document plan de projet	Pablo Chaussé-Cossio
2020-01-31	0.2	Rédaction de la section 2.3	Pablo Chaussé-Cossio
2020-02-04	0.3	Rédaction des sections 1 ,2.3 et 3.2	Pablo Chaussé-Cossio
2020-02-06	0.4	Rédaction des sections 2.1, 2.2, 3.1, 3.2, 3.4, 6.1,6.2	Pablo Chaussé-Cossio
2020-02-06	0.5	Rédaction de la section 3.3 sur la gestion de risque et de la présentation de Jacob Dorais	Jacob Dorais
2020-02-07	0.6	Rédaction de l'échéancier (section 4), de la section 6 (jumelage des sous sections 6.1 et 6.2) et de la présentation de Pablo Chaussé-Cossio	Pablo Chaussé-Cossio
2020-02-07	1.0	Correction du français et formatage	François-Xavier Legault
2020-04-12	1.1	Révision des sections 1,2,3.1,3.2,3.4,6.1,6.2	Pablo Chaussé-Cossio

# 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
2.2.1. ressources humaines	4
2.2.2. équipement	5
2.2.3. échéancier	5
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	6
3.4. Gestion de configuration	8
<b>4. Échéancier du projet</b>	<b>8</b>
<b>5. Équipe de développement</b>	<b>11</b>
<b>6. Entente contractuelle proposée</b>	<b>12</b>

# Plan de projet

## 1. Introduction

Ce document présente la planification du projet P'inturé, une application du jeu de *Fais-moi un dessin* en ligne. D'abord, la section 2 décrit la solution proposée, pose les hypothèses et les contraintes applicables et énumère les biens livrables du projet. Ensuite, la section 3 aborde la gestion des exigences, le contrôle de qualité, la gestion de risque et la gestion de configuration. Par la suite, la section 4 développe l'échéancier du projet. Finalement, la section 5 présente les membres de l'équipe de développement et la section 6 présente l'attente contractuelle proposée.

## 2. Énoncé des travaux

### 2.1. Solution proposée

La solution que nous proposons se nomme P'inturé. Du côté client, cette application consistera d'un client léger Android et d'un client lourd Windows. Le client léger sera conçu pour une tablette Galaxy Tab A et le client lourd devra fonctionner sur toute distribution PC de Windows 10. Pour le client lourd comme pour le client léger, l'application permet la création et la connexion à un compte personnel qui sauvegarde des statistiques et un historique d'utilisation. Une fois connectée, l'application permet de clavarder avec d'autres membres à travers des canaux de discussion. Ensuite, les deux clients offrent différents modes de jeux à partir desquels le joueur peut commencer une partie ou rejoindre une partie existante. Le client lourd se distingue du client léger, car il contient l'outil de création de parties. Avec cet outil, le joueur peut créer son propre *Fais-moi un dessin* à partir d'images qu'il choisit. Pour les deux clients, les parties se déroulent de la même manière. Un joueur dessine le mot qui lui est présenté alors que les autres tentent de deviner ce mot. Notre application permet l'ajout de joueur virtuel à une partie. Ces derniers peuvent dessiner automatiquement un dessin, mais ne participent pas en tant que joueur qui devine. De plus, les deux clients sont supportés par un même serveur. Ainsi, les joueurs mobiles peuvent affronter les joueurs PC. Pour accompagner les joueurs à travers ces fonctionnalités, un tutoriel est disponible pour les joueurs dans le client léger et dans le client lourd. Une exploration plus approfondie de notre solution se trouve dans le document de spécification des requis du système (SRS), qui explore l'entièreté des requis liés aux exigences de notre application.

### 2.2. Hypothèses et contraintes

#### 2.2.1. ressources humaines

L'équipe de développement est composée de 6 étudiants en génie logiciel à Polytechnique qui peuvent fournir 1080 heures-personnes au projet. Ces étudiants ont des expériences variables avec les technologies utilisées, une période de découverte et d'apprentissage sera donc nécessaire, particulièrement pour le développement d'application mobile en Kotlin.

### 2.2.2. équipement

Chaque membre de l'équipe a accès à l'équipement nécessaire au développement, c'est-à-dire, un ordinateur Windows. Par contre, aucun membre ne possède de tablette Galaxy tab A. Il faudra donc installer une machine virtuelle dans l'environnement de travail pour développer l'application mobile et emprunter une tablette pour l'équipe pour effectuer des tests. Le serveur sera distribué sur la machine d'un des membres de l'équipe. La machine qui héberge le serveur est muni d'un Intel i5 8600K (stock clock speed) et possède 16 GB RAM en Dual Channel (2400 MHz). La machine se trouve dans un réseau d'une capacité de 30 Mb/s en téléchargement et 10 Mb/s en téléversement. Le réseau internet sans-fil de Polytechnique Montréal supportera les communications avec les clients. Nous supposons qu'elle alloue un trafic de 10MBPS pour fournir une connexion stable et efficace.

