Société e-nov data
Services Informatiques
Développement
13009 Marseille

Société GAME-IN
163 avenue de Luminy case 901
13288 Marseille Cedex 9

QTpilot

Octobre - Décembre 2008

Objet :

Dossier de suivi du projet « QTpilot »

Auteurs :

Auteurs	**Approbateurs**	**Validation**
Zeina BAALBAKI Stephanie MALAKIAN	Stéphane GRANGE Christophe DEVIN	Laurent TICHIT
Livré le : *12/12/08*	Approuvé le : *12/12/08*	*En attente de validation*

Diffusion :

Diffusion	*Externe*
À :	Laurent TICHIT
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Stephanie MALAKIAN, Michael REZAC.

Historique :

Version du document	**Date de modification**	**Modifications apportées**	**Auteurs des modifications**
V1.0	23/10/08	Création du document	Zeina BAALBAKI, Stephanie MALAKIAN
V1.1	29/10/08	Cahier des charges V1.1	Stéphane GRANGE, Christophe DEVIN
V2.0	08/11/08	Maquette, DAT V1.0, Manuel de déploiement V1.0, Manuel d'utilisation V1.0	Équipe e-nov data au complet
V2.1	19/11/08	Mise à jour planning réel, Mise à jour annexes	Christophe DEVIN, Stephanie MALAKIAN, Michael REZAC
V2.2	05/12/08	Mise à jour DAF, DAT, Annexes	*e-nov data*
Vfinale	12/12/08	Mise à jour complète	*e-nov data*

Table des matières

1 Introduction **1**
1.1 Présentation de « QTpilot » 1
1.2 Enjeux du projet 1

2 Présentation des acteurs du projet **2**

I Plan d'Assurance Qualité **3**

3 Objet et caractéristiques du document **4**
3.1 Objectifs du plan 4
3.2 Domaine d'application 4
3.3 Responsabilités de réalisation et de suivi du plan 4
3.4 Documents applicables 4
3.5 Documents de référence 5

4 Organisation et suivi du projet **5**
4.1 Les acteurs du projet 5
4.2 La méthodologie de gestion du projet 5
4.3 Les réunions 5
4.4 Les outils de travail 6

5 Terminologie **6**
5.1 Glossaire des termes 6
5.2 Abréviations 6

6 Objectifs et engagements qualité du projet **7**

7 Démarche de développement du Système d'Information **7**

8 Gestion de la documentation **8**
8.1 Structure des documents 8
8.2 Nomenclature 8

II Cahier des Charges **9**

9 Objectif **10**

10 Description du jeu *QTpilot* **10**
10.1 Présentation 10
10.2 Les paramètres 10
10.3 Les différents modes de jeu 10

10.3.1 Le mode « drapeau » . . . 11
10.3.2 Le mode « death-match » . . . 11
10.3.3 Le mode « grand-prix » . . . 11
10.4 Les options supplémentaires . . . 11

11 Les documents fournis **11**

12 Contrainte technique **12**

III Découpage en lots/livrables **13**

IV Analyse des risques **17**

V Planification **20**

13 Planning prévisionnel **22**
13.1 Planning prévisionnel détaillé . . . 22
13.2 Planning prévisionnel : récapitulatif des dates-clés . . . 24

14 Planning réel **27**
14.1 Planning réel détaillé . . . 27
14.2 Planning réel : récapitulatif des dates-clés . . . 29

15 Bilan de la planification **33**

VI Analyse financière **34**

VII Spécifications de l'interface **37**

16 Page d'accueil **38**

17 Règles du jeu **40**

18 Page des options **41**

19 Page du jeu **42**
19.1 Jeu en réseau . . . 43
19.1.1 Création d'une partie . . . 43
19.1.2 Rejoindre une partie . . . 44

VIII Dossier d'Analyse Fonctionnelle 45

20 Cas d'utilisations 47
20.1 Cas d'utilisation général . . . 47
20.2 Cas d'utilisation « jouer » . . . 48

21 Diagramme de séquence 49

22 Diagramme métier 51

23 Diagramme des classes 53

24 Diagramme de navigation 55

25 Diagramme de déploiement 56

IX Dossier d'Analyse Technique 57

26 Choix matériels 58
26.1 Les outils de travail . . . 58

27 Choix logiciels 59
27.1 Interface graphique du menu . . . 59
27.2 La gestion du réseau . . . 59
27.3 Le graphisme de l'application . . . 59
27.3.1 Création des divers objets graphiques de l'application . . . 59
27.3.2 Affichage des divers objets graphiques . . . 59
27.3.3 Gestion des cartes . . . 59
27.4 Version et composants livrés . . . 59

X Manuel de Maintenance 60

28 Convention de codage 61

29 Partie graphisme 61
29.1 La classe QtP_Map.cpp . . . 61
29.2 La classe QtP_Player.cpp . . . 62
29.3 La classe QtP_Gravity.cpp . . . 62
29.4 La classe QtP_Shoot.cpp . . . 63
29.5 La classe QtP_Wall.cpp . . . 63
29.6 Gestion des maps via un fichier xml / schéma xsd . . . 63

30 Partie réseau **64**
30.1 Le serveur : classe QtP_ Reseau_serv.cpp 64
30.2 Le serveur : classe QtP_ Reseau_client.cpp 65

XI Manuel de déploiement **66**

31 Pré-requis d'utilisation **67**

32 Téléchargement de « QTpilot » **67**

33 Installation de « QTpilot » sur Windows **67**

34 Installation de « QTpilot » sur GNU Linux **68**

XII Manuel d'utilisation de « QTpilot » **69**

35 Faire déplacer son vaisseau dans la carte **70**
35.1 Avancer 70
35.2 Tourner 70
35.3 Reculer et freiner 70
35.4 Augmenter sa vitesse 70

36 Tirer **71**

37 Éviter les zones de plein ou les points de gravité... **71**

XIII Dossier d'Évolution **72**

38 Suppression de la map des vaisseaux des joueurs déconnectés **73**

39 Introduction des points de vie pour chaque joueur **73**

40 Possibilité de tirs arrières et sur les côtés **73**

41 Création d'une mini-map « radar » **74**

42 Introduction des objets **74**

43 Création d'un harpon pour chaque vaisseau **75**

44 Choix des maps **75**

45 Création de deux modes de jeu en réseau supplémentaires **75**

46 Création de cartes par l'utilisateur directement via le menu **76**

47 Choix de la configuration des touches ou utilisation d'une manette **76**

48 Création du mode de jeu « Entraînement »contre l'ordinateur **76**

49 Crash en cas de collision **77**

XIV Bilan final du projet 78

XV Annexes 79

50 Tableau de Bord **79**
50.1 Tableau de bord des courriels émis 80
50.2 Tableau de bord des courriels reçus 82
50.3 Gestion des livraisons 84

51 Tableau de bord des réunions internes **85**

52 Comptes rendus des réunions externes **86**
52.1 Compte rendu de la réunion externe n°1 du 22/10/08 88
52.2 Compte rendu de la réunion externe n°2 du 05/11/08 90
52.3 Compte rendu de la réunion externe n°3 du 12/11/08 92
52.4 Compte rendu de la réunion externe n°4 du 19/11/08 93
52.5 Compte rendu de la réunion externe n°5 du 01/12/08 94
52.6 Compte rendu de la réunion externe n°6 du 08/12/08 95

53 Comptes rendus des réunions internes **96**
53.1 Compte rendu de la réunion interne n°1 du 22/10/08 98
53.2 Compte rendu de la réunion interne n°2 du 30/10/08 99
53.3 Compte rendu de la réunion interne n°3 du 19/11/08 100
53.4 Compte rendu de la réunion interne n°4 du 26/11/08 101

54 Fiches d'itération **102**
54.1 Fiche d'itération n°0 (itération de lancement) 104
54.2 Fiche d'itération n°1 105
54.3 Fiche d'itération n°2 106
54.4 Fiche d'itération n°3 107
54.5 Fiche d'itération n°4 (finale) 109

1 Introduction

La société *GAME-IN* a fait appel à nos services afin de réaliser un jeu vidéo : le jeu « QTpilot ».

1.1 Présentation de « QTpilot »

« QTpilot » est un jeu multi-joueurs. Chaque joueur appartient à une équipe et pilote un vaisseau à travers une carte.
La carte représente des zones de « vide » à travers lesquelles les vaisseaux peuvent évoluer et des zones de « plein » sur lesquelles ils peuvent s'écraser, voire rebondir.
Des objets peuvent intervenir dans une carte : par exemple des drapeaux, des tirs, des missiles, des mines etc...
Par ailleurs, les jeux sont paramétrables : par exemple, l'utilisateur doit pouvoir choisir de paramétrer la densité de l'atmosphère ou encore la gravité.
Trois modes de jeu sont possibles :

- le mode « drapeau », pour lequel le but du jeu est de capturer un objet et de le ramener dans sa « goal-zone »,
- le mode « death-match », pour lequel le but du jeu est d'éliminer un maximum d'adversaires de l'équipe adverse,
- le mode « grand-prix », qui est en fait une course (circuit ou grand 8) avec les autres joueurs (et dans ce cas les tirs doivent être désactivés).

1.2 Enjeux du projet

Le travail des membres de l'équipe de *e-nov data* est donc de mettre en place cette application, du développement logiciel à la réalisation d'une documentation complète en passant par les différentes phases intervenant dans le projet (Spécifications, Analyse, Conception, Implémentation et Tests).

Le produit final se doit d'être ergonomique, clair d'un point de vue de l'utilisation, mais aussi attractif d'un point de vue ludique.

2 Présentation des acteurs du projet

Acteur pour la société cliente « GAME-IN » :

Laurent TICHIT :
Activité : Représentant de la société GAME-IN
✉ : tichit@lidil.univ-mrs.fr

Acteurs pour la société « e-nov data » :

Zeina BAALBAKI :
Activité : Analyste Programmeur chez e-nov data, Responsable qualité du projet
✉ : zeinabaalbaki@gmail.com
✆ : 06.32.79.79.13

Fouzi BENMAKHLOUF :
Activité : Analyste Programmeur chez e-nov data
✉ : fouzi.benmakhlouf@gmail.com
✆ : 06.70.05.51.82

Christophe DEVIN :
Activité : Analyste Programmeur chez e-nov data
✉ : devinchristophe2@gmail.com
✆ : 06.86.92.48.39

Stéphane GRANGE :
Activité : Analyste Programmeur chez e-nov data
✉ : stephane.grange@dil.univ-mrs.fr
✆ : 06.79.35.24.53

Stephanie MALAKIAN :
Activité : Chef de projet, Analyste Programmeur chez e-nov data
✉ : malakian.stephanie@gmail.com
✆ : 06.19.93.78.14

Michael REZAC :
Activité : Analyste Programmeur chez e-nov data
✉ : rezac.michael@tsundere.fr
✆ : 06.84.91.21.65

Première partie

Plan d'Assurance Qualité

Objet : L'objectif du Plan d'Assurance Qualité est de préciser les éléments permettant de s'assurer de la mise en œuvre et de l'efficacité des activités prévues pour ce projet afin d'obtenir la qualité requise. Ce document définit donc les méthodes, l'organisation et les activités d'assurance et de contrôle de la qualité spécifiques au projet *QTpilot*.

Auteurs :

Auteurs	**Approbateurs**	**Validation**
Zeina BAALBAKI Stephanie MALAKIAN	Christophe DEVIN Fouzi BENMAKHLOUF	Laurent TICHIT
Livré le : *26/10/08*	Approuvé le : *25/10/08*	Validé le : *29/10/08*

Diffusion :

Diffusion	*Externe*
À :	Laurent TICHIT
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Stephanie MALAKIAN, Michael REZAC.

Historique :

Version du document	**Date de modification**	**Modifications apportées**	**Auteurs des modifications**
V1.0	23/10/08	Création du document	Zeina BAALBAKI, Stephanie MALAKIAN

3 Objet et caractéristiques du document

Le Plan d'Assurance Qualité est le document de base qui définit l'organisation spécifique technique et fonctionnelle mise en place de la société « e-nov data », afin de garantir la qualité des logiciels qu'elle développe.
Ce document présente l'objectif et les caractéristiques de ce plan, la conduite du projet, l'identification des acteurs et leur niveau d'intervention. Il permet également de clarifer les enjeux et les attentes du client, représentant de la société « GAME-IN », concernant la réalisation du jeu vidéo « QTpilot ».

3.1 Objectifs du plan

Le présent document énonce les pratiques, la méthodologie, les moyens et principes d'organisation qualité à respecter au niveau de notre société. Il permet de décrire l'organisation de l'équipe du projet, de définir le rôle de chacun des intervenants et d'optimiser les méthodes de travail afin de satisfaire aux besoins de notre client, représentant de la société « GAME-IN ».
Notre société s'engage ainsi à prendre en charge et mener à bien la qualité de l'application d'intégration et ses responsabilités d'assurance de contrôle et de suivi du projet « QTpilot ».

3.2 Domaine d'application

Les diverses consignes exprimées à travers ce plan d'assurance qualité s'appliquent depuis la phase de lancement du projet, jusqu'à la phase d'implémentation finale, en passant par les phases de conception, d'analyse et de tests, sans oublier la phase de documentation.

