
Les Décalés

LNAH 2K17
Plan de projet

Version 1.7

LNAH 2K17	Version : 1.7
Plan de projet	Date : 2017-09-29

Historique des révisions

Date	Version	Description	Auteur
2017-09-18	1.0	Introduction, solution proposée, biens livrables du projet, hypothèse et contraintes (idées), gestion de risque	Pierre To
2017-09-19	1.1	Rédaction des sections : gestion des exigences, gestion de la qualité, gestion de la configuration et équipe de développement	Philippe Lelièvre
2017-09-20	1.2	Rédaction des hypothèses et contraintes, gestion des exigences, gestion de la qualité, échéancier du projet, tableau de l'équipe de développement	Pierre To
2017-09-20	1.3	Rédaction gestion des risques, gestion de la qualité, échéancier global et détaillé du projet	Philippe Lelièvre
2017-09-21	1.4	Rédaction gestion des risques, gestion de la configuration, échéancier global et détaillé du projet, entente contractuelle	Pierre To
2017-09-24	1.5	Rédaction d'une introduction pour gestion des risques, mise en forme des tableaux	Pierre To
2017-09-28	1.6	Mise en page	Philippe Lelièvre
2017-09-29	1.7	Relecture et mise en page	Pierre To

LNAH 2K17	Version : 1.7
Plan de projet	Date : 2017-09-29

Table des matières

1.	Introduction	4
2.	Énoncé des travaux	4
2.1	Solution proposée	4
2.2	Hypothèses et contraintes	4
2.3	Biens livrables du projet	5
3.	Gestion et suivi de l'avancement	6
3.1	Gestion des exigences	6
3.2	Contrôle de la qualité	6
3.3	Gestion de risque	6
3.4	Gestion de configuration	8
4.	Échéancier du projet	9
5.	Équipe de développement	13
6.	Entente contractuelle proposée	14

LNAH 2K17	Version : 1.7
Plan de projet	Date : 2017-09-29

Plan de projet

1. Introduction

Le plan de projet établit les bases de la gestion du projet d'évolution du jeu d'*air hockey*. Plus précisément, la section 2 décrit les travaux à réaliser. Il est question d'expliquer la solution proposée, d'énumérer les hypothèses et contraintes et de lister les différents artéfacts qui devront être créés. La section 3 aborde les méthodes de gestion et de suivi de l'avancement. Le document se concentre sur les méthodes de contrôle de la qualité et la gestion des exigences, de risque et de configuration. Pour finir, les trois dernières sections mettent respectivement en lumière l'échéancier du projet, l'équipe de développement et l'entente contractuelle. Ce document s'adresse principalement au promoteur, aux responsables du projet et à l'équipe de développement.

2. Énoncé des travaux

2.1 Solution proposée

De façon générale, ce projet consiste à faire évoluer un jeu d'*air hockey*. À la base, celui-ci se joue localement sur un ordinateur, à savoir le client lourd. La solution proposée consiste à étendre les fonctionnalités du produit. Les joueurs auront désormais la possibilité de jouer en réseau et d'éditer les zones de jeu à l'aide d'un iPad. Comme les modes de jeu et d'édition pourront s'effectuer en ligne, le programme offrira aux utilisateurs une interface de communication. De plus, un serveur fera le lien entre les clients lourds et légers. Toutes les fonctionnalités déjà présentes seront conservées ou améliorées alors que des nouvelles seront également ajoutées.

D'ailleurs, notre équipe propose d'ajouter des fonctionnalités associées à l'interaction entre les joueurs. Entre autres, une liste d'amis, un profil utilisateur, un classement, un mode spectateur sur le client léger et une intégration des réseaux sociaux permettront de rehausser l'expérience entre les joueurs. De plus, le mode de jeu sur le client lourd sera modifié pour améliorer la jouabilité. À la fin d'une partie, un utilisateur pourra cumuler des points. Il sera possible de les dépenser dans un magasin intégré au jeu permettant de changer l'esthétique de son équipement ou de modifier les mécaniques de jeu. Bref, l'ensemble des exigences fonctionnelles et non fonctionnelles est détaillé dans le document de spécifications des requis du système (SRS).

2.2 Hypothèses et contraintes

Étant un projet d'évolution de logiciel, il dépend de l'implémentation de la base, à savoir le jeu d'*air hockey*. Toutes les fonctionnalités déjà présentes et les raccourcis de claviers doivent demeurer fonctionnels. Nous supposons que les utilisateurs ont un débit Internet suffisant pour jouer confortablement au jeu.