### 2.2.3. échéancier

La réponse à l'appel d'offre est à soumettre le 7 février 2020 et le produit final est à rendre le 13 avril. Entre-temps, l'avancement du projet va être contrôlé par des rencontres hebdomadaires. Nous prévoyons d'adapter l'échéancier en considérant les charges de travail liées aux autres cours auxquels les membres de l'équipe sont inscrits.

## 2.3. Biens livrables du projet

7 février 2020, Réponse à l'appel d'offre:

- Liste d'exigences

- Plan de projet

- Spécification des requis du système (SRS)

- Document d'architecture logicielle

- Protocole de communication

- Prototype

13 avril 2020, Remise finale:

- Mise à jour des artefacts remis précédemment

- Plan de tests

- Résultats de tests

- Produit final incluant Client lourd (exécutable), Client Léger(code source), Serveur (code source)

## 3. Gestion et suivi de l'avancement

### 3.1. Gestion des exigences

D'abord, les exigences ont été déclarées dans le document de spécification des requis (SRS). Elles ont été définies à la suite d'une analyse du complément pédagogique et choisies par les membres

de notre groupe. À partir des exigences, nous définirons des items de travail et leurs sous-tâches. L'outil de gestion de projet Redmine nous permettra de faire un suivi de ces items et l'outil de gestion de projet Trello nous permettra de suivre l'évolution des sous-tâches. En utilisant ces outils, nous pourrions suivre l'évolution des exigences et décider des ajustements à apporter lors de rencontres hebdomadaires. Des exigences pourraient être appelées à changer, soit par la révision du client ou par un changement des priorités de l'équipe. Dans ce cas, elles devront être modifiées, nous contacterons le client pour lui présenter les changements et les faire approuver. Le document SRS pourra ensuite être modifié.

### 3.2. Contrôle de la qualité

Pour le contrôle de la qualité du prototype et du produit final, nous dotons notre environnement de production d'un outil d'intégration continue (Travis CI). Configuré sur le logiciel de contrôle de version GitHub, Travis CI nous permet d'assurer la qualité du code ajoutée en imposant une révision par les pairs avant d'intégrer du nouveau contenu au projet. Cette révision est nécessaire à chaque fusion d'une branche auxiliaire à notre branche principale. Ainsi, lorsqu'un problème survient, les pairs ayant fait la révision peuvent bloquer l'ajout du code et laisser un commentaire à l'auteur. Il peut alors entreprendre une action corrective et refaire sa demande de fusion. Nous avons aussi décidé de jumeler deux développeurs pour chaque livrable. Cette organisation nous permet de travailler en programmation par paires sur certains problèmes, ce qui contribue à l'amélioration de la qualité du code produit.

Pour assurer la qualité des artefacts, nous travaillerons à plusieurs sur tous les documents et aurons des rencontres dédiées à l'élaboration de notre architecture logicielle et de notre protocole de communication. Nous utiliserons aussi le logiciel de correction grammaticale Antidote pour assurer la qualité du français. Une action corrective peut être entreprise à tout moment suite à la détection d'un problème.

### 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.

01 - nouvelle technologie				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
6	Utilisation de nouvelles technologies pour les développeurs. Pour certain, c'est la première fois que nous utilisons kotlin et WPF. Cela pourrait amener du temps supplémentaire de développement	M	Temps planifié pour les tâches  Nombre de bogues  Nombre de fonctionnalités implémenté	Chaque développeur va ajouter du temps lors de l'estimation de temps des tâches pour servir de marge de sécurité.

02 - absence de membre				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
4	Il y a une possibilité que certains membres de l'équipe doivent s'absenter pour certaines réunions, ce qui pourrait avoir un impact sur l'efficacité de la communication de l'information.	M	Délai avant de pouvoir commencer nos tâches	Nous utilisons un serveur discord pour pouvoir partager l'information rapidement et efficacement.