3.3 Responsabilités de réalisation et de suivi du plan

La réalisation de ce plan d'assurance qualité ainsi que son suivi sont du ressort du responsable qualité du projet, qui est en relation directe avec le chef de projet afin de lui soumettre directement les éventuelles actions à entreprendre afin de garantir la bonne exécution du plan.

3.4 Documents applicables

Les documents applicables correspondent à l'ensemble des documents produits au cours du projet, sur lesquels doivent s'appliquer ce plan, il s'agit de :

- ◇ L'analyse financière
- ◇ L'analyse des risques
- ◇ La planification du projet
- ◇ Le Dossier d'Analyse Fonctionnelle
- ◇ Le Dossier d'Analyse Technique
- ◇ Le dossier de Conception
- ◇ Les fiches de tests
- ◇ Le manuel de déploiement de l'application
- ◇ Le manuel d'utilisation de l'application

- ◇ Le dossier de maintenance
- ◇ Le dossier d'évolution
- ◇ Les comptes rendus de réunions (internes et externes)
- ◇ Les fiches d'itération.

3.5 Documents de référence

Les documents de référence sont les documents sur lesquels s'appuient les documents applicables. Il s'agit de :

- ◇ Le Cahier des charges (cf. **partie II**, à la **page 9**)
- ◇ Les documents mis à la disposition de l'équipe sur ce site.
- ◇ Le site de documentation UML en français
- ◇ Le site developpez.com.

4 Organisation et suivi du projet

4.1 Les acteurs du projet

Les responsabilités associées à la réalisation du projet « QTpilot » sont distribuées aux personnes suivantes :

- Le chef de projet
- Le responsable qualité du projet
- L'équipe de développement

L'équipe rattachée au projet « QTpilot » est présentée dans la fiche des acteurs du projet (cf. **section 2** à la **page 2**).

4.2 La méthodologie de gestion du projet

La méthodologie de gestion du projet « QTpilot » est l'XP (*Extreme Programming*), dont les principes fondamentaux sont :

- Le représentant de la société cliente est intégré à plein temps dans l'équipe de développement afin d'atteindre une réactivité optimale
- La programmation se fait en binôme
- L'intégration est continue, dès le début du projet
- Les livraisons sont fréquentes.

La démarche de développement utilisée s'appuie sur un cycle de vie en spirale.

4.3 Les réunions

Réunions internes Afin de garantir une bonne coordination de l'équipe de développement, des réunions internes hebdomadaires sont organisées et regroupent l'ensemble des membres de l'équipe de

travail.
Les réunions internes donnent lieu à la rédaction d'un compte rendu par deux des membres de l'équipe.

Réunions externes Des réunions externes sont faites une fois par semaine, en présence du client ce qui permet d'être en contact direct avec lui, et ainsi d'avoir une réactivité optimale en ce qui concerne ses attentes et ses besoins.
Les réunions externes donnent lieu elles aussi à la rédaction d'un compte rendu par deux des membres de l'équipe, qui est livré au client dans un délai maximum de 48 heures.

4.4 Les outils de travail

Voici la liste des différents outils utilisés tant bien pour l'organisation et le développement que pour le suivi du projet[1] :

- ▷ **Systèmes d'exploitation** : Linux, Windows XP, Windows Vista
- ▷ **Planification** : Planner 0.14
- ▷ **Modélisation UML** : BOUML
- ▷ **Langage de programmation** : C++ / Qt (version 4.x)
- ▷ **Génération de documentation de code** : DOXYGEN
- ▷ **Gestion des versions** : SVN (Subversion)[2]
- ▷ **Mise en page du rapport** : LaTeX

5 Terminologie

5.1 Glossaire des termes

Un glossaire des termes intervenant dans le projet est disponible sur ce site.

5.2 Abréviations

Voici la liste des abréviations utilisées au cours du cycle de vie du projet :

- **UML** : *Unified Modeling Language*
- **XP** : *Extreme Programming*
- **CU** : Cas d'Utilisations
- **PAQ** : Plan d'Assurance Qualité
- **DAF** : Dossier d'Analyse Fonctionnelle
- **DAT** : Dossier d'Analyse Technique
- **IHM** : Interface Homme Machine
- **CR** : Compte Rendu

[1] Une justification des outils utilisés est présentée dans le Dossier d'Analyse Technique, à la section IX, à la page 57.
[2] SVN : un repository (ou dépôt) a été créé via Google afin que tous les membres de l'équipe puisse accéder aux divers documents.

6 Objectifs et engagements qualité du projet

La société « e-nov data » s'engage envers la société « GAME-IN » à mettre en œuvre les moyens nécessaires au développement du jeu « QTpilot » afin de garantir :

- la stabilité et la fiabilité,
- l'ergonomie,
- la simplicité d'utilisation,
- et la maintenabilité de l'application.

7 Démarche de développement du Système d'Information

Cette section regroupe sous forme de tableau les objectifs et résultats des différentes phases intervenant dans le projet.

Phase	Objectifs	Résultats
Lancement	- Définition des enjeux du projet - Définir le comité de pilotage - Établir une planification - Analyser les risques - Gérer le budget	- Fiche des acteurs - Planning prévisionnel - Analyse des risques - Analyse financière - PAQ
Analyse des besoins du client	- Étude informelle des fonctionnalités du système à réaliser - Dialogue avec le client pour cerner ses attentes	- Première version du cahier des charges
Spécifications	- Identifier ce que doit faire le système à réaliser (fonctionnalités)	- Cahier des charges complété - Début de rédaction du DAF - Diagrammes des cas d'utilisation
Analyse du Système	- Comprendre et modéliser le système hors de toute considération technique - Identifier les éléments intervenant (hors acteurs) dans le système : fonctionnalités, structures, relations	- DAF complété - Diagrammes de séquence, de navigation, de classes

Tab. 1 – Démarche de développement du Système d'Information.

Phase	Objectifs	Résultats
Conception et Implémentation	- Déterminer comment faire le système en établissant des choix techniques - Préparer à la définition des phases suivantes	- Programmes exécutables - Rédaction du DAT - Début de rédaction du dossier de maintenance
Tests unitaires	- Tester les composants logiciels de chaque itération - Valider ou non	- Fiches de tests unitaires - Dossier de maintenance complété
Intégration	- Intégrer chaque composant logiciel additionnel à l'application générale	- Code source complété
Validation	- Valider le bon fonctionnement de l'application	- Manuel de déploiement - Manuel d'utilisation - Dossier d'évolution complet

TAB. 2 – Démarche de développement du Système d'Information (suite).

8 Gestion de la documentation

Cette section vise à décrire la manière dont sont gérés les divers documents réalisés au cours du projet.

8.1 Structure des documents

Chaque document diffusé possède une structure générique qui précise :
- Les auteurs, les approbateurs (qui doivent être différents des auteurs) ainsi que le ou les validateur(s) du document
- Le type de diffusion (externe ou interne) ainsi que les personnes à qui le document est diffusé
- L'historique des versions du document ainsi que les raisons des modifications et leurs auteurs.

8.2 Nomenclature

Tout document est nommé de la façon suivante :

$$[nom\_du\_document]\text{-v}[numero\_de\_version]$$

Le numéro de version est de la forme : x.y, où x est un chiffre compris entre 1 et 9, et y entre 0 et 9. Lors d'une modification, le chiffre y est incrémenté de 1. Si la modification apportée est considérée comme importante, alors x est incrémenté de 1 et y passe à 0.

Deuxième partie

Cahier des Charges

Objet : L'objectif de ce document est de présenter les besoins et attentes du client concernant le produit.

Auteurs :

Auteurs	**Approbateurs**	**Validation**
Zeina BAALBAKI Stephanie MALAKIAN	Christophe DEVIN Michael REZAC	Laurent TICHIT
Livré le : *26/10/08*	Approuvé le : *25/10/08*	Validé le : *29/10/08*

Diffusion :

Diffusion	*Externe*
À :	Laurent TICHIT
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Stephanie MALAKIAN, Michael REZAC.

Historique :

Version du document	**Date de modification**	**Modifications apportées**	**Auteurs des modifications**
V1.0	23/10/08	Création du document	Zeina BAALBAKI, Stephanie MALAKIAN

9 Objectif

L'objectif de ce cahier des charges est de présenter de manière précise et informelle les fonctionnalités attendues par le maître d'ouvrage (qui est le représentant de la société cliente GAME-IN) concernant le produit *QTpilot*.
En aucun cas nous ne considérons ici l'aspect technique de l'application.

10 Description du jeu *QTpilot*

10.1 Présentation

L'application à développer est un jeu vidéo multi-joueurs : chaque joueur appartient à une équipe et pilote un vaisseau à travers une carte.
QTpilot est un jeu en réseau, c'est-à-dire que chaque joueur joue sur son propre ordinateur, les ordinateurs de chaque joueur étant connectés entre eux via un réseau. Ainsi, chaque joueur joue avec les autres utilisateurs connectés, en temps réel et de manière simultanée, sur la même carte.

Un vaisseau possède un réacteur, un canon et un harpon électro-magnétique.
Un vaisseau peut accélérer, freiner, tirer, lancer le harpon pour capturer un objet.
La carte représente des zones de « vide » à travers lesquelles les vaisseaux peuvent évoluer, et des zones de « plein » sur lesquelles ils peuvent s'écraser.
Dans une carte peuvent se trouver plusieurs objets (tirs multiples, tirs plus rapides, tirs arrières ou par côté, missiles à tête chercheuse, réservoirs de carburant, « after-burners »[3], « cloaking devices »[4], boucliers anti-chocs, anti-canon, drapeau ennemi, goal-zone, canons indépendants, mines activées ou non, etc...).
Chaque objet a une masse.
Dans les zones de « plein » se trouvent parfois des champs de carburant pour se ravitailler.

10.2 Les paramètres

Les jeux sont paramétrables : par exemple, le joueur peut choisir de paramétrer la densité de l'atmosphère (plus l'atmosphère est dense, plus les frottements de l'air sont importants vis-à-vis du vaisseau), l'intensité de la gravité (gravité centrale, qui correspond à l'attraction vers un point de la carte, ou axiale, qui correspond à l'attraction vers un des bords de la carte).
De plus, le joueur a le choix entre crash ou rebond en cas de collision, en fonction de la vitesse.

10.3 Les différents modes de jeu

QTpilot peut se jouer selon trois modes de jeu :

[3] « after-burners » : gain de puissance pendant un petit laps de temps.
[4] « cloaking devices » : permet de rendre invisible

10.3.1 Le mode « drapeau »

Le mode ***drapeau*** consiste à récupérer le drapeau présent au centre de la carte, qu'un joueur peut capturer à l'aide du harpon dont est équipé le vaisseau. Le joueur doit ensuite ramener ce drapeau dans sa goal-zone, sachant que ce drapeau a une certaine masse (et va donc nécessiter plus ou moins de puissance de la part du vaisseau).

10.3.2 Le mode « death-match »

Le mode ***death-match*** correspond à l'affrontement de deux équipes, chacune essayant de tuer un maximum d'adversaires de l'autre équipe.

10.3.3 Le mode « grand-prix »

Le mode ***grand-prix*** est une course (circuit ou grand 8) avec les autres joueurs. Pour ce mode, les tirs ne sont pas activés.

10.4 Les options supplémentaires

À cela viennent s'ajouter d'autres options supplémentaires, telles que l'option ***radar*** qui permet à un joueur de voir dans une petite fenêtre la position en temps réel des autres joueurs (représentées par des points rouge) par rapport à sa position (représentée par un point vert) dans la carte, ou encore l'option ***entraînement*** pour chaque mode de jeu (qui permet en fait de jouer seul pour s'entraîner).

11 Les documents fournis

Tout au long du cycle de développement du jeu *QTpilot*, les documents suivants seront fournis :
- ◇ Le Plan d'Assurance Qualité
- ◇ L'analyse financière
- ◇ L'analyse des risques
- ◇ La planification du projet
- ◇ Le Dossier d'Analyse Fonctionnelle
- ◇ Le Dossier d'Analyse Technique
- ◇ Le dossier de Conception
- ◇ Les fiches de tests
- ◇ Le manuel de déploiement de l'application
- ◇ Le manuel d'utilisation de l'application
- ◇ Le dossier de maintenance
- ◇ Le dossier d'évolution
- ◇ Les annexes

12 Contrainte technique

Le langage de programmation utilisé pour coder l'application est le C++, et la librairie utilisée pour la gestion des fenêtres est Qt.
Le langage ainsi que la version de Qt (4.4.x) ont été imposés par le client.

Troisième partie

Découpage en lots/livrables

Objet : L'objectif de ce document est de détailler le contenu des différents lots et livrables qui constituent l'application.

Auteurs :

Auteurs	**Approbateurs**	**Validation**
Christophe DEVIN Stéphane GRANGE	Stephanie MALAKIAN Michael REZAC	Laurent TICHIT
Livré le : *26/10/08*	Approuvé le : *25/10/08*	Validé le : *29/10/08*

Diffusion :

Diffusion	*Externe*
À :	Laurent TICHIT
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Stephanie MALAKIAN, Michael REZAC.