2.2.1 Ressources humaines

L'équipe de développement est composée de six étudiants pouvant fournir au minimum 15 heures de travail par semaine. Le projet s'étalant sur 13 semaines, il est attendu que chaque membre consacre autour de 180 heures au total. Nous supposons que la composition de l'équipe ne changera pas durant le projet. De prime abord, tous les membres de l'équipe ne sont pas familiers avec le projet de base. Toutefois, deux des membres ont déjà développé à l'aide du même cadre lors de leur dernier projet.

LNAH 2K17	Version : 1.7
Plan de projet	Date : 2017-09-29

2.2.2 Équipement et environnement de déploiement

Tout d’abord, pour permettre le développement du projet, les ordinateurs du laboratoire L-4810 donnent accès aux environnements de développement Visual Studio 2017 et à XCode. Ainsi, dans le cas du client lourd, il est possible de déployer l’application sur Windows 10. De plus, un iPad de 4e génération est prêté à l’équipe pour permettre le déploiement sur iOS 10.3. Le réseau filial et sans-fil *eduroam* permettent l’implémentation d’un serveur. Nous supposons que le débit de téléchargement et de téléversement est suffisant pour assurer une communication fluide entre les clients. Pour faciliter la planification du développement et la compilation des heures travaillées, un accès Redmine est offert à l’équipe de développement. Pour permettre la gestion de versions, un répertoire Git est accessible à tous les membres.

2.2.3 Échéancier

L’équipe de développement a reçu les informations de l’appel d’offres le 31 août 2017. La date pour soumettre une réponse à l’appel d’offres est le 29 septembre 2017. Sur allocation du projet, l’équipe de développement propose l’entente contractuelle telle que spécifiée à la section 6. Le produit final doit être livré au plus tard le 29 novembre 2017. Les semaines de travail sont du jeudi au mercredi. L’équipe de développement s’engage à faire un suivi hebdomadaire avec le client. Redmine servira d’outil pour compiler les heures travaillées chaque jour et pour témoigner l’avancement du projet.

2.3 Biens livrables du projet

Les biens livrables du projet sont énumérés dans le tableau #1 ci-contre. Ce tableau détaille les différents livrables ainsi que leurs dates prévues de livraison.

Tableau #1 : Livrables du projet

Livable	Artéfact	Code source	Date de soumission
Éléments de planification	<ul style="list-style-type: none"> Calendrier de planification des tâches sur Redmine Feuilles de temps soumis sur Redmine 		Hebdomadaire (mercredi à 23h59)
Réponse à l’appel d’offres	<ul style="list-style-type: none"> Plan de projet Spécification des requis du système Liste d’exigences Document d’architecture logicielle Document du protocole de communication 	<ul style="list-style-type: none"> Prototype de communication client lourd-serveur Prototype de communication client léger-serveur 	29 septembre 2017
Suivi de projets	<ul style="list-style-type: none"> Évaluation par les pairs 		29 septembre 2017
Produit final	<ul style="list-style-type: none"> Mise à jour des artéfacts de la réponse à l’appel d’offres Plan de tests Résultats des tests 	<ul style="list-style-type: none"> Code source et exécutable du produit (client lourd, client léger, serveur) 	29 novembre 2017
Présentation du produit final			30 novembre 2017
Suivi de projets	<ul style="list-style-type: none"> Évaluation par les pairs 		30 novembre 2017

LNAH 2K17	Version : 1.7
Plan de projet	Date : 2017-09-29

3. Gestion et suivi de l'avancement

3.1 Gestion des exigences

Tout d'abord, la gestion des exigences du projet est basée sur deux documents importants. Le premier document est la liste d'exigences qui servira de barème pour le projet. Le deuxième document porte sur la spécification des requis du système (SRS). Ce document apporte un niveau de précision supérieur sur chacune des fonctionnalités que l'équipe doit d'implémenter.

De plus, pour ce qui est du suivi des exigences, cela se fera grâce à Redmine qui offre un ensemble d'outils de gestion de projet fortement utiles pour le développement logiciel. En effet, il permettra à l'équipe du projet d'avoir une bonne vue d'ensemble sur les fonctionnalités développées, en cours de développement ainsi que sur celles restantes.