03 - manque de fonctionnalités				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
3	Si nous n'arrivons pas à implémenter les fonctionnalités à temps, le client ne sera pas satisfait du produit livré ou mécontent d'avoir un délai pour recevoir son produit.	C	Nombre de fonctionnalités  Nombre de bogues	Nous ferons un suivi d'équipe les mardis à l'école où nous discuterons de l'avancement. De cette manière, nous pouvons diriger notre énergie où il le faut et nous pourrions nous rendre compte si une partie traîne plus.

04 - départ d'un membre				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
4	Un membre pourrait devoir partir de l'équipe de projet. Ceci mettrait plus de pression sur les membres restants	M	Temps planifié pour les tâches	Les développeurs sont responsables de faire du code lisible et d'afficher le suivi de leurs tâches sur redmine et le trello. De cette manière, si un membre doit quitter, il serait plus facile de savoir où était-il rendu.

			Nombre de tâches assignées par personnes	
--	--	--	--	--

05 - versions incohérentes				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
5	Si le client lourd est incohérent avec le client léger, ou que les communications entre client et serveur sont difficile dû à un manque de planification	E	Temps à ré-implanter des fonctionnalités  Nombre de temps nécessaire pour terminer	Avant de commencer à implémenter les requis du système, nous nous assurons de faire un plan de l'architecture désirée et un protocole de communication et que tous les membres ont lu et compris ces documents avant de concevoir les clients et le serveur, Si nous avons encore des problèmes d'incohérences, nous ferons une réunion d'équipe pour comprendre ce qui s'est passé et régler la situation.

### 3.4. Gestion de configuration

Lorsqu'un changement est nécessaire ou qu'un bogue est soulevé, un item RedMine est créé et assigné à quelqu'un. Ensuite, une branche dans le répertoire respectif "ftr/" ou "bug/" du git est créée. La soumission de changement se fait sur git. Les changements sont soumis à nos épreuves de contrôle de qualité avant d'être intégrés au projet. Les mises à jour sur la branche de master ne peuvent qu'être faites si la soumission de changement a été approuvée par au moins un développeur de l'équipe. Si un individu met à jour la branche directement sans avoir une branche pour fonctionnalité, celui-ci sera alors bloqué.

Pour les artefacts, les versions sont mises à jour sur un Google Doc, à chaque modification, l'auteur des changements doit mettre à jour le numéro de version manuellement. La version initiale est la version 0.1. Les suivantes incrémentent de 0.1 à chaque fois.

## 4. Échéancier du projet

Tâche	Effort (heures-personne)	Date de début	Date de fin
Réponse à l'appel d'offre	180h	2020-01-14	2020-02-07
Spécification des requis du système	36h	2020-01-14	2020-01-26



(SRS)v1			
Spécification des requis du système (SRS)v2	6h	2020-02-04	2020-02-07
Protocole de communication	10h	2020-01-14	2020-02-07
Plan du projet	10h	2020-01-14	2020-02-07
Document d'architecture logicielle	10h	2020-01-14	2020-02-07
Prototype de communication client léger - serveur	54h	2020-01-14	2020-02-07
Prototype de communication client léger - serveur	54h	2020-01-14	2020-02-07
Remise de la réponse à l'appel d'offre			2020-02-07
Produit final	900h	2020-02-11	2020-04-13
Clavardage - Intégrations	45h	2020-02-11	2020-02-17
Clavardage - Canaux de discussion (display unsub channels)	45h	2020-02-11	2020-02-17
Profil d'utilisateur et historique	35h	2020-02-11	2020-02-17
Interfaces:	45h	2020-02-18	2020-03-02

Choix du mode de jeux, lobby, zone de jeu			
Serveur: lobby, groupe de discussion public et privée, invitation groupe privée	40h	2020-02-18	2020-03-02
Client lourd: Creation de jeux	40h	2020-02-18	2020-03-02
Base de données: schémas pour les parties, les groupes, les modes de jeux	30h	2020-02-18	2020-03-02
Client lourd: Modes de jeux	50h	2020-03-03	2020-03-10
Serveur: Création de jeux	25h	2020-03-03	2020-03-10
Client léger: Modes de jeux	50h	2020-03-03	2020-03-10
Client: joueur virtuels	55h	2020-03-11	2020-03-16

Client léger: Tutoriel	20h	2020-03-11	2020-03-16
Serveur : modes de jeux	25h	2020-03-11	2020-03-16
Client lourd: tutoriel	20h	2020-03-17	2020-03-23
Client léger: effet sonore et visuel, amélioration du UX	25h	2020-03-17	2020-03-23
Serveur: communicati on en temps réels des dessins + joueurs virtuels	45h	2020-03-17	2020-03-23
Client léger: clavardage emoticones + notifications	45h	2020-03-24	2020-03-30
Client lourd: tutoriel	30h	2020-03-24	2020-03-30
Client lourd: effets visuels et sonores + éléments de design relatifs au niveau du profil	45h	2020-03-24	2020-03-30

