

---

# **RazorSharp Technologies**

---

## **Hockedu Plan de projet**

**Version 1.0**

Hockedu	Version : 1.0
Plan de projet	Date : 2013-02-07

## Historique des révisions

Date	Version	Description	Auteur
2013-02-08	1.0	Version initiale	Mathieu M-Gosselin
2013-02-05	1.1	Ajout de contenu dans plusieurs sections	Vincent Lemire

Hockedu	Version : 1.0
Plan de projet	Date : 2013-02-07

## 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	4
3.	Gestion et suivi de l'avancement	5
3.1	Gestion des exigences	5
3.2	Contrôle de la qualité	5
3.3	Gestion de risque	5
3.4	Gestion de configuration	6
4.	Échéancier du projet	7
4.1	Client lourd	7
4.2	Client léger	7
4.3	Serveur de jeu	8
4.4	Serveur maître	8
4.5	Serveur web	8
4.6	Jalons	8
5.	Équipe de développement	9
6.	Entente contractuelle proposée	10

Hockedu	Version : 1.0
Plan de projet	Date : 2013-02-07

# Plan de projet

## 1. Introduction

Ce document donne un aperçu global du plan de projet monté par l'équipe de développement. Il présente d'abord la solution proposée pour répondre au problème, en précisant les contraintes et les livrables. Il affiche ensuite tout ce qui a trait à la gestion du projet et au suivi de ses différents aspects. Finalement, il précise l'échéancier et décrit l'expertise de l'équipe de développement.

## 2. Énoncé des travaux

### 2.1 Solution proposée

Pour répondre à l'appel d'offre, l'équipe de développement propose un jeu de hockey sur coussin d'air jouable en réseau comportant les modules suivants :

- Un serveur de jeu permettant de gérer les parties de hockey
- Un serveur maître permettant la gestion des serveurs et de la connexion du client lourd
- Un serveur web permettant la gestion des profils utilisateurs et de la connexion du client léger, ainsi que la gestion des terrains
- Une application iPad permettant d'éditer des terrains de jeu et de les transmettre au serveur de jeu (client léger)
- Une application Windows permettant d'éditer des terrains de jeu et de jouer des parties (client lourd)

### 2.2 Hypothèses et contraintes

Ce plan de projet ainsi que la réussite de sa conception reposent sur les hypothèses suivantes :

- Les six membres de l'équipe demeurent disponibles tout au long de la session d'hiver 2013
- Le local L-4810 demeure à notre disposition lors des périodes réservées au projet (mardi et mercredi de 8h30 à 12h20)
- L'équipement informatique du local L-4810 demeure fonctionnel tout au long de la session d'hiver 2013

### 2.3 Biens livrables du projet

Le projet possède deux livrables :

- Prototype (date de remise : 8 février 2013)  
Ce livrable comprendra les artéfacts suivants :
  - Le plan de projet (présent document)
  - Le document d'architecture logicielle
  - Le document de protocole de communication
  - Le document de spécification des requis du système (SRS)
  - Un prototype démontrant la communication réseau du client léger et du client lourd
- Projet final (date de remise : 8 avril 2013)  
Ce livrable comprendra les artéfacts suivants :

Hockedu	Version : 1.0
Plan de projet	Date : 2013-02-07

- Le plan de tests logiciels
- Le document de résultats des tests logiciels
- Toutes les composantes fonctionnelles énumérées au point 2.1

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

#### 3.1 Gestion des exigences

L'outil utilisé pour la gestion des exigences est la plateforme web Redmine. Les fonctionnalités suivantes sont exploitées :

- Versions
  - Utilisées pour la gestion des livrables (prototype et projet final)
  - Permettent de faire un suivi de la progression générale d'un livrable (pourcentage des tâches accomplies)
- Demandes
  - Utilisée pour la gestion des tâches (une demande représente une tâche/activité pertinente pour le projet)
  - Une tâche doit être le plus unitaire possible pour faciliter le suivi de l'avancement
  - Une tâche créée doit être assignée à un ou plusieurs développeurs
  - Permet de gérer les priorités en assignant des cotes de priorité aux tâches

#### 3.2 Contrôle de la qualité

Voici les méthodes utilisées pour le contrôle de la qualité :