Enfin, pour ce qui est de la gestion des changements, l'équipe établit d'abord une rencontre pour analyser l'impact du changement. Le changement est ensuite rapporté sur Redmine par une note sur la demande et informe le client des décisions à la prochaine rencontre hebdomadaire. Si le changement est majeur, nécessitant une réévaluation de la liste d'exigences ou des exigences fonctionnelles ou non fonctionnelles du SRS, l'équipe informe immédiatement le client par courriel.

3.2 Contrôle de la qualité

Tout au long du projet, l'équipe se rencontrera à chaque semaine pour une réunion lors de laquelle chacun des coéquipiers fera part de ses avancements. La réunion sera donc très importante pour tenir tous les membres du projet à jour sur ce qui a été fait et ce qu'il reste à faire. Ce sera aussi un bon moment pour émettre des commentaires constructifs si un coéquipier voit un problème dans la manière qu'un autre coéquipier code ou agit.

De plus, avant que du code puisse être envoyé sur la branche principale, les changements devront être révisés par un autre membre n'ayant pas participé à l'élaboration du code. Cela permettra au développeur du nouveau code d'obtenir une rétroaction constructive sur sa façon de programmer. De plus, cela permettra aussi au réviseur du code en question d'être familier avec des parties de l'application qu'il n'a pas développées. Ces revues de code auront lieu à chaque itération de développement pour toutes les demandes liées. Les actions correctives seront effectuées directement sur le code source, communiquées aux auteurs des fichiers modifiés et comptabilisées sur Redmine sous la même demande.

Un plan de tests logiciel sera conçu. Celui-ci détaillera les tests unitaires assurant ainsi l'implémentation de modules. Le plan de test couvrira aussi les tests d'acceptation pour chaque exigence permettant de vérifier le respect des besoins du client. Des tests d'utilisabilité pourront aussi déterminer si l'expérience utilisateur est satisfaisante et si les exigences non fonctionnelles sont respectées. Le plan de tests sera livré en même temps que le produit final. Les tests seront effectués à chaque version, à savoir jalon, tel que spécifiée à la section 4. Si une action corrective est nécessaire suite à la découverte d'un problème, celui-ci est répertorié sur Redmine en associant une nouvelle demande du type Anomalies liée à la demande dont le problème a été découvert.

3.3 Gestion de risque

L'analyse de la gestion du risque est détaillée à travers cinq tableaux. Chaque tableau mentionne l'ampleur, la description, l'impact, les facteurs et la stratégie de gestion d'un risque. À noter que l'ampleur est sur une échelle de 1 à 10, 10 étant le risque le plus élevé. De plus, l'impact est une échelle sur la portée du risque où les lettres C, E, M et F correspondent respectivement à un risque critique, élevé, moyen et faible.

LNAH 2K17	Version : 1.7
Plan de projet	Date : 2017-09-29

1 - Incompréhension du code source du projet de base

Ampleur	Description	Impact	Facteurs	Stratégie de gestion
8	Comme aucun membre de l'équipe n'a participé au développement du projet de base, il y a une courbe d'apprentissage associé au cadriciel, au code source et aux exigences. Il se peut aussi qu'il y ait des bogues dans le code existant.	M	Nombre de bogues	Mitiger Comme deux membres de l'équipe ont déjà travaillé avec le cadriciel, ceux-ci pourront partager leur expertise. Tous les membres doivent lire la documentation du code source, naviguer à travers celui-ci et poser les questions. La lecture du document de vision et du SRS du cours de INF2990 est également de mise.

2 - Problème avec la librairie SignalR (lié au socket)

Ampleur	Description	Impact	Facteurs	Stratégie de gestion
4	Les membres de l'équipe n'ont pas beaucoup d'expérience avec la librairie SignalR liée aux <i>sockets</i> . Il existe donc un risque de mal développer l'architecture de notre programme pouvant entraîner des interblocages ou bien du code incompréhensible.	M	Délai de réponse entre les clients et le serveur	Contourner En cas de problèmes graves entraînant l'incapacité de l'équipe d'avancer le développement, il faudra changer de méthodes de communication pour se diriger vers les <i>sockets</i> existants provenant de la librairie .NET.

3 - Utilisation du langage Swift

Ampleur	Description	Impact	Facteurs	Stratégie de gestion
3	Le langage Swift 3, utilisé pour le client léger (iPad), est relativement nouveau. Cela implique donc la possibilité que la documentation soit incomplète et que les exemples soient plus rares sur Internet.	M	Nombre de bogues Densité des commentaires	Mitiger Il faudrait utiliser des librairies iOS existantes, car elles sont normalement maintenues et testées.

