

Projet de Programmation  
Création d'IA pour Rasende Roboter  
Analyse des besoins

Olivier Braïk  
Alexandre Delesse  
Gaëtan Lussagnet  
Alexandre Mourany  
Dimitri Ranc

8 février 2013

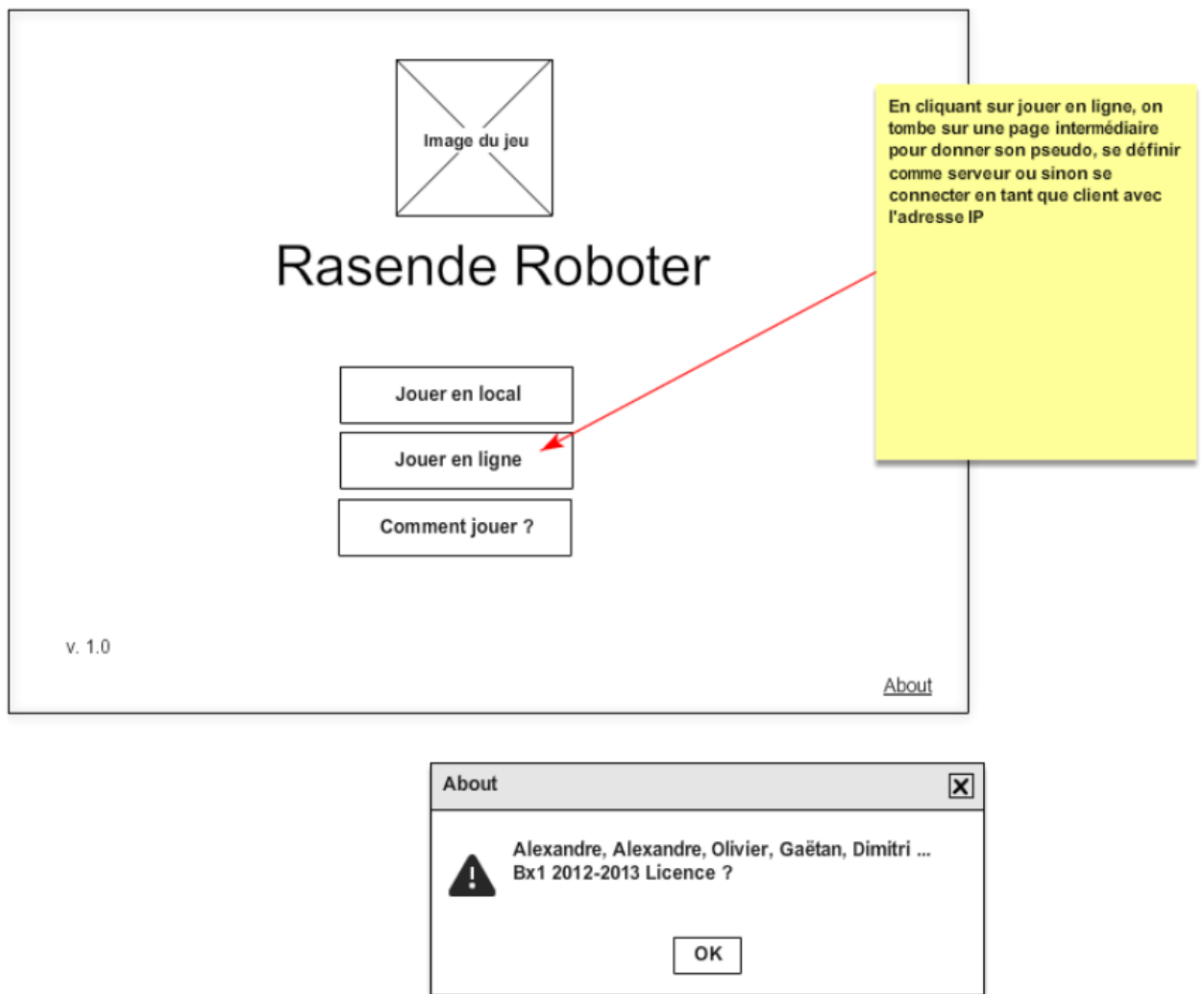
# Table des matières

<b>1</b>	<b>Scénario</b>	<b>3</b>
1.1	Accueil . . . . .	3
1.2	Création terrain . . . . .	4
1.3	Jeu . . . . .	5
<b>2</b>	<b>Besoins fonctionnels</b>	<b>6</b>
2.1	Écran d'accueil . . . . .	6
2.2	Jouer en local . . . . .	6
2.2.1	Création du terrain . . . . .	6
2.2.2	Déroulement de la partie . . . . .	6
2.3	Jouer en ligne . . . . .	7