- Révision de code
  - Un développeur termine une tâche affichée sur Redmine
  - Il crée une tâche de révision de code et l'associe à la tâche qu'il vient de terminer
  - De préférence, il l'assigne explicitement à un autre développeur
  - Cet autre développeur (ou un volontaire, s'il n'a pas assigné la demande) révisé le code en comparant avec la révision précédente
  - Il signale tout problème, anomalie, mauvaise pratique de codage au développeur
  - Le développeur corrige les problèmes en cas de besoin
- Tests unitaires
- Tests d'interface

#### 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

Hockedu	Version : 1.0
Plan de projet	Date : 2013-02-07

- 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 – Affichage multithreading				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
7	La séparation en threads de la mise à jour du modèle logique du programme et de l’affichage graphique échoue, possiblement parce qu’elle entraîne des problèmes de synchronisation ou de blocage.	F	Affichage	Revenir à un affichage linéaire, mais moins performant.

2 – Intégration de Box2D				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
3	Box2D ne fonctionne pas	F	Physique	Box2D excluable du projet en une ligne pour basculer à la physique maison déjà éprouvée. (Mais légèrement moins fiable)

3 – Intégration réseau				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
8	Le système de réseautique	C	Réseau	Trouver une librairie de réseautique et l’approviser.

### 3.4 Gestion de configuration

La gestion de configuration s’effectue avec la logiciel Git :

- Permet la gestion de conflit
- Permet la création de plusieurs branches si des développeurs veulent expérimenter sans nuire au reste de l’équipe
- Est lié à la plateforme Redmine, ce qui facilite la gestion du temps passé sur les tâches et les révisions de code

Hockedu	Version : 1.0
Plan de projet	Date : 2013-02-07

## 4. Échéancier du projet

Étant donné que le projet est limité à 1080 heures-personnes, nous avons calculé que cela revenait environ à 3 heures par personne par jour, à raison de 5 jours par semaine. Ainsi, nos jours-personnes sont de 3 heures.

### 4.1 Client lourd

- Interface de base des menus, du mode jeu et pour communiquer avec les serveurs
  - Début : 10 février 2013
  - Fin : 23 février 2013
  - Effort estimé : 20 jours-personne
- Interface de base du mode édition
  - Début : 24 février 2013
  - Fin : 9 mars 2013
  - Effort estimé : 15 jours-personne
- Finition de l'interface
  - Début : 10 mars 2013
  - Fin : 23 mars 2013
  - Effort estimé : 10 jours-personne
- Communication de base avec les serveurs
  - Début : 10 février 2013
  - Fin : 23 février 2013
  - Effort estimé : 4 jours-personne
- Mode jeu en réseau
  - Début : 24 février 2013
  - Fin : 23 mars 2013
  - Effort estimé : 50 jours-personne
- Communication avec le serveur maître
  - Début : 10 mars 2013
  - Fin : 6 avril 2013
  - Effort estimé : 10 jours-personne
- Gestion des accomplissements (« achievements »)
  - Début : 10 mars 2013
  - Fin : 6 avril 2013
  - Effort estimé : 30 jours-personne

### 4.2 Client léger

- Interface usager
  - Début : 10 février 2013
  - Fin : 9 mars 2013
  - Effort estimé : 30 jours-personne
- Éditeur de terrain
  - Début : 24 février 2013
  - Fin : 9 mars 2013
  - Effort estimé : 20 jours-personne
- Communication avec le serveur web
  - Début : 10 mars 2013
  - Fin : 23 mars 2013
  - Effort estimé : 10 jours-personne

Hockedu	Version : 1.0
Plan de projet	Date : 2013-02-07

#### 4.3 Serveur de jeu

- Communication avec le serveur maître
  - Début : 10 février 2013
  - Fin : 23 février 2013
  - Effort estimé : 12 jours-personne
- Mode jeu en réseau
  - Début : 24 février 2013
  - Fin : 23 mars 2013
  - Effort estimé : 30 jours-personne

#### 4.4 Serveur maître

- Communication avec le serveur de jeu
  - Début : 10 février 2013
  - Fin : 23 février 2013
  - Effort estimé : 12 jours-personne
- Communication avec la base de données
  - Début : 10 février 2013
  - Fin : 23 février 2013
  - Effort estimé : 4 jours-personne
- Communication avec le client lourd
  - Début : 10 mars 2013
  - Fin : 6 avril 2013
  - Effort estimé : 10 jours-personne