LNAH 2K17	Version : 1.7
Plan de projet	Date : 2017-09-29

4 - Utilisabilité du produit final				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
2	Le produit final peut être rempli de bogues et s'avérer inutilisable. L'expérience utilisateur peut être dégradée par ces problèmes.	C	<p>Nombre d'exigences fonctionnelles complétées et valides</p> <p>Pourcentage de couverture de code par les tests</p>	<p>Mitiger</p> <p>Il faut s'assurer de faire des tests d'acceptation pour valider si les requis sont satisfaits. Il faut faire le suivi de tous changements avec le client lors des réunions chaque semaine.</p>

5 - Implémentation des exigences				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
2	Par manque de temps, il se peut que l'équipe de développement n'ait pas le temps d'implémenter toutes les exigences dans les délais prescrits.	E	<p>Nombre d'exigences fonctionnelles complétées et valides</p>	<p>Mitiger</p> <p>Il faut s'assurer de planifier les différents jalons et d'effectuer tout de suite les corrections nécessaires lorsqu'un retard se fait sentir. De plus, il faut être réaliste avec les objectifs de chaque itération.</p>

3.4 Gestion de configuration

Pour planifier les lots de travail et pour effectuer le suivi des tâches, l'outil Redmine est utilisé. Lorsqu'un problème sera détecté, il sera reporté sur Redmine sous forme d'une demande de type Anomalie. Le problème sera nommé suivant la syntaxe suivante « 3.X.X Description du problème » où 3.X.X correspond à la fonctionnalité dont le comportement est anormal. Nous associerons cette nouvelle demande à la fonctionnalité boguée par un lien « Bloqué par ». Enfin, on associera les soumissions Git associées à la résolution de cette anomalie sur Redmine.

De plus, vu que l'équipe adoptera l'outil de gestion de versions Git, celle-ci doit définir le modèle de branche qu'elle utilisera pour travailler de façon efficace. Un projet fonctionnel sera toujours disponible sur la branche principale, intitulée master, du serveur Git de l'équipe. Les autres changements plus risqués devront être envoyés sur la branche de développement qui sera nommé dev. Les changements seront donc plus fréquents sur la branche dev que master.

Chaque fonctionnalité nécessitera une branche à part dont le nom suivra le modèle suivant : « feat-3.X.X-description de l'exigence ». Celui qui sera responsable de réparer un bogue quelconque devra effectuer les changements dans une branche à part dont le nom devra suivre le modèle suivant : « fix-3.X.X-description du bogue ».

LNAH 2K17	Version : 1.7
Plan de projet	Date : 2017-09-29

4. Échéancier du projet

Le tableau #2 présente l'échéancier global du projet. Il est à noter qu'il y a quatre jalons : la réponse à l'appel d'offres, la remise d'un produit partiel, la remise du produit final et la présentation du produit final au client. La charge de travail requise pour la livraison du projet est de 1080 heures-personnes. De plus, les trois premiers jalons sont détaillés respectivement dans les tableaux #3, #4 et #5. Ainsi, l'équipe de développement aura un cycle de développement selon un modèle itératif avec des itérations d'une durée de deux semaines. Il est à noter que la 3e itération a une durée de trois semaines, car elle intègre la semaine de lecture. De plus, le temps estimé est mesuré en heures-personnes (hp).

Tableau #2 : Échéancier global du projet

# jalon	Description	Tâches principales	Temps estimé (hp)	Date de début (JJ-MM-AA)	Date de fin (JJ-MM-AA)
1	Réponse à l'appel d'offres	<ul style="list-style-type: none"> • SRS • Liste d'exigences • Document d'architecture • Plan de projet • Protocole de communication • Prototypes de communication 	215	31-08-17	29-09-17
2	Produit partiel <ul style="list-style-type: none"> • 65 % des exigences essentielles • 15 % des exigences souhaitables 	<ul style="list-style-type: none"> • Développement <ul style="list-style-type: none"> ○ Serveur ○ Site web ○ Client lourd ○ Client léger 	440	28-09-17	1-11-17
3	Produit final <ul style="list-style-type: none"> • 35 % des exigences essentielles • 85 % des exigences souhaitables • 100 % des exigences optionnelles 	<ul style="list-style-type: none"> • Développement <ul style="list-style-type: none"> ○ Site web ○ Client lourd ○ Client léger 	395	2-11-17	29-11-17
4	Présentation au client du produit final	<ul style="list-style-type: none"> • Présentation orale • Support 	30	23-10-17	30-10-17