Historique :

Version du document	**Date de modification**	**Modifications apportées**	**Auteurs des modifications**
V1.0	23/10/08	Création du document	Christophe DEVIN, Stéphane GRANGE

Voici un tableau détaillant l'application qui doit être développée, découpée en différents lots et livrables :

N° Lot	Nom du lot	Description du lot	Date de livraison	N° de livraison
1	qtpilot_v1.0.tar.gz	**Version initiale du jeu avec les bases :** – Un vaisseau se déplace sur une carte (la plus simple possible : pas de zone de plein pour le moment). – Mise en place du réseau (pour 2 joueurs)	07/11/08	1
2	qtpilot_v1.0.pdf	**Documentation :** – Phase de lancement complète – Cahier des charges partiel – DAT partiel – Manuel de déploiement – Manuel d'utilisation initial – Fiches d'itération n°0 et n°1 – CR réunions externes n°1 et 2	07/11/08	1
3	qtpilot_v2.0.tar.gz	**Nouvelle version du jeu :** – Ajout des zones de plein sur la carte – Gestion des collisions contre ces zones de plein – Ajout des tirs	14/11/08	2
4	qtpilot_v2.0.pdf	**Documentation :** – Cahier des charges complet – DAF partiel – DAT complété – Manuel d'utilisation mis à jour – Fiche d'itération n°2 – CR réunion externe n°3	14/11/08	2

N° Lot	Nom du lot	Description du lot	Date de livraison	N° de livraison
5	qtpilot_v3.0.tar.gz	**Ajout de caractéristiques complexes (paramètres) :** – Gestion de la vitesse / accélération – Prise en compte de la masse – Gestion de la gravité (centrale et axiale) – Prise en compte de la densité de l'atmosphère	24/11/08	3
6	qtpilot_v3.0.pdf	**Documentation :** – DAF complété – Manuel d'utilisation mis à jour – Fiche d'itération n°3 CR réunion externe n°4	24/11/08	3
7	qtpilot_v4.0.tar.gz	**Ajout d'options diverses :** – Introduction des objets (harpon, bouclier, bonus) – Prise en compte du temps (chronomètre) – Gestion des scores – Gestion des bonus (carburant, after-burners, cloaking devices)	28/11/08	4
8	qtpilot_v4.0.pdf	**Documentation :** – Manuel d'utilisation mis à jour – Fiche d'itération n°4 – CR réunion externe n°5	28/11/08	4
9	qtpilot_v5.0.tar.gz	**Possibilité du choix du mode de jeu :** – Création des 3 modes de jeu – Création de l'option « entraînement » pour le mode grand prix seulement – Ajout des menus (choix du mode, choix des paramètres, lancement d'une partie)	05/12/08	5
10	qtpilot_v5.0.pdf	**Documentation :** – Manuel d'utilisation mis à jour – Fiche d'itération n°5 – CR réunion externe n°6	05/12/08	5

N° Lot	Nom du lot	Description du lot	Date de livraison	N° de livraison
11	qtpilot_vfinale.tar.gz	**Amélioration du graphisme et ajout de diverses options :** – Option « radar » – Plus de 2 joueurs en réseau – Défilement de l'écran – Option des rebonds sur les zones de plein – Option « entraînement » pour les modes *drapeau* et *death-match* (joueur contre machine : intelligence artificielle)	12/12/08	6
12	qtpilot_vfinale.pdf	**Documentation :** – Manuel d'utilisation mis à jour – Manuel d'évolution – Fiche de Test – Manuel de maintenance – Fiches des acteurs/activités – Fiche d'itération n°6 – CR réunion externe n°7	12/12/08	6

TAB. 3 – Découpage en lots/livrables du produit.

Quatrième partie

Analyse des risques

Objet : L'objectif de ce document est de lister les risques susceptibles d'intervenir dans le projet, de déterminer leur facteur, leur probabilité d'apparition, leur gravité mais aussi les solutions proposées pour y palier.

Auteurs :

Auteurs	**Approbateurs**	**Validation**
Zeina BAALBAKI Christophe DEVIN	Stéphane GRANGE Stephanie MALAKIAN	Laurent TICHIT
Livré le : *26/10/08*	Approuvé le : *23/10/08*	Validé le : *29/10/08*

Diffusion :

Diffusion	*Externe*
À :	Laurent TICHIT
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Stephanie MALAKIAN, Michael REZAC.

Historique :

Version du document	**Date de modification**	**Modifications apportées**	**Auteurs des modifications**
V1.0	23/10/08	Création du document	Zeina BAALBAKI, Christophe DEVIN

Voici un tableau récapitulatif concernant la gestion des risques du projet (leur facteur, solutions pour y palier, gravité et probabilité[5]) :

Risque	Facteur de risque	Solutions proposées	Gravité	Probabilité
Retard de livraison	Absence d'un membre (maladie)	Répartition de la tâche affectée à la personne absente sur le reste de l'équipe	Faible	Faible
Démission d'un membre de l'équipe	Manque de motivation, changement de sa formation, raison personnelle	La personne ayant l'intention de quitter s'engage à prévenir le groupe 2 semaines avant son départ	Haute	Faible
Conflit entre les membres du groupe	– Incompatibilité dans la manière de travailler du binôme – Incompatibilité de personnalité	Séparer les membres en conflit	Moyenne	Faible
Problème de portabilité	Incompatibilité des logiciels utilisés en lieu de travail et chez soi	– Trouver des logiciels compatibles avec les différents systèmes d'exploitation – Installer le même environnement utilisé au travail chez soi	Faible	Élevée
Manque de connais-sances	Nouvelle méthode ou nouveau langage ou logiciel à utiliser	– Donner la tâche au plus compétent qui va ensuite aider les autres personnes à se former – Formation et apprentissage individuels	Haute	Très Élevée
Mauvaise coordination du temps et mauvaise réalisation du planning prévisionnel	– Chaque membre a ses propres heures de travail (affaires personnelles, activités extra professionnelles) et capacités différentes de celle des autres – Mauvaise évaluation des durées des tâches	Réunion de toute l'équipe pour voir les disponibilités, et les capacités de chacun	Haute	Moyenne

[5] **Faible** : 0-25%, **Moyenne** : 25-50%, **Élevée** : 50-75%, **Très Élevée** : 75-100%

Risque	Facteur de risque	Solutions proposées	Gravité	Probabilité
Temps insuffisant pour la réalisation du projet	Cours, autres projets, retard accumulé	Travail pendant les vacances, weekend, et tard la nuit	Haute	Très élevée
Attente de réponse de la part du client à propos d'un problème	Indisponibilité du client	– Travail sur d'autres tâches n'ayant pas de relation avec le problème rencontré – En cas de retard excessif et l'impossibilité de joindre le client avec les différents moyens possibles, l'équipe doit prendre une décision qui semble la mieux adaptée aux besoins du client	Moyenne	Faible
Perte des données	– Problème du système d'exploitation – Panne du disque dur – Serveur en panne	Toujours conserver un backup sur différents supports (serveur ftp, disque dur secondaire, clé usb, CD)	Haute	Faible

TAB. 4 – Analyse des risques pour le projet « QTpilot ».

Cinquième partie

Planification

Objet : L'objectif de cette partie est de présenter en détail les planifications prévisionnelle et réelle du projet « QTpilot ».

Auteurs :

Auteurs	**Approbateurs**	**Validation**
Stephanie MALAKIAN Zeina BAALBAKI	Christophe DEVIN Stéphane GRANGE	Laurent TICHIT
Livré le : *12/12/08*	Approuvé le : *12/12/08*	*En attente de validation*

Diffusion :

Diffusion	*Externe*
À :	Laurent TICHIT
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Stephanie MALAKIAN, Michael REZAC.

Historique :

Version du document	Date de modification	Modifications apportées	Auteurs des modifications
V1.0	21/10/08	Création du document	Stephanie MALAKIAN, Zeina BAALBAKI
V1.1	24/10/08	Ajout du planning prévisionnel détaillé et général	Stephanie MALAKIAN, Zeina BAALBAKI
V1.2	31/10/08	Ajout de la première et deuxième semaine des dates-clés du planning réel	Stéphane GRANGE, Michael REZAC
V1.3	08/11/08	Ajout de la troisième semaine des dates-clés du planning réel	Zeina BAALBAKI, Christophe DEVIN
V1.4	19/11/08	Ajout de la quatrième semaine des dates-clés du planning réel	Christophe DEVIN, Michael REZAC
V1.5	05/12/08	Ajout de la cinquième semaine des dates-clés du planning réel	Fouzi BENMAKHLOUF, Christophe DEVIN
V2.0	11/12/08	Ajout de la dernière semaine des dates-clés du planning réel	Zeina BAALBAKI, Fouzi BENMAKHLOUF
V2.1	12/12/08	Ajout du planning réel détaillé et rédaction du bilan	Stéphane GRANGE, Stephanie MALAKIAN

13 Planning prévisionnel

13.1 Planning prévisionnel détaillé

Voici le planning prévisionnel détaillé du projet *QTpilot*, réparti du mercredi 22 octobre 2008 au vendredi 12 décembre 2008, ainsi que le diagramme de Gantt correspondant :

QTpilot

Start: October 22, 2008
Finish: December 12, 2008
Phase: Lancement
Report Date: October 24, 2008

Gantt Chart

WBS	Name	Work
1	**Phase de lancement**	**10d**
1.1	Réunion externe n°1 + CR	1d
1.2	Réunion interne n°1 + CR	1d
1.3	Planning prévisionnel	2d
1.4	Analyse des risques	1d
1.5	Analyse financière	1d
1.6	Plan Assurance Qualité	3d
1.7	Formation C++/Qt en réseau	1d
2	**Analyse des besoins**	**6d**
2.1	Description des besoins	2d
2.2	Définition du cahier des charges	4d
3	**Analyse fonctionnelle et Conception**	**12d**
3.1	Maquette	2d
3.2	Diagramme des cas d'utilisations	3d
3.3	Diagramme de navigation	1d
3.4	Diagramme de séquence	3d
3.5	Diagramme de classes	3d
4	Formation C++/Qt graphisme	3d
5	Formation C++/Qt en réseau	3d
6	Réunion interne n°2 + CR	1d
7	**Implémentation version 1.0**	**12d**
7.1	Implémentation de la carte (map)	3d
7.2	Implémentation vaisseau	3d
7.3	Implémentation réseau (client/serveur)	3d
7.4	Fiche de test n°1	1d

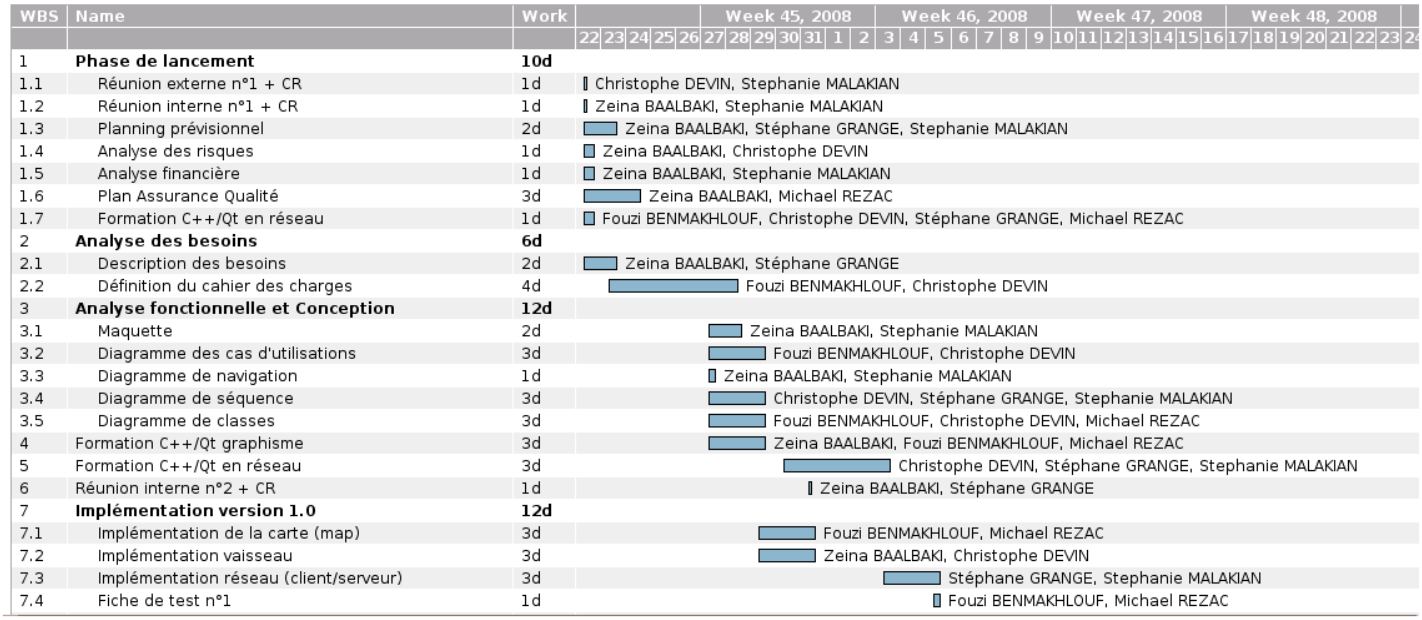


FIG. 1 – Planning prévisionnel : du 22/10/08 au 05/11/08.