#### 4.5 Serveur web

- Possibilité de créer un utilisateur
  - Début : 10 février 2013
  - Fin : 23 février 2013
  - Effort estimé : 1 jour-personne
- Communication avec la base de données
  - Début : 10 mars 2013
  - Fin : 23 mars 2013
  - Effort estimé : 10 jours-personne
- Interface web
  - Début : 10 mars 2013
  - Fin : 6 avril 2013
  - Effort estimé : 20 jours-personne

#### 4.6 Jalons

- 23 février 2013
  - Interface de base des menus, du mode jeu et pour communiquer avec les serveurs
  - Communication de base avec les serveurs
  - Communication entre le serveur maître et le serveur de jeu
  - Communication entre la base de données et le serveur maître
  - Possibilité de créer un utilisateur avec l'interface web
- 9 mars 2013
  - Interface usager du client léger
  - Mode édition pour les deux clients
- 23 mars 2013
  - Finition de l'interface du client lourd
  - Mode jeu en réseau
  - Communication entre le serveur web et le client léger
  - Communication entre la base de données et le serveur web



Hockedu	Version : 1.0
Plan de projet	Date : 2013-02-07

- 6 avril 2013
  - Communication avec le serveur maître
  - Gestion des accomplissements (« achievements »)
  - Interface web
  - Communication entre le serveur maître et le client lourd

## 5. Équipe de développement

L'équipe de développement est composée des 6 membres suivants, chacun possédant son expertise et ayant des responsabilités propres au sein du projet.

**Michaël Ferris**, témoignant d'une discipline et d'un leadership exceptionnels, est en charge du projet et s'occupe de gérer la direction générale que prendra l'architecture du code en ce qui a trait à la partie C++, qui constitue le cœur du projet. Puisqu'il a des antécédents en développement de jeux vidéo pour une entreprise, il s'occupe de gérer l'intégration de la librairie de simulation physique Box2D au projet. Il contribue également à l'élaboration du client de jeu en C#.

**Vincent Lemire** s'occupe de l'élaboration du prototype client-lourd en C#, en plus d'être le maître d'œuvre du client de jeu et de l'éditeur en C#. Son expérience précédente avec le C# fait de lui une référence en la matière au sein de l'équipe.

**Mathieu Parent** dirige la conception de l'architecture réseau puisqu'il a une expérience passée à travailler en bas niveau avec les échanges de paquets lors de l'élaboration de la base d'une librairie de réseautique. De plus, il possède une expérience de développement web, spécifiquement du côté client, qui sera utilisée lors du développement du portail communautaire de Hockedu.

**Charles Étienne Lalonde** possède de fortes habiletés avec le langage Python ce qui fait de lui un atout pour tout ce qui se rapproche au *scripting*, qui sera notamment utilisé pour gérer la flotte de serveurs *master* distants. Il consacre également beaucoup de son temps à l'élaboration des fonctionnalités de réseautique.

**Samuel Ledoux** est en charge du développement dans l'environnement Macintosh, ce qui n'est pas sans inclure les appareils roulants sous iOS. Il contribue au développement de l'éditeur de cartes sur iPad, mais joue également un rôle essentiel dans l'élaboration du prototype client-léger sur la même plate-forme. Son expérience de développement sur Mac (contribuant à une suite bureautique populaire, sur cette plate-forme) est indispensable et sa forte appréciation de cet environnement de développement est un atout pour ses collègues qui seraient en difficulté.

**Mathieu Marengère-Gosselin** s'occupe de gérer le bon déroulement des activités et d'effectuer la préparation et le suivi de ces dernières. Il est en charge de la rédaction des divers documents nécessaires au bon avancement du projet. Sa facilité à s'exprimer et son niveau d'organisation exemplaire font de lui la personne désignée pour ces tâches. Finalement, il contribue de façon importante au développement de l'éditeur de cartes sur iPad.

Hockedu	Version : 1.0
Plan de projet	Date : 2013-02-07

## 6. Entente contractuelle proposée

L'équipe de développement propos un contrat de type clé en main.

Le prix du contrat s'élève à 101500\$.

Le produit final sera livré au client en date du 8 avril 2013 à 23h55.