2.3.1	Sélection du serveur de jeu/Création d'un serveur . . .	7
2.3.2	Création de la partie . . . . .	7
2.3.3	Déroulement de la partie . . . . .	8
<b>3</b>	<b>Besoins non-fonctionnels</b>	<b>9</b>
3.1	Jouabilité . . . . .	9
3.2	Systèmes d'exploitation . . . . .	9
3.3	Réutilisation . . . . .	9
<b>4</b>	<b>Priorités</b>	<b>9</b>
<b>5</b>	<b>Planning Gantt</b>	<b>10</b>
<b>6</b>	<b>Tableau des risques</b>	<b>12</b>

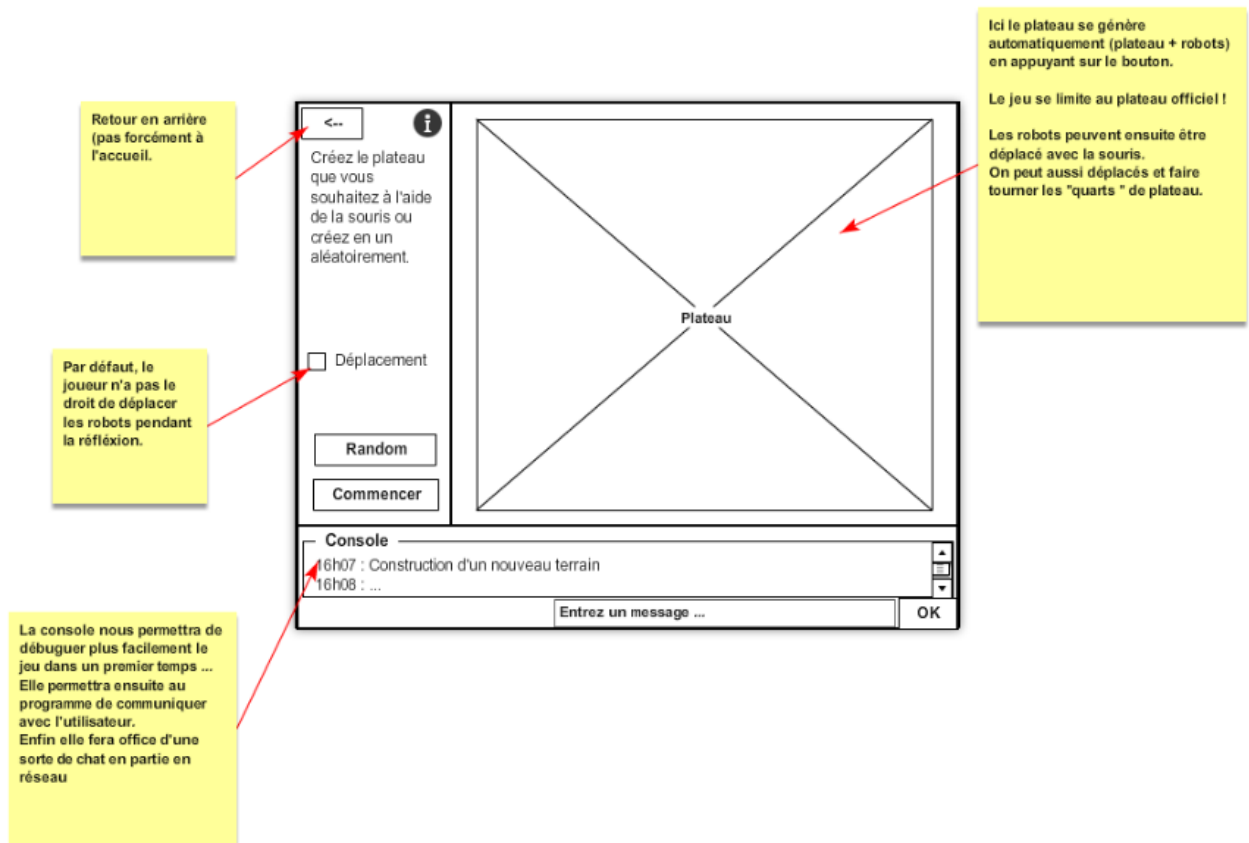
# 1 Scénario

Les besoins que nous avons définis avec les clients sont extraits du scénario qui a été élaboré.