Serveur: progression du profil	35h	2020-03-24	2020-03-30
Client lourd: amélioration du UX	20h	2020-03-31	2020-04-06
Client: Sprint COOP	35h	2020-03-31	2020-04-06
Client léger: tests d'utilisation de l'application	35h	2020-03-31	2020-04-06
Serveur: Sprint COOP	25h	2020-03-31	2020-04-06
Exigences souhaitables	35h	2020-04-07	2020-04-13
Remise du produit final			2020-04-13
Total	1080	2020-01-14	2020-04-13

## 5. Équipe de développement

### Jeremy Boulet

Étudiant de troisième année en génie logiciel à Polytechnique.

Expertise: Programmation full stack en développement web (Angular, NodeJs, Typescript, Javascript). Expérience en développement .NET Core, C++ et Java.

Responsabilités: Client Lourd, Serveur

### Pablo Chaussé-Cossio

Étudiant en 3e année à Polytechnique, Diplômé en Sciences Informatiques et Mathématique du

CÉGEP de Maisonneuve.

Expertise: programmation full stack dans le domaine du développement web (Angular/Node.js, Vue.js/C#), penchant pour la conception d'expérience utilisateur. Maîtrise des langages Java, C++/C, C#, JavaScript, TypeScript.

Responsabilité: Développement du client léger, organisation du projet, planification des sprints et répartition des tâches

### **Jacob Dorais**

Étudiant en génie logiciel rendu à sa troisième année. Expérience majoritairement en C++. En plus de son parcours scolaire, il a participé à un hackatowen où il fait une maquette pour un site qui centralise les places de stationnement dans les rues de Montréal. Il a participé à un game jam où il a fait un jeu de stratégie qui touche le style d'auto-chess et tower defense. Il a réalisé son stage chez Warner Bros Games à Montréal. Il était dans le département de gameplay et a travaillé pendant une session en C++ avec Unreal. Il a passé du temps pour explorer les commandes plus approfondies de git tel que le rebase.

Responsabilité: En charge du côté serveur.

### **Arthur Garnier**

Étudiant de troisième année en génie logiciel.

Expertise: développement d'interface utilisateur sur le web et mobile. Bonnes connaissances avec Angularjs, HTML ainsi que NodeJS.

Responsabilité: Maintenir et développer le client lourd. Faire des revues de codes.

### **François-Xavier Legault**

Étudiant de troisième année en génie logiciel.

Expertise: programmation dans le domaine du multimédia, spécifiquement dans le domaine du jeu vidéo (Unity et Unreal). Bonnes connaissances dans la programmation full stack dans le domaine du développement web ainsi que dans le langage de programmation C++/C.

Responsabilité: Interface et logique du client léger.

### **Duc-Thien Nguyen**

Étudiant de troisième année en génie logiciel. Expertise: programmation full stack dans le domaine d'application web (Angular, React, NodeJS, Typescript, JavaScript) et mobile (Android Studio, Java). Expérience en développement backend (Java, Spring-boot, C#) et développement de jeu (C#, Unity)

Responsabilité: Maintenir et développer le serveur et la base de données. Faire des revues de code sur les soumissions de fonctionnalité. Programmer légèrement sur le client léger.

## **6. Entente contractuelle proposée**

Cette proposition d'entente du système P'intùré contient plusieurs conditions formant ainsi l'ensemble des clauses de l'entente contractuelle proposée par l'équipe 110. L'entreprise PolyApps et l'équipe 110 se sont entendues sur les éléments suivants. Le type du contrat choisi par l'équipe 110 est un contrat fixe selon toutes les exigences mentionnées au niveau des documents de la livraison de l'appel d'offres. L'équipe 110 se doit de livrer le système P'intùré, les plans de tests et les résultats de tests au plus tard le 13 avril 20 à 23h59. Le projet est estimé à 1080 heures-personnes avec une équipe de développeur de 6 personnes aux taux horaires de 100\$/h. Suivant une entente et la remise du projet en bonne et due forme, le client s'engage à payer la somme de 108 000\$ afin d'obtenir le produit final, réparti sur l'ensemble de la durée du projet, afin d'octroyer aux développeurs la qualité de vie nécessaire au travail.