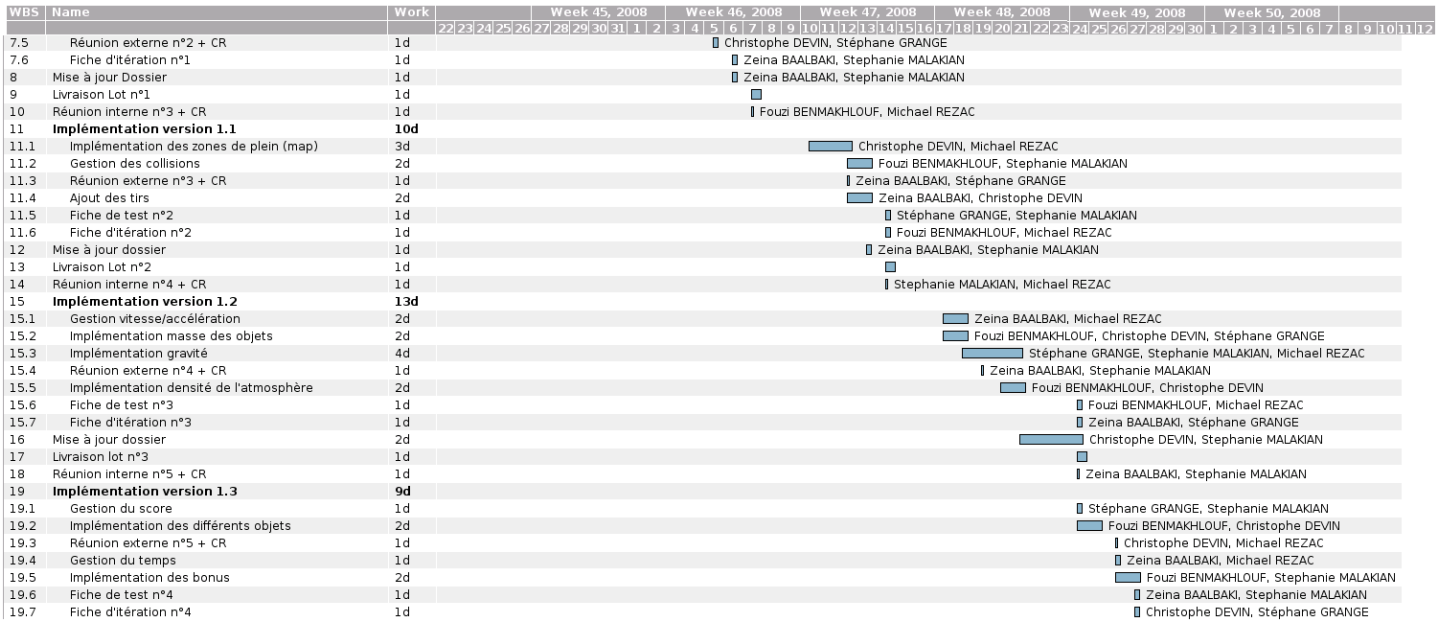


FIG. 2 – Planning prévisionnel : du 05/11/08 au 28/11/08.

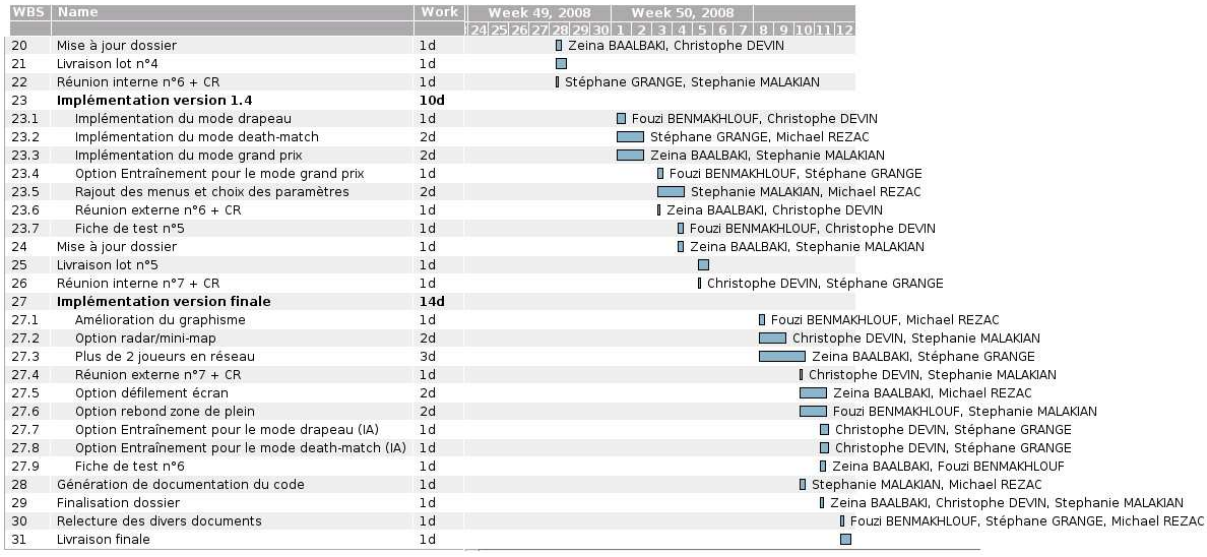


FIG. 3 – Planning prévisionnel : du 28/11/08 au 12/12/08.

13.2 Planning prévisionnel : récapitulatif des dates-clés

Voici l'organisation hebdomadaire générale prévisionnelle donnant un aperçu global de ces 7 semaines de projet, en mettant en évidence les dates-clés :

	lun 20/10	**mar 21/10**	**mer 22/10**	**jeu 23/10**	**ven 24/10**
8-10h			Cours GL	TP XML	Anglais
10-12h	Cours Réseaux	Cours Complexité	Début Projet, **Réunion externe n°1**	TP XML	Phase de lancement
14-16h	TP Réseaux	TD Complexité	Réunion interne n°1	TP XML	Analyse des besoins
16-18h	TD Réseaux	TP Complexité		TP XML	

	lun 27/10	**mar 28/10**	**mer 29/10**	**jeu 30/10**	**ven 31/10**
8-10h	*Début itération n°1*				
10-12h	Début Analyse fonctionnelle	Projet Complexité	Devoir XML	Projet Réseaux	Exposés Anglais
14-16h	Début Conception	Projet Complexité	Devoir XML	Projet Réseaux	Réunion interne n°2
16-18h	Formation C++/Qt (graphisme + réseau)		Début implémentation v1.0		

	lun 03/11	**mar 04/11**	**mer 05/11**	**jeu 06/11**	**ven 07/11**
8-10h			Cours GL		Anglais
10-12h	Cours Réseaux	Cours Complexité	**Réunion externe n°2**	Cours Prolog	Projet XML à rendre
14-16h	Soutenances Réseaux	TD Complexité	Suite Implémentation v1.0	TD Prolog	Livraison n°1, *Fin itération n°1*, Réunion interne n°3
16-18h	TD Réseaux	TP Complexité	Phase de tests	TP Prolog	

	lun 10/11	mar 11/11	mer 12/11	jeu 13/11	ven 14/11
8-10h	*Début itération n°2*				Anglais Exposés
10-12h	Cours Réseaux	Début Implémentation v1.1	**Réunion externe n°3**	Cours Prolog	Livraison n°2
14-16h	TP Réseaux	Suite Implémentation v1.1	Phase de tests	TD Prolog	Fin itération n°2, *Fin itération n°2*, Réunion interne n°4
16-18h	TD Réseaux			TP Prolog	

	lun 17/11	mar 18/11	mer 19/11	jeu 20/11	ven 21/11
8-10h	*Début itération n°3*, Début implémentation v1.2				Anglais
10-12h	Cours Réseaux	Cours Complexité	**Réunion externe n°4**	Cours Prolog	Suite implémentation v1.2
14-16h	TP Réseaux	TD Complexité	Suite implémentation v1.2	TD Prolog	Phase de tests
16-18h	TD Réseaux	TP Complexité		TP Prolog	

	lun 24/11	mar 25/11	mer 26/11	jeu 27/11	ven 28/11
8-10h	Livraison n°3, *Fin itération n°3*, Réunion interne n°5	*Début itération n°4*, Début implémentation v1.3			Anglais
10-12h	Cours Réseaux	Cours Complexité	**Réunion externe n°5**	Cours Prolog	Phase de tests
14-16h	TP Réseaux	TD Complexité	Suite implémentation v1.3	TD Prolog	Livraison n°4, *Fin itération n°4*, Réunion interne n°6
16-18h	TD Réseaux	TP Complexité		TP Prolog	Préparation Anglais

	lun 01/12	mar 02/12	mer 03/12	jeu 04/12	ven 05/12
8-10h	*Début itération n°5*, Début implémentation v1.4	Préparation Anglais		Préparation Anglais	Anglais GroupWorks
10-12h	Cours Réseaux	Cours Complexité	**Réunion externe n°6**	Cours Prolog	Phase de tests
14-16h	TP Réseaux	TD Complexité	Suite implémentation v1.4	TD Prolog	Livraison n°5,, *Fin itération n°5*, Réunion interne n°7
16-18h	TD Réseaux	TP Complexité		TP Prolog	

	lun 08/12	mar 09/12	mer 10/12	jeu 11/12	ven 12/12
8-10h	*Début itération n°6*				Anglais
10-12h	Début implémentation vfinale	Cours Complexité	**Réunion externe n°7**	Cours Prolog	Génération documentation, Finalisation rapport
14-16h	Implémentation vfinale	TD Complexité	Implémentation vfinale	TD Prolog	*Fin itération n°6*, **Livraison finale n°6**
16-18h		TP Complexité	Phase de tests	TP Prolog	

14 Planning réel

14.1 Planning réel détaillé

Voici le planning réel détaillé du projet *QTpilot*, réparti du mercredi 22 octobre 2008 au vendredi 12 décembre 2008, ainsi que le diagramme de Gantt correspondant :

QTpilot

Start: October 22, 2008
Finish: December 12, 2008
Phase: Lancement
Report Date: December 6, 2008

Gantt Chart

WBS	Name	Work	Week 45, 2008					Week 46, 2008							Week 47, 2008							Week 48, 2008						
			22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	**Phase de lancement**	**10d**																																	
1.1	Réunion externe n°1 + CR	1d	Stephanie MALAKIAN, Zeina BAALBAKI																																
1.2	Réunion interne n°1 + CR	1d	Stephanie MALAKIAN, Zeina BAALBAKI																																
1.3	Planning prévisionnel	2d				Stephanie MALAKIAN, Zeina BAALBAKI																														
1.4	Analyse des risques	1d	Christophe DEVIN, Zeina BAALBAKI																																
1.5	Analyse financière	1d	Stephanie MALAKIAN, Zeina BAALBAKI																																
1.6	Plan Assurance Qualité	3d		Stephanie MALAKIAN, Zeina BAALBAKI																															
1.7	Formation C++/Qt en réseau	1d																																	
2	**Analyse des besoins**	**4d**																																	
2.1	Description des besoins	2d	Stéphane GRANGE, Fouzi BENMAKHLOUF																																
2.2	Définition du cahier des charges	2d		Stephanie MALAKIAN, Zeina BAALBAKI																															
3	Formation C++/Qt graphisme	3d																																	
4	Formation C++/Qt en réseau	3d																																	
5	Réunion interne n°2 + CR	1d													Stephanie MALAKIAN, Christophe DEVIN																				
6	**Itération de lancement**	**3d**																																	
6.1	Fiche d'itération n°0	1d				Stephanie MALAKIAN, Zeina BAALBAKI																															
6.2	Trame Générale du Dossier	1d				Stephanie MALAKIAN, Zeina BAALBAKI																															
6.3	Livraison Phase de Lancement	1d																																	
7	**Analyse fonctionnelle et Conception**	**12d**																																	
7.1	Maquette	2d									Stéphane GRANGE, Christophe DEVIN																										
7.2	Diagramme des cas d'utilisations V1	3d									Stephanie MALAKIAN, Christophe DEVIN, Zeina BAALBAKI																										
7.3	Diagramme de navigation V1	1d									Stephanie MALAKIAN, Zeina BAALBAKI																										
7.4	Diagramme de séquence V1	3d									Michael REZAC, Stéphane GRANGE, Fouzi BENMAKHLOUF																										
7.5	Diagramme de classes V1	3d									Michael REZAC, Stephanie MALAKIAN, Zeina BAALBAKI																										

FIG. 4 – Planning réel : du 22/10/08 au 29/10/08.