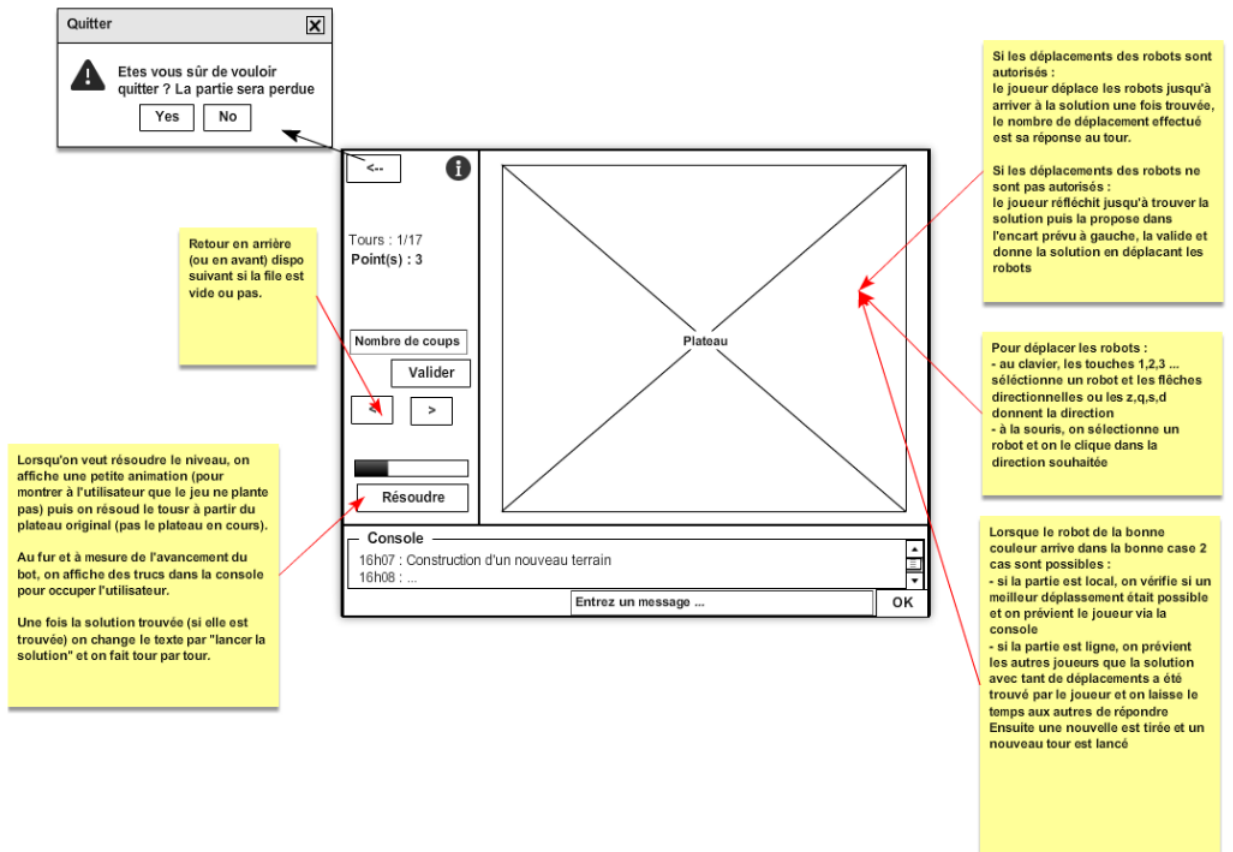
## 1.1 Accueil



## 1.2 Création terrain



## 1.3 Jeu



## **2 Besoins fonctionnels**

### **2.1 Écran d'accueil**

Il s'agit du premier écran qui s'affiche lorsque l'on lance le jeu. Il contient :

- Une image/logo du jeu et le titre du jeu (s'il n'est pas déjà dans l'image)
- Des boutons « Jouer en local », « Jouer en ligne » et « Comment jouer ? »
- Un lien vers les indications de la licence.