LNAH 2K17	Version : 1.7
Plan de projet	Date : 2017-09-29

Tableau #3 : Échéancier complet du premier jalon

Tâches principales	Temps estimé (hp)	Date de début (JJ-MM-AA)	Date de fin (JJ-MM-AA)
Itération 1	100	31-08-17	13-09-17
Rédaction de la liste d'exigences	40		
Rédaction du document de la spécification des requis du système	40		
Implémentation du protocole REST API sur le serveur	10		
Implémentation d'une interface utilisateur pour le clavardage sur le client léger	10		
Itération 2	115	14-09-17	27-09-17
Rédaction du plan de projet	15		
Rédaction du document d'architecture de haut niveau	20		
Rédaction du protocole de communication	15		
Implémentation des <i>sockets</i> sur le serveur	10		
Implémentation d'une interface utilisateur pour l'authentification sur le client léger	5		
Implémentation de la fonctionnalité de clavardage sur le client léger	20		
Implémentation d'une interface utilisateur pour l'authentification et le clavardage sur le client lourd	10		
Implémentation de la fonctionnalité de clavardage sur le client lourd	20		

LNAH 2K17	Version : 1.7
Plan de projet	Date : 2017-09-29

Tableau #4 : Échéancier complet du deuxième jalon

Composant	Tâches principales	Catégorie d'exigence	Temps estimé (hp)	Date de début (JJ-MM-AA)	Date de fin (JJ-MM-AA)
Itération 3 + semaine de relâche			240	28-09-17	18-10-17
Client léger	Authentification (3.1)	Essentiel	5		
	Menu principal (3.2)	Essentiel	5		
	Édition des objets (3.10)	Essentiel	75		
	Déplacement de la vue (3.11)	Essentiel	25		
	Sauvegarde des cartes (3.13)	Essentiel	10		
Client lourd	Authentification (3.1)	Essentiel	5		
	Menu principal (3.2)	Essentiel	5		
	Jeu en ligne (3.5)	Essentiel	75		
	Gestion de la déconnexion (3.6)	Essentiel	20		
	Sauvegarde des cartes (3.13)	Essentiel	10		
Site web	Authentification (3.1)	Essentiel	5		
Itération 4			200	19-10-17	1-11-17
Client léger	Édition en ligne (3.9)	Essentiel	50		
	Configuration de la zone de jeu et des objets (3.12)	Essentiel	20		
	Clavardage (3.16)	Essentiel	20		
	Interface utilisateur - gesture (3.18)	Essentiel	15		
Client lourd	Édition en ligne (3.9)	Essentiel	50		
	Jeu en ligne (3.5)	Souhaitable	10		
	Confidentialité des cartes (3.8)	Essentiel	15		
	Clavardage (3.16)	Essentiel	20		

LNAH 2K17	Version : 1.7
Plan de projet	Date : 2017-09-29

Tableau #5 : Échéancier complet du troisième jalon

Composant	Tâches principales	Catégorie d'exigence	Temps estimé (hp)	Date de début (JJ-MM-AA)	Date de fin (JJ-MM-AA)
Itération 5			205	2-11-17	15-11-17
Client léger	Profil utilisateur (3.3)	Essentiel	10		
	Liste d'amis (3.4)	Essentiel	20		
	Confidentialité des cartes (3.8)	Essentiel	15		
	Mode spectateur (3.14)	Souhaitable	30		
	Effets visuels, sonores, matériels (3.19)	Essentiel	20		
Client lourd	Profil utilisateur (3.3)	Essentiel	10		
	Liste d'amis (3.4)	Essentiel	20		
	Mode spectateur (3.14)	Souhaitable	30		
Site web	Profil utilisateur (3.3)	Essentiel	10		
	Site web (3.20)	Essentiel	40		
Itération 6			190	16-11-17	29-11-27
Client léger	Présentation des cartes (3.7)	Essentiel	20		
	Édition en ligne (3.9)	Optionnel	5		
	Magasin (3.15)	Souhaitable	25		
	Chiffrement de la communication (3.17)	Optionnel	5		
	Tutoriel (3.22)	Essentiel	15		
Client lourd	Profil utilisateur (3.3)	Optionnel	5		
	Jeu en ligne (3.5)	Souhaitable	10		
	Présentation des cartes (3.7)	Souhaitable	10		
	Édition en ligne (3.9)	Optionnel	5		
	Magasin (3.15)	Souhaitable	25		
	Chiffrement de la communication (3.17)	Optionnel	5		
	Intégration des réseaux sociaux (3.21)	Optionnel	10		
	Tutoriel (3.22)	Essentiel	15		
	Mise à jour des artefacts de la réponse à l'appel d'offres		10		
	Plan de tests et résultats des tests		25		