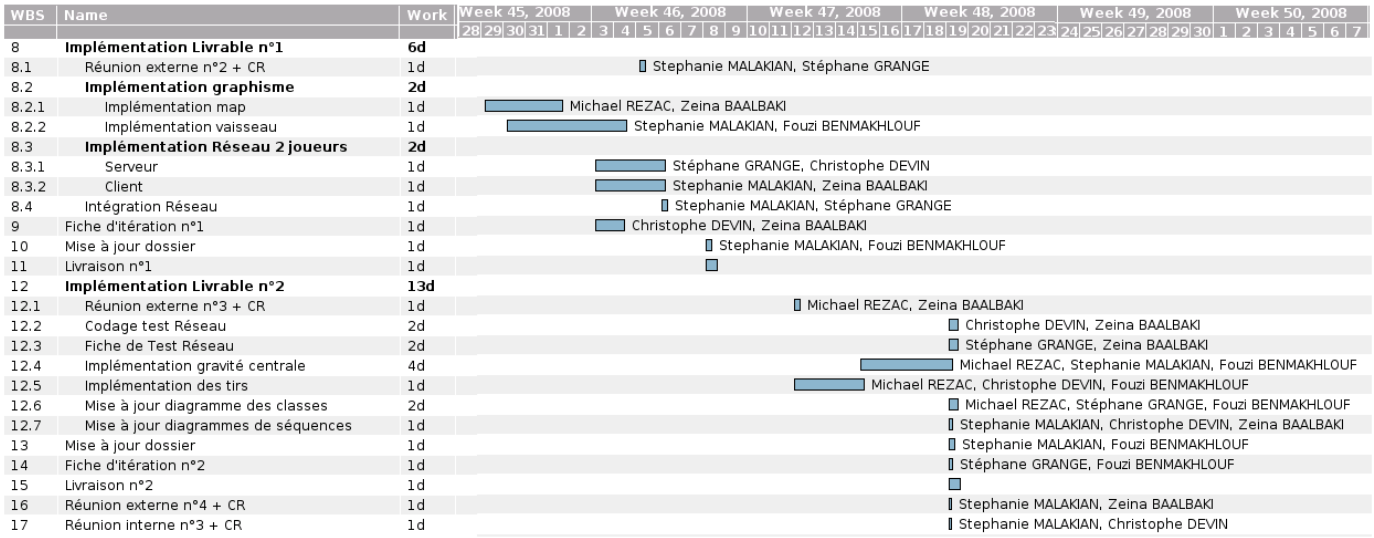


FIG. 5 – Planning réel : du 29/10/08 au 19/11/08.

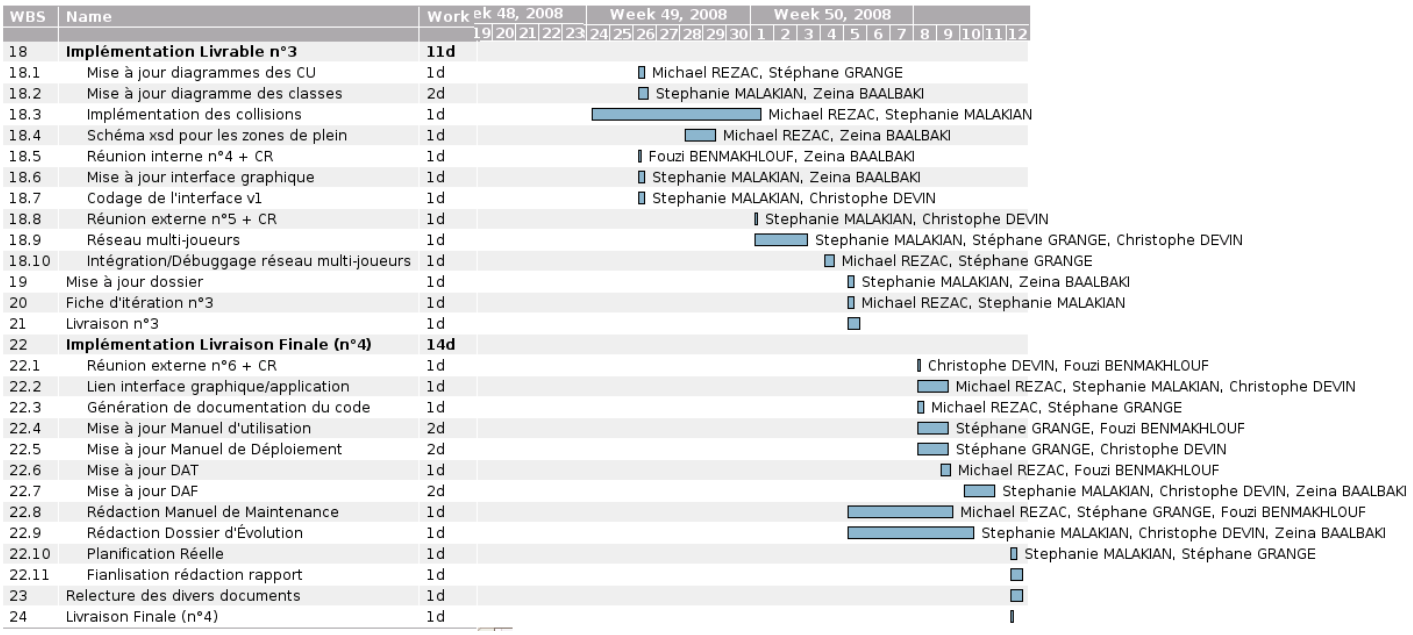


FIG. 6 – Planning réel : du 19/11/08 au 12/12/08.

14.2 Planning réel : récapitulatif des dates-clés

Voici l'organisation générale réelle, semaine par semaine, du projet :

	lun 20/10	mar 21/10	mer 22/10	jeu 23/10	ven 24/10
8-10h			Cours GL Début Projet	TP XML	Anglais
10-12h	Cours Réseaux	Cours Complexité	*Début Itération n°0* **Réunion externe n°1**	TP XML	Phase de lancement : découpage en lots + planning prévisionnel
14-16h	TP Réseaux	TD Complexité	Réunion interne n°1	TP XML	Analyse des besoins, Trame du rapport
16-18h	TD Réseaux	TP Complexité	Formation C++/Qt (réseau et graphisme)	TP XML	Formation C++/Qt (réseau et graphisme), *Fin Itération n°0*

	lun 27/10	mar 28/10	mer 29/10	jeu 30/10	ven 31/10
8-10h	**Livraison phase de lancement** *Début Itération n°1* Diagrammes des CU	Projet Complexité		Devoir XML	Formation C++/Qt (réseau)
10-12h	Détails des cas d'utilisations	Projet Complexité	Début Maquette	Devoir XML	Implémentation V1.0
14-16h	Formation C++/QT (réseau)	Début implémentation V1.0	Maquette, Diagramme de navigation	Devoir XML, Réunion interne n°2, Mise à jour rapport	Projet Réseaux
16-18h	Exposé Anglais		Formation C++/Qt (réseau et graphisme)		Projet Réseaux

	lun 03/11	mar 04/11	mer 05/11	jeu 06/11	ven 07/11	sam 08/11
8-10h	Projet XML	Implémentation réseau	Cours GL	Intégration réseau à l'application graphique	Anglais	
10-12h	Cours Réseaux	Cours Complexité	**Réunion externe n°2**	Cours Prolog	**Démonstration avec le client**	Mise à jour rapport
14-16h	Soutenances Réseaux	TD Complexité	Suite Implémentation (réseau)	TD Prolog	Projet XML	**Livraison n°1**
16-18h	TD Réseaux	TP Complexité Projet	Projet XML	TP Prolog	Projet XML	*Fin Itération n°1*

	lun 10/11	mar 11/11	mer 12/11	jeu 13/11	ven 14/11
8-10h	*Début Itération n°2*	Préparation Anglais			Anglais Exposés
10-12h	Cours Réseaux	Préparation Anglais	**Réunion externe n°3**	Cours Prolog	Devoir Complexité
14-16h	Réseaux Projet Freenet	Devoir Complexité	Début codage tests, Début codage des tirs	TD Prolog	Devoir Complexité
16-18h	TD Réseaux	Devoir Complexité	Diagrammes des CU	TP Prolog	Devoir Complexité

	lun 17/11	mar 18/11	mer 19/11	jeu 20/11	ven 21/11
8-10h	Devoir Complexité		Codage test Client, Fiche de test n°1, Codage gravité centrale		Anglais Exposés
10-12h	Cours Réseaux	Cours Complexité	Diagramme des classes v1.0	Cours Prolog	
14-16h	Réseaux Projet Freenet	TD Complexité	**Réunion externe n°4** Mise à jour rapport	TD Prolog	Vidéo Anglais
16-18h	TD Réseaux	TP Complexité	*Fin Itération n°2* **Livraison n°2** Réunion interne n°3	TP Prolog	Devoir 2a de Complexité

	lun 24/11	mar 25/11	mer 26/11	jeu 27/11	ven 28/11
8-10h	*Début itération n°3*	Devoir 2a de Complexité	Avancement analyse fonctionnelle	Implémentation gestion des collisions	Anglais
10-12h	Cours Réseaux	Cours Complexité	Implémentation gestion des collisions	Cours Prolog	Devoir 2b de Complexité
14-16h	Réseaux Projet Freenet	TD Complexité	Réunion interne n°4	TD Prolog	Vidéo Anglais
16-18h	Projet Freenet	TP Complexité	Codage de l'interface	TP Prolog	Vidéo Anglais

	lun 01/12	mar 02/12	mer 03/12	jeu 04/12	ven 05/12
8-10h	Montage Vidéo Anglais	Montage Vidéo Anglais			Anglais
10-12h	Cours Réseaux	Cours Complexité	Mise à jour interface graphique	Cours Prolog	Devoir 2b de Complexité
14-16h	Réseaux Projet Freenet	TD Complexité	Mise à jour réseau multi-joueurs	TD Prolog	Devoir 2b de Complexité
16-18h	**Réunion externe n°5**	TP Complexité	Mise à jour analyse fonctionnelle	TP Prolog	*Fin itération n°3* **Livraison n°3**

	lun 08/12	mar 09/12	mer 10/12	jeu 11/12	ven 12/12
8-10h	*Début itération n°4*		Mise à jour DAF, DAT		Anglais
10-12h	Manuels d'Évolutions, de Maintenance, d'utilisation	Cours Complexité	Débuggage intégration menu	Cours Prolog	Manuel de déploiement, Planning réel
14-16h	Réseaux Projet Freenet	TD Complexité		TD Prolog	Finalisation rapport
16-18h	**Réunion externe n°6**	Débuggage intégration menu	Débuggage intégration menu	TP Prolog	*Fin itération n°4* **Livraison Finale n°4**

15 Bilan de la planification

D'un point de vue général, le planning prévisionnel a été respecté en partie seulement. En effet, le projet et les différentes tâches prévues étaient un peu trop ambitieuses *a priori*, compte tenu du grand nombre d'options proposées par le client et du peu de temps imparti au projet.

Du point de vue de la chronologie des tâches, le planning prévisionnel a également été respecté en partie : en effet les activités prévues pour les itérations 1 et 2 ont eu lieu comme prévu et ont donné lieu aux livraisons 1 et 2 (bien que le livrable n°2 ait été livré avec 3 jours de retard). En ce qui concerne la troisième itération, son contenu a été inversé avec celui de l'itération n°2, en raison du fait que certaines options prévues pour l'itération n°3 avaient déjà été implémentées lors de la seconde itération. D'autre part, nous avons consacré un nombre considérable d'heures en ce qui concerne l'implémentation du réseau, ce qui a inévitablement entraîné du retard dans la suite de nos diverses tâches.

Par ailleurs, étant donné que les différents projets extérieurs ont commencé à se télescoper à partir de la semaine du 17/11/08, l'itération n°3, initialement prévue de commencer le 17/11/08, n'a pas débuté avant le 24/11/08. De ce fait, et au vu de la complexité des différentes parties du jeu à implémenter, il a fallu passer plus de temps pour cette itération, qui s'est finalement terminée le 05/12/08, et qui a donné lieu à la troisième livraison le 05/12/08 (au lieu du 24/11/08). Durant cette semaine, aucune réunion externe n'a eu lieu étant donné que nous n'avions pas assez avancé dans l'élaboration du troisième livrable.

Pour ce qui est de la quatrième itération, prévue de commencer la semaine du 24/11/08, elle n'a pu commencer que le 08/12/08 (qui correspond à la dernière semaine du projet) à cause du retard accumulé. De plus, elle consistait initialement à l'implémentation de nouvelles options (notamment l'introduction des objets), mais a finalement été consacrée au débuggage de l'application ainsi qu'à la mise à jour et finalisation de la documentation.

En conclusion, il est clair que nos ambitions prévisionnelles ont dû être revues à la baisse : une partie des options du jeu (correspondant aux itérations n°5 et 6) n'a pas pu être implémentée certes, mais cependant un jeu *QTpilot* « de base », avec possibilité de jeu en réseau multi-joueurs et fonctionnalités de base (tirs, collisions, mode grand-prix) a été réalisé et fonctionne correctement, ainsi qu'une documentation (dans laquelle nous détaillons entre autres les différentes options non implémentées dans le dossier d'évolution) ont pu être livrés pour la date butoire du 12/12/08.

Sixième partie

Analyse financière

Objet : L'objectif de ce document est de présenter l'analyse financière du projet.

Auteurs :

Auteurs	**Approbateurs**	**Validation**
Stephanie MALAKIAN Zeina BAALBAKI	Christophe DEVIN Stéphane GRANGE	Laurent TICHIT
Livré le : *26/10/08*	Approuvé le : *24/10/08*	Validé le : *29/10/08*

Diffusion :

Diffusion	*Externe*
À :	Laurent TICHIT
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Stephanie MALAKIAN, Michael REZAC.