### **2.2 Jouer en local**

#### **2.2.1 Création du terrain**

Lors du clic sur ce bouton un nouvel écran s'affiche, il contient :

- Le plateau de jeu
- Un bouton permettant de générer le plateau de manière aléatoire (il sera aussi possible de générer le plateau grâce à la souris)
- Un bouton permettant de régler quelques options (difficulté de l' IA, ...)
- Un bouton permettant de commencer la partie
- Un bouton pour retourner en arrière (page d'accueil ?)
- Une zone console au bas de l'écran permettant de déboguer durant le développement et d'afficher des informations au joueur ensuite

#### **2.2.2 Déroulement de la partie**

Lors du clic sur le bouton commencer l'écran change pour afficher :

- Le plateau généré à l'écran précédent
- Un compteur de tour de jeu
- Un compteur de déplacement
- Un compteur de tour gagné par l'utilisateur
- Un bouton « Résoudre » permettant de trouver le chemin le plus court grâce à l' IA du jeu
- Une petite animation qui apparaît lorsque l'on clique sur le bouton « Résoudre » (elle permet de montrer à l'utilisateur que le jeu n'a pas planté)
- Un bouton « Quitter » permettant de quitter la partie (si l'utilisateur clic dessus un message informera le joueur que la partie sera perdue et

lui demandera de valider son choix)

- Deux boutons permettant de revenir au déplacement précédent ou suivant s'il y a lieu.

Spécificités de cet écran :

- Il n'est plus possible de changer la disposition du plateau de jeu
- Le 1<sup>er</sup> couple de couleur de robot/emplacement à atteindre est défini (un nouveau couple sera redéfini à chaque nouveau tour de jeu)
- Déplacement à la souris ou au clavier pour résoudre la partie
- Si le joueur trouve une solution, on lance l' IA pour rechercher une meilleure solution (si l' IA en trouve une le joueur perd, sinon il gagne ?)

## **2.3 Jouer en ligne**

### **2.3.1 Sélection du serveur de jeu/Création d'un serveur**

Lors du clic sur le bouton « Jouer en ligne » un nouvel écran apparaît, il contient :

- Un champ de texte permettant de rentrer une adresse IP (+ un port ?)
- Un bouton « Rejoindre » qui permet de rejoindre la partie qui se trouve à l'adresse IP définie dans le champ de texte (si elle existe)
- Un bouton « Créer » qui permet de créer une partie

### **2.3.2 Création de la partie**

Après avoir rejoint la partie les joueurs arrivent sur un écran d'attente le temps que la partie démarre. Cet écran sert à attendre que tous les joueurs soient connectés à la partie et soient prêts. Cet écran contient :

- La liste des pseudos de tous les joueurs connectés à la partie (le pseudo sera demandé lorsque l'on rejoindra une partie)
- Le statut du joueur (prêt / non-prêt)
- Un bouton « Prêt »
- Une console faisant office de chat
- Un bouton « Lancer la partie » visible uniquement par l'hébergeur de la partie qui s'activera quand les conditions de lancement de la partie seront remplies

Spécificité de cet écran :

- Une partie pourra être lancée uniquement si elle contient au moins deux joueurs et que tous les joueurs sont prêts
- L'hébergeur de la partie aura la possibilité d'expulser un joueur qui sera inactif pendant trop longtemps

### 2.3.3 Déroulement de la partie

Lorsque l'hébergeur lance la partie, l'écran change pour afficher :

- Le plateau généré par le programme « Serveur »
- Un compteur de tour de jeu
- Un compteur de déplacement
- Un compteur de tour gagné par l'utilisateur
- Un bouton « Quitter » permettant de quitter la partie (si l'utilisateur clic dessus un message informera le joueur que la partie sera perdue et lui demandera de valider son choix)
- Deux boutons permettant de revenir au déplacement précédent (resp. suivant) s'il existe

Spécificités de cet écran :

- Le 1<sup>er</sup> couple de couleur de robot/emplacement à atteindre est défini par le programme « Serveur » (un nouveau couple sera redéfini à chaque nouveau tour de jeu)
- Déplacement à la souris ou au clavier pour résoudre la partie
- Si le joueur trouve une solution, les autres joueurs ont une minute pour proposer une meilleure solution (le joueur ayant la meilleure solution gagne)