LNAH 2K17	Version : 1.7
Plan de projet	Date : 2017-09-29

5. Équipe de développement

L'équipe de développement est composée de six individus. Le tableau #5 spécifie le domaine d'expertise et les responsabilités de chaque membre de l'équipe. Tous les membres de l'équipe sont responsables de documenter l'avancement de leurs tâches à travers Redmine, de notifier tous changements ou problèmes et de comptabiliser leurs heures de travail.

Tableau #6 : Composition de l'équipe de développement

Nom, Prénom	Expertise	Rôle	Responsabilités
Al-Romhein, Jean-Marc	<ul style="list-style-type: none"> Développement frontal et dorsal Infographie 	Développeur	<ul style="list-style-type: none"> Implémenter le client lourd (interface utilisateur et fonctionnalités) Implémenter le client léger (être un développeur hybride) Concevoir des interfaces utilisateurs
Ferland, Mikaël	<ul style="list-style-type: none"> Développement frontal et dorsal avec Java Connaissances des processus Agile 	Développeur	<ul style="list-style-type: none"> Implémenter le client léger (interface utilisateur et fonctionnalités) Valider l'expérience utilisateur Vérifier la qualité du produit (ex. respect des besoins)
Lelièvre, Philippe	<ul style="list-style-type: none"> Développement d'applications web en Java Infographie 	Développeur	<ul style="list-style-type: none"> Implémenter le client lourd (interface utilisateur et fonctionnalités) Concevoir l'architecture du système
Sghaïer, Michaël	<ul style="list-style-type: none"> Développement dorsal en C++ et Python Réseaux 	Développeur	<ul style="list-style-type: none"> Implémenter le serveur Implémenter le client lourd (fonctionnalités) Développer le protocole de communication entre les clients et le serveur
To, Pierre	<ul style="list-style-type: none"> Conception et développement d'un jeu vidéo Développement dorsal d'applications orientées objet Développement frontal d'interfaces utilisateurs Connaissance des activités de développement Agile 	Gestionnaire de projet	<ul style="list-style-type: none"> Développement du client léger (interface utilisateur et fonctionnalités) Organiser les rencontres formelles et informelles d'équipe (ex. <i>scrum</i>, revue de <i>sprint</i>, activité de cohésion) Planifier les différents lots de travail sur Redmine Assurer l'avancement des tâches par différentes plateformes de communication
Tourangeau-Laberge, Ariane	<ul style="list-style-type: none"> Développement d'applications web et bureau (C#) Réseautique Expérimentée en développement Agile 	Développeur	<ul style="list-style-type: none"> Implémenter le serveur Implémenter le client lourd (fonctionnalités) Développer le protocole de communication entre les clients et le serveur

LNAH 2K17	Version : 1.7
Plan de projet	Date : 2017-09-29

6. Entente contractuelle proposée

Nous proposons une entente contractuelle selon un contrat fixe, c'est-à-dire un contrat de livraison clé en main. Considérant l'échéancier du projet de la section 4, le produit final sera livré le 29 novembre 2017. Le prix demandé est de 112 500\$.

Le budget tient en compte les lots de travail détaillés à la section 4, totalisant 1080 heures-personnes pour l'équipe de développement mentionnée à la section 5. Plus précisément, chaque membre de l'équipe fournit 180 heures de travail réparties sur 13 semaines de développement. De plus, le budget se calcule en considérant un taux de 100\$/h pour un développeur et de 125\$/h pour un gestionnaire de projet.

Il est à mentionner que le contrat clé en main est préconisé par le fait que la liste des exigences a déjà été évaluée et validée avec le client. Également, il est à considérer que l'équipe de développement suppose que seul un suivi hebdomadaire avec le client est nécessaire, mettant ainsi en lumière le suivi minimal des travaux.

Sachant que PolyJeux Inc. s'intéresse au produit final afin d'évaluer le marché potentiel d'un jeu commercial, notre équipe concède les droits sur le produit final à PolyJeux Inc. pour un développement d'une version commerciale, une mise en marché et une distribution du système. Suite à la mise en marché de cette version commerciale, une nouvelle entente contractuelle sera négociée pour déterminer les redevances en fonction des ventes.