Historique :

Version du document	**Date de modification**	**Modifications apportées**	**Auteurs des modifications**
V1.0	23/10/08	Création du document	Zeina BAALBAKI, Stephanie MALAKIAN
V2.0	12/12/08	Finalisation du document	Christophe DEVIN, Stephanie MALAKIAN

Voici un tableau récapitulant la liste des contenus des différents livrables[6], leur coût ainsi que leur date de livraison :

N° Livraison	N° des lots constituant le livrable	Contenu du livrable	Date de livraison prévue **Date de livraison réelle ✓**	Nb. heures de travail	Coût (TTC en €)
1✓	1 et 2	– Version initiale du jeu avec les bases – Documentation initiale	07/11/08 08/11/08 ✓	30 h	4000 €
2✓	3 et 4 5 et 6 ✓	– Nouvelle version du jeu – Documentation mise à jour	14/11/08 19/11/08 ✓	25 h	3000 €
3✓	5 et 6 3 et 4 ✓	– Ajout de caractéristiques complexes (paramètres) – Documentation mise à jour	24/11/08 05/12/08 ✓	40 h	6000 €
4✓	7 et 8	– Ajout d'options diverses – Documentation mise à jour	28/11/08 12/12/08 ✓	30 h	4000 €
5 ✗	9 et 10	– Possibilité du choix du mode de jeu – Documentation mise à jour	05/12/08	25 h	3000 €
6 ✗	11 et 12	– Amélioration du graphisme et ajout de diverses options – Documentation mise à jour	12/12/08	60 h	8000 €
Montant Total :					**28000 €**

TAB. 5 – Analyse financière du projet « QTpilot ».

Le premier livrable représente le « socle » de base de l'application : il contient le lot n°1 qui représente les fonctionnalités importantes de base du jeu (avec notamment la création du graphisme et la mise en place du réseau à deux joueurs), ainsi que le lot n°2 qui est la première version de la documentation accompagnant le produit.

Le second livrable est constitué des lots n°3 et 4, dont notamment le rapport complété (Cahier des Charges, DAF et DAT).

Le troisième livrable nécessite plus de temps que les autres livrables, des options plus complexes devant être implémentées.

Le livrable n°4 contient une nouvelle version de l'application, améliorée avec de nouvelles options.

[6]Le contenu détaillé des lots se trouve à la section III à la page 13.

Le livrable n°5 contient les lots n°9 et 10, le lot n°9 étant l'application qui propose les différents modes de jeu, ainsi que la création du menu.

Enfin, le dernier livrable correspond à la version finale du jeu (lot n°11) ainsi que de la documentation l'accompagnant (lot n°12), avec notamment les dossiers d'évolution et de maintenance, ainsi qu'une fiche de tests.

Septième partie

Spécifications de l'interface

Objet : L'objectif de cette partie est de spécifier les besoins du client d'un point de vue de l'interface graphique de l'application « QTpilot ».

Auteurs :

Auteurs	Approbateurs	Validation
Stéphane GRANGE Christophe DEVIN	Zeina BAALBAKI Fouzi BENMAKHLOUF	Laurent TICHIT
Livré le : *12/12/08*	Approuvé le : *12/12/08*	*En attente de validation*

Diffusion :

Diffusion	*Externe*
À :	Laurent TICHIT
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Stephanie MALAKIAN, Michael REZAC.

Historique :

Version du document	Date de modification	Modifications apportées	Auteurs des modifications
V1.0	21/10/08	Création du document	Christophe DEVIN , Stéphane GRANGE
V2.0	10/12/08	Mise à jour des screenshot	Zeina BAALBAKI, Stephanie MALAKIAN

Voici un aspect général de l'application d'un point de vue graphique.

16 Page d'accueil

La page d'accueil propose de lancer le jeu « QTpilot » à proprement dit.

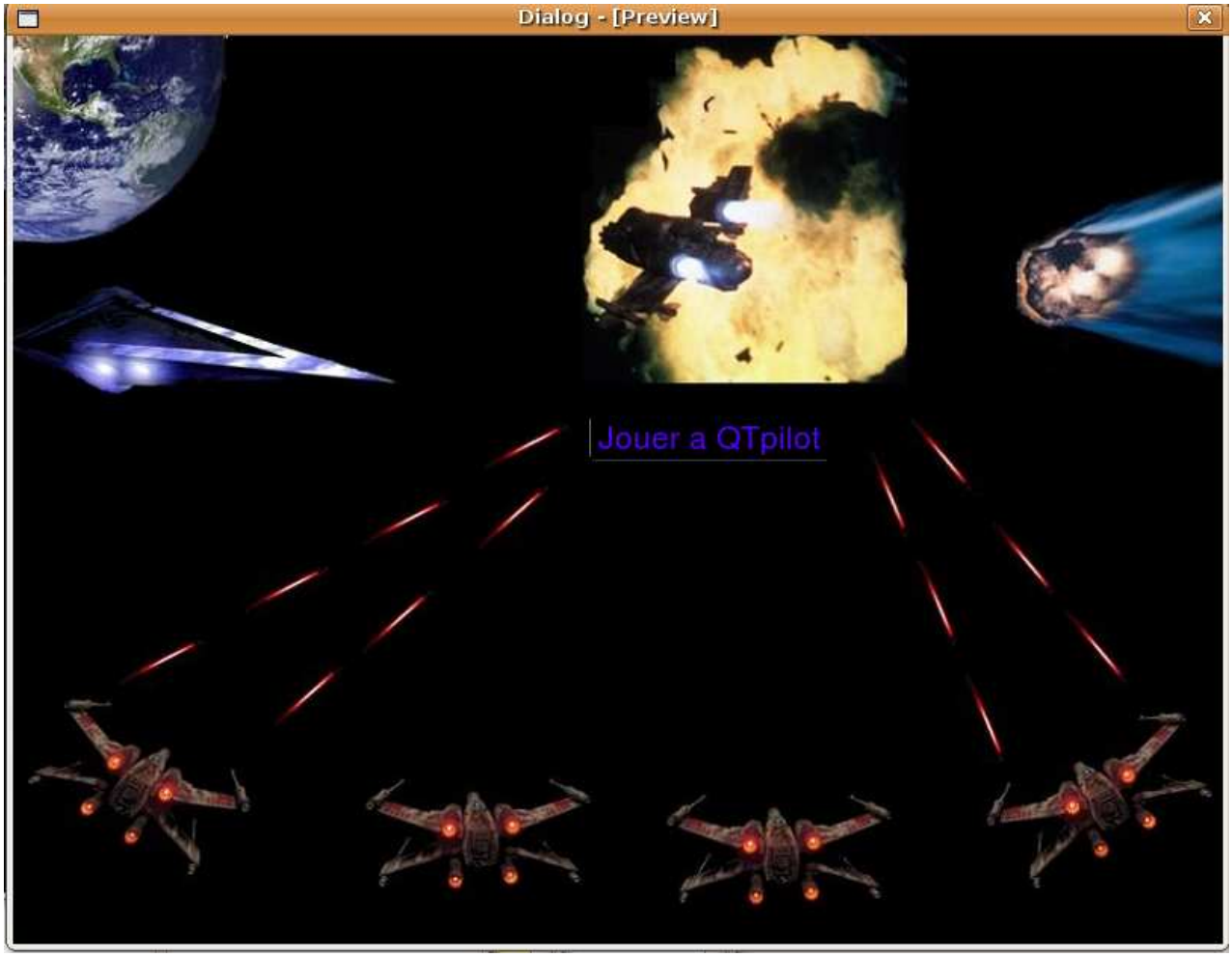
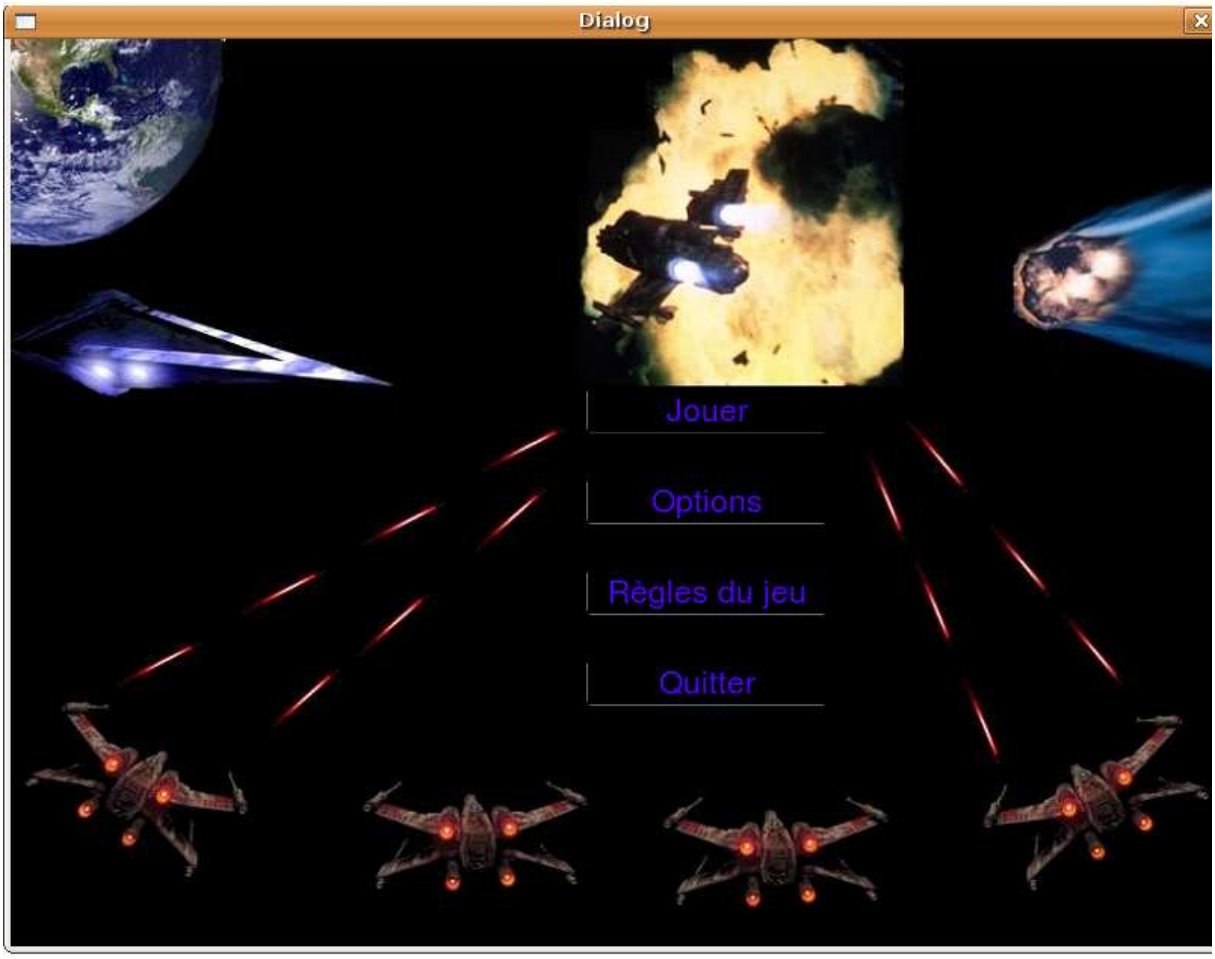


FIG. 7 – Page d'accueil de « QTpilot ».

Ensuite, l'application propose de lancer le jeu à proprement dit (en d'autres termes, de commencer une partie : jouer), mais également de choisir ses options, ou encore de consulter les règles du jeu de « QTpilot ». Elle propose également de quitter l'application.



17 Règles du jeu

Cette page permet de détailler les règles de *QTpilot*, et de revenir en arrière.

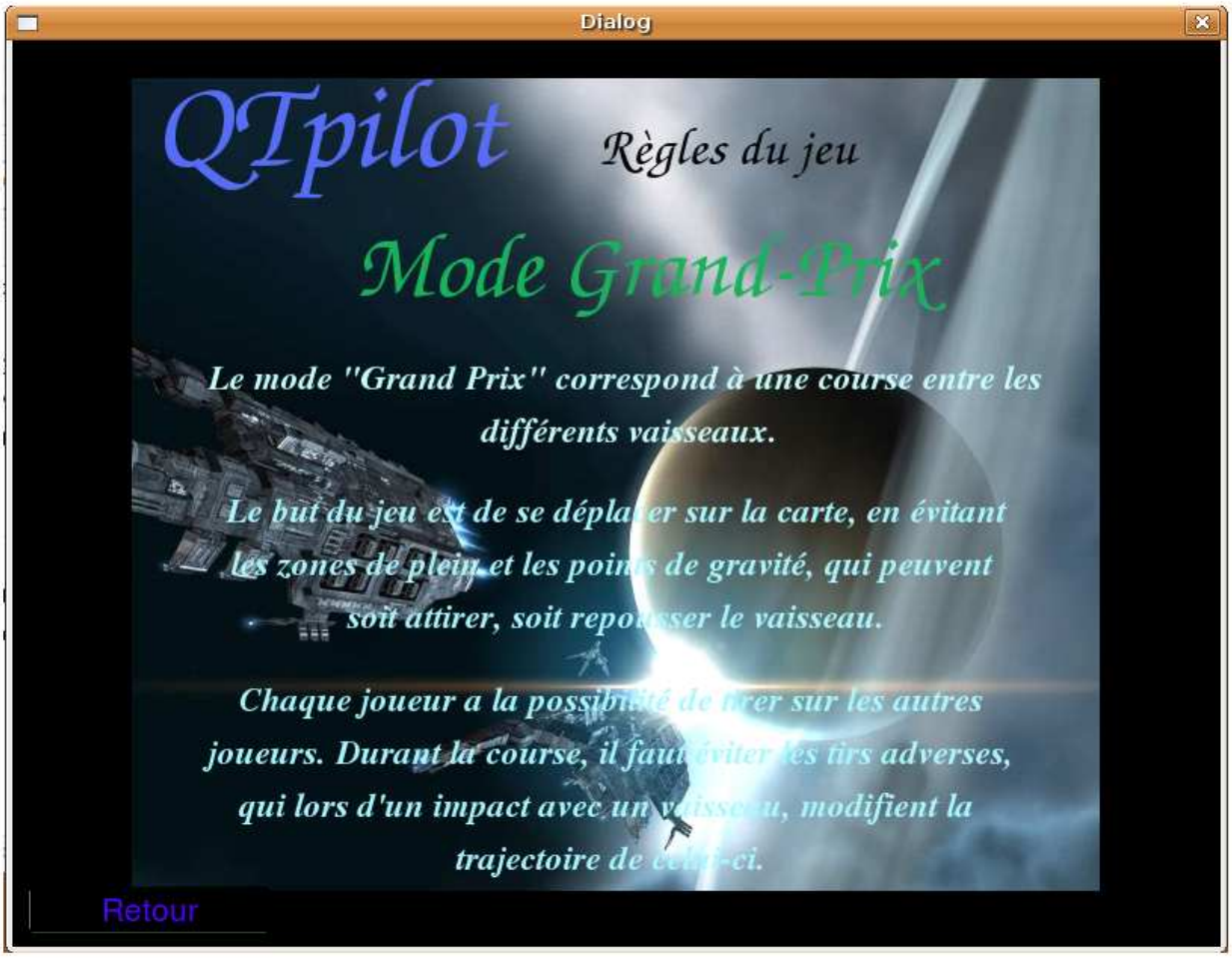


FIG. 8 – Page des options de « QTpilot ».

18 Page des options

La page des options permet à l'utilisateur de choisir ses préférences (comme par exemple modifier son pseudo).

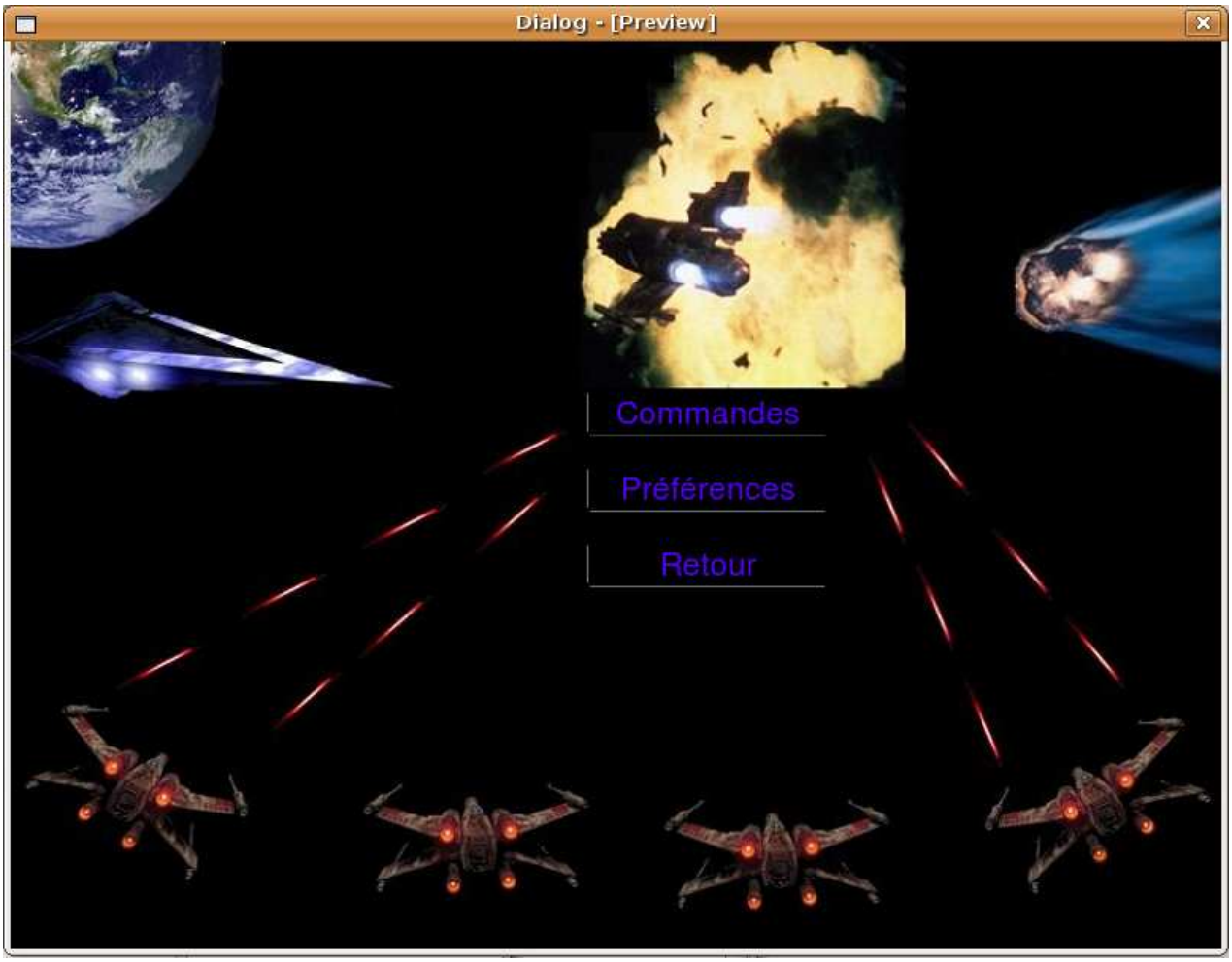


Fig. 9 – Page des options de « QTpilot ».

19 Page du jeu

Pour jouer, l'utilisateur doit pouvoir choisir s'il préfère jouer seul (entraînement) ou en réseau avec d'autres joueurs. Le mode « Entraînement » correspond en fait à un seul joueur (un client connecté au

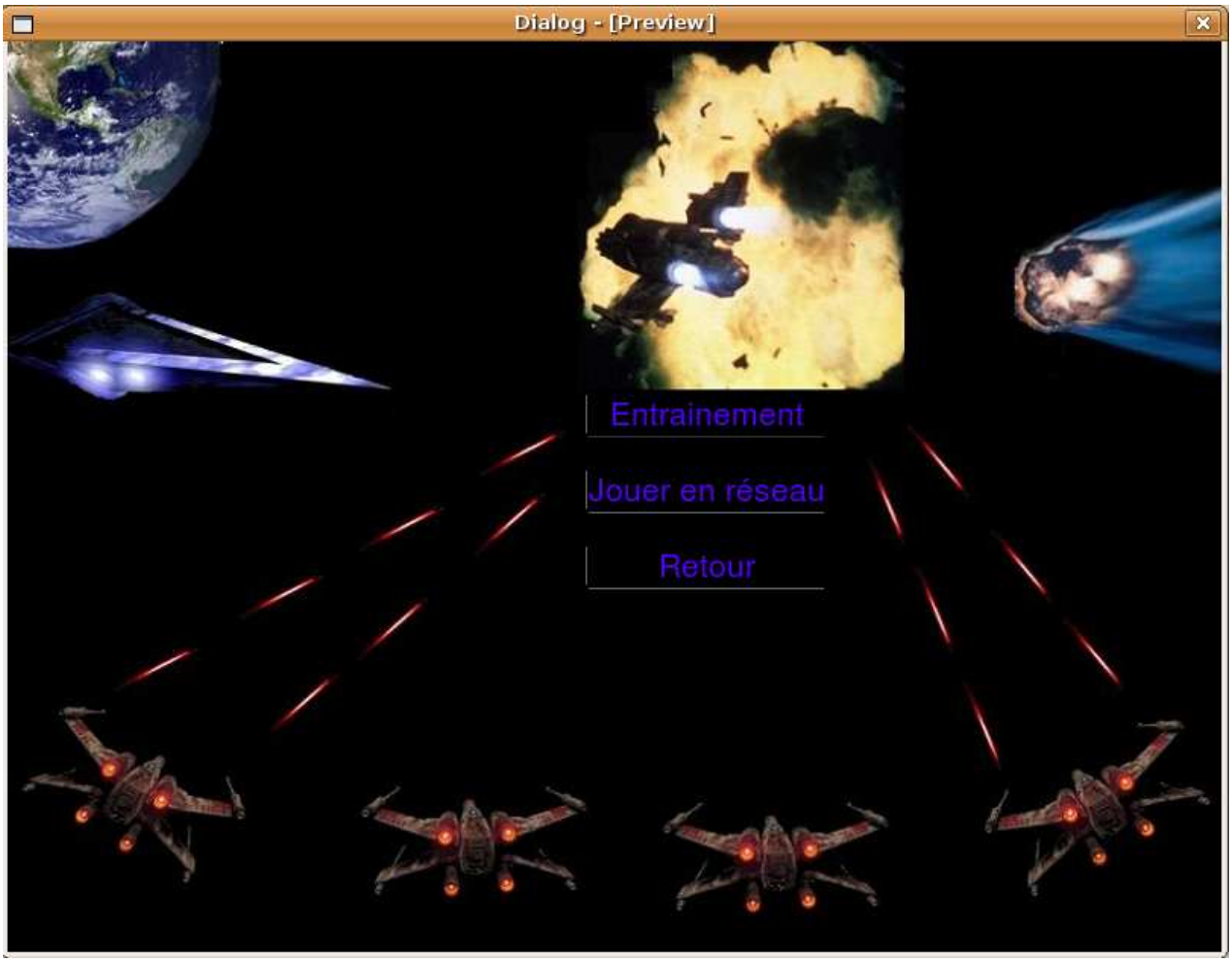


FIG. 10 – Page de jeu de « QTpilot ».

serveur, dont le rôle est joué par l'utilisateur en question) qui évolue sur la carte.

19.1 Jeu en réseau

Si l'utilisateur choisit de jouer en réseau, alors il a deux possibilités : soit il créé une partie (qui créera un serveur), soit il choisit de rejoindre une partie (ce qui suppose qu'un serveur a déjà été lancé, et que l'utilisateur qui souhaite rejoindre la partie connaisse l'adresse IP du serveur).

19.1.1 Création d'une partie

Dans le cas de la création d'une partie, l'écran suivant s'affiche alors : Il s'agit du choix de la carte

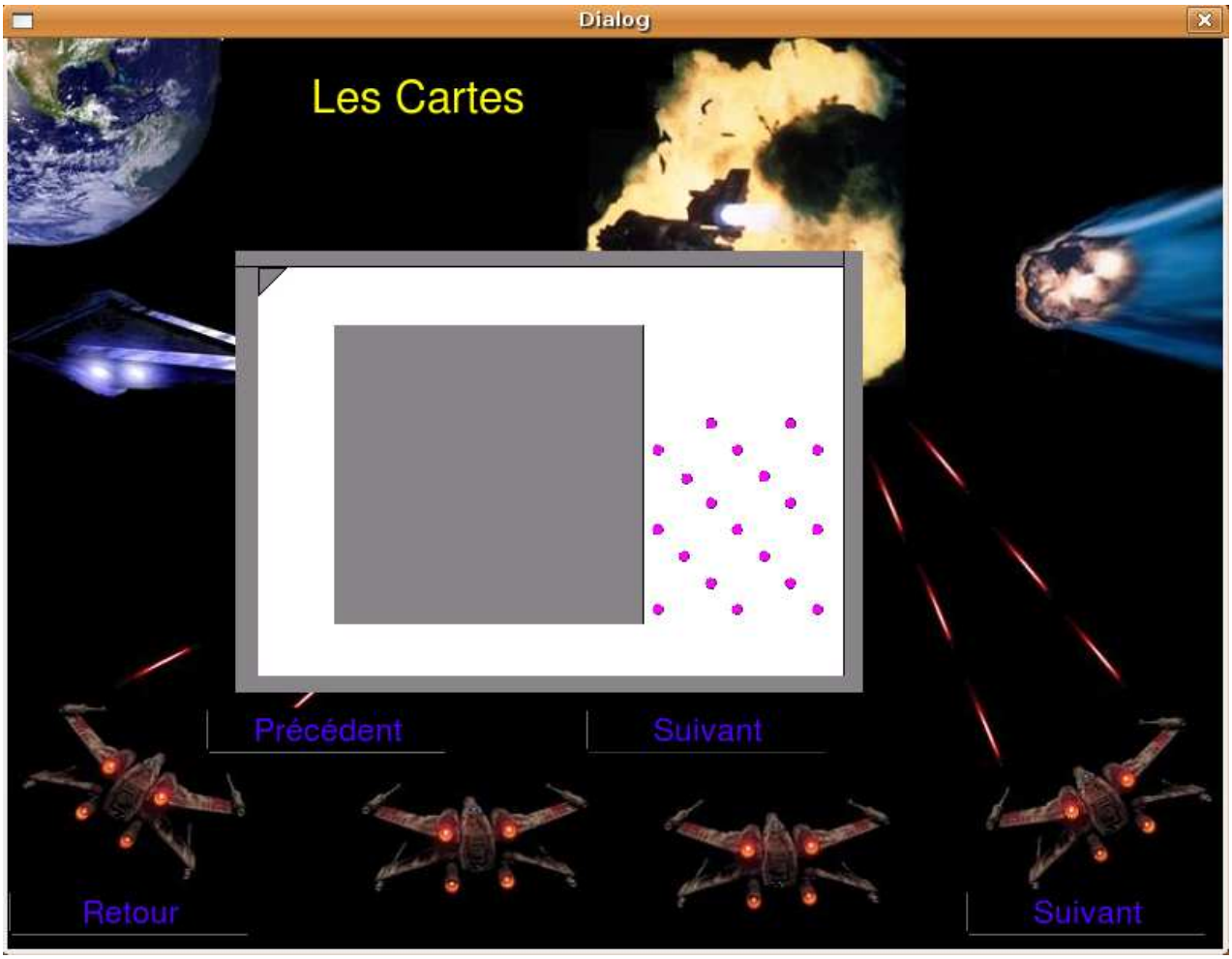


FIG. 11 – Choix de la carte de la partie.

pour le jeu (une seule carte dans la version finale de *QTpilot*).

19.1.2 Rejoindre une partie

Si l'utilisateur choisit de rejoindre une partie, alors la boîte de dialogue suivante s'affiche : Elle

FIG. 12 – Saisie de l'adresse IP du serveur.

propose à l'utilisateur de saisir l'adresse IP du serveur auquel il souhaite se connecter pour rejoindre la partie.

Si l'utilisateur ne saisit aucune adresse IP mais clique toutefois sur le bouton OK, alors une alerte indiquant un message d'erreur apparaît :

FIG. 13 – Message d'alerte en cas de non saisie de l'adresse IP et validation.

Huitième partie

Dossier d'Analyse Fonctionnelle

Objet : L'objectif de ce document est de spécifier les besoins du client en les modélisant sous la forme de diagrammes UML.

Auteurs :

Auteurs	**Approbateurs**	**Validation**
Michael REZAC Fouzi BENMAKHLOUF	Zeina BAALBAKI Stéphane GRANGE	Laurent TICHIT
Livré le : *12/12/08*	Approuvé le : *12/12/08*	*En attente de validation*

Diffusion :

Diffusion	*Externe*
À :	Laurent TICHIT
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Stephanie MALAKIAN, Michael REZAC.

Historique :

Version du document	Date de modification	Modifications apportées	Auteurs des modifications
V1.0	27/10/08	Création du document	Fouzi BENMAKHLOUF, Michael REZAC
V1.1	29/10/08	Première version des cas d'utilisations	Zeina BAALBAKI, Christophe DEVIN, Stephanie MALAKIAN
V1.2	12/11/08	Diagramme de séquence V1.0	Zeina BAALBAKI, Michael REZAC
V1.3	12/11/08	Diagramme des cas d'utilisations V2.0	Fouzi BENMAKHLOUF, Christophe DEVIN
V1.4	12/11/08	Diagramme métier	Stéphane GRANGE, Stephanie MALAKIAN
V2.0	03/12/08	Mise à jour	Zeina BAALBAKI, Christophe DEVIN, Stephanie MALAKIAN, Michael REZAC
Vfinale	12/12/08	Mise à jour finale du document	Zeina BAALBAKI, Christophe DEVIN

20 Cas d'utilisations

Cette section présente et détaille les différents cas d'utilisations possibles pour le jeu « QTpilot ».

20.1 Cas d'utilisation général

De façon générale, l'application doit permettre à un joueur de :

- consulter les règles du jeu de « QTpilot »,
- effectuer des réglages concernant ses préférences, comme saisir son pseudo,
- jouer à *QTpilot*.

Voici donc le diagramme des cas d'utilisations dans le cadre général du jeu :

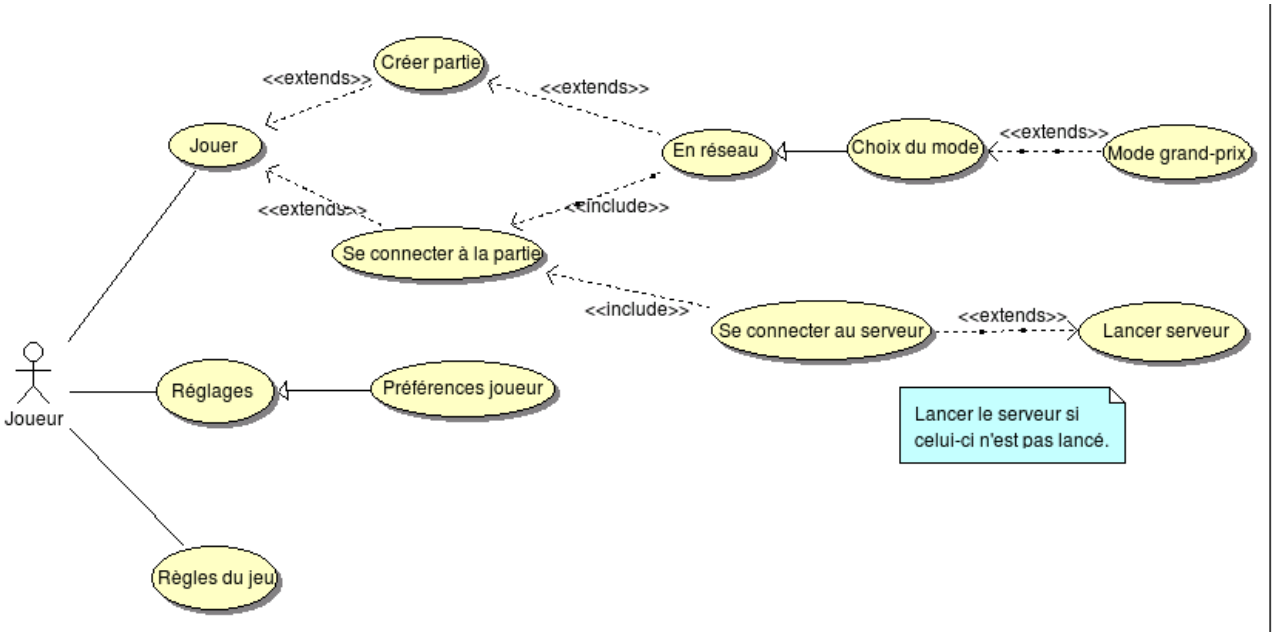


Fig. 14 – Cas d'utilisation général du jeu.

Lorsqu'un joueur décide de jouer, il a le choix entre créer la partie, et rejoindre une partie. Dans le cas de la création d'une partie, le joueur choisit son mode de jeu, à savoir « Grand-Prix ».
Dans le cas où le joueur rejoint une partie, il doit alors se conecter au serveur qui héberge la partie. Dans le cas où ce-dernier n'est pas lancé, il faut le lancer pour que le joueur puisse jouer.

20.2 Cas d'utilisation « jouer »

Intéressons-nous de plus près au cas d'utilisations « Jouer ». Lorsqu'un joueur joue une partie, il pilote un vaisseau. Plusieurs cas sont alors possibles :

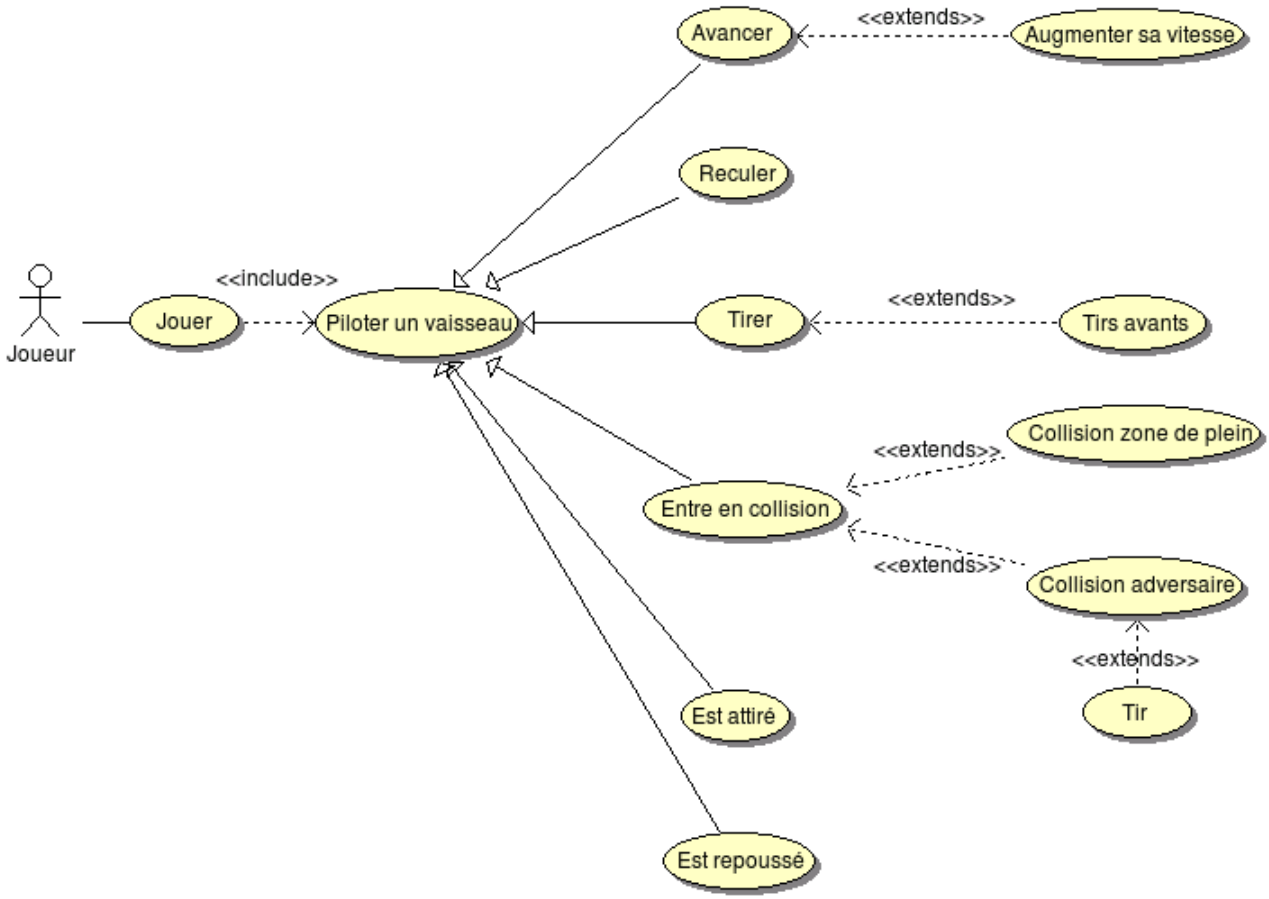


Fig. 15 – Cas d'utilisation « jouer ».

Le joueur peut avancer (en éventuellement augmentant sa vitesse), reculer, tirer (par tirs avant), mais également entrer en collision (soit contre une zone de plein, soit contre le tir d'un joueur adverse), être attiré ou repoussé par des points de gravité.

21 Diagramme de séquence

Le diagramme de séquence qui suit représente le cas d'utilisation de « jouer en réseau ».

Un utilisateur qui créé une partie, créé un serveur, qui lorsqu'il est lancé, se met en attente de connexion. Un nouveau client est alors automatiquement créé et se connecte au serveur.
Lorsqu'un utilisateur souhaite rejoindre une partie déjà existante, un nouveau client va tenter de se connecter au serveur en passant en argument une adresse IP : c'est ce que modélise la trame seConnecte(IP). Si l'adresse IP est valide, et que le nombre de clients connectés est inférieur à 3, alors le serveur accepte la connexion et envoie au client un identificateur (accepte(Id_Client)). Ensuite, et ce toutes les 50 millisecondes, le client envoie au serveur un paquet contenant ses informations (envoiePaquet(Paquet)). Instantanément, le serveur renvoie le(s) paquet(s) reçu(s) à tous ses clients (renvoiePaquets()).

Dans le cas où l'IP n'est pas valide, le client affiche un message d'erreur, et le serveur reste en attente de connexion.

Dans le cas où le nombre de clients connectés est supérieur ou égal à trois, si un nouveau client se connecte, alors le serveur va refuser la connexion, et se remettre en attente de connexion.

Voici donc le diagramme de séquence correspondant :