## 3 Besoins non-fonctionnels

### 3.1 Jouabilité

Le jeu doit être jouable, il est donc nécessaire d'avoir :

- une interface ergonomique et simple.
- des commandes intuitives, avec par exemple la possibilité d'utiliser les flèches pour le déplacement des robots et l'initiale de la couleur du robot pour en sélectionner un.
- un temps de réponse correct .
- la possibilité de varier la difficulté en option (algorithme de l'IA ou plateau de jeu plus ou moins difficiles)



Il est également nécessaire pour garder un intérêt dans le jeu de tout faire pour éviter les possibilités de triche. Il faut notamment vérifier pendant les parties en réseau qu'il n'y a pas de possibilité de tricher lorsqu'il y a un temps limite.

### **3.2 Systèmes d'exploitation**

Le jeu doit être portable sur Linux et Mac.

### **3.3 Réutilisation**

Le choix d'une licence libre permettra au projet de continuer dans le temps et d'être repris. La mise en place d'une documentation (du type Javadoc) est aussi utile pour permettre au code d'être éventuellement modifié dans le futur.

## **4 Priorités**

1. Partie Solo.
2. Algorithme trouvant la meilleure solution possible.
3. Partie en réseau.
4. En option : génération aléatoire de la carte ou générée à partir de fichiers (création par l'utilisateur).
5. En option : autres algorithmes qui permettraient par exemple de varier la difficulté.

## **5 Planning Gantt**

## Diagramme de Gantt - Projet Rasende Reboter

Tâche/Semaine	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14
<b>PHASE I</b>														
Bibliographie + Existant	TOUS								V					
Analyse des besoins		M+R+L							A					
Gantt		B+D							C					
Cahier des tests			TOUS						N					
Architecture			TOUS	TOUS					C					
									E					
<b>PHASE II</b>									S					
Mise en place de la base du projet					B+R									
Interface + Création d'élém. graphiques					L+D									
Partie locale fonctionne sans IA					B+M+R	B+R								
Travail sur algo de meilleure solution					D+R	D+R	D+R							
Parties en réseau						M+L	D+M+L	D+M+L						
Option : carte aléatoire							R+B	R+B	V					
Tests Beta + rattrapage éventuel retard									A	TOUS	TOUS			
Option : autre IA									C	?	?			
Version Beta									A		29 mars			
Correction bugs									N			TOUS		
Tests Finaux									C			TOUS		
Fin du mémoire									E		TOUS	TOUS	TOUS	
Version Finale									S				9 avril	
Soutenance														16 avril

### Calendrier :

S1	14-janv	20-janv
S2	21-janv	27-janv
S3	28-janv	03-févr
S4	04-févr	10-févr
S5	11-févr	17-févr
S6	18-févr	24-févr
S7	25-févr	03-mars
S8	04-mars	10-mars
S9	11-mars	17-mars
S10	18-mars	24-mars
S11	25-mars	31-mars
S12	01-avr	07-avr

	Début	Fin
S13	08-avr	14-avr
S14	15-avr	21-avr

### Légende :

B = Braik Olivier  
 D = Delesse Alexandre  
 R = Ranc Dimitri  
 L = Lussagnet Gaetan  
 M = Mourany Alexandre  
 TOUS = ensemble de l'équipe  
 ? = option si membres disponibles

## 6 Tableau des risques

N°	Explication	Risque (1 à 3)	Criticité (1 à 3)	Plan d'action
1	Perte de données	1	3	Utiliser un SVN + sauvegardes
2	Problème de jouabilité	1	1	Revoir certaines règles avec les clients
3	Problèmes liées à la partie réseau	2	2	
4	Problème lié aux algorithmes de l'IA	2	2	Utiliser la bibliographie
5	Problème de temps	3	2	Abandon des options
6	Problèmes liés à l'architecture	2	3	Demander avis du prof de TD
7	Mauvaise prise en compte des besoins	2	2	Rencontre régulières avec clients
8	Documentation non créée ou mise à jour	2	1	Utiliser Javadoc
9	Manque d'éléments pour le mémoire	1	1	Reprendre les éléments déjà créés (bibliographie, besoins...) et compléter au fur et à mesure

**Légende :**

1 = faible

2 = moyen

3 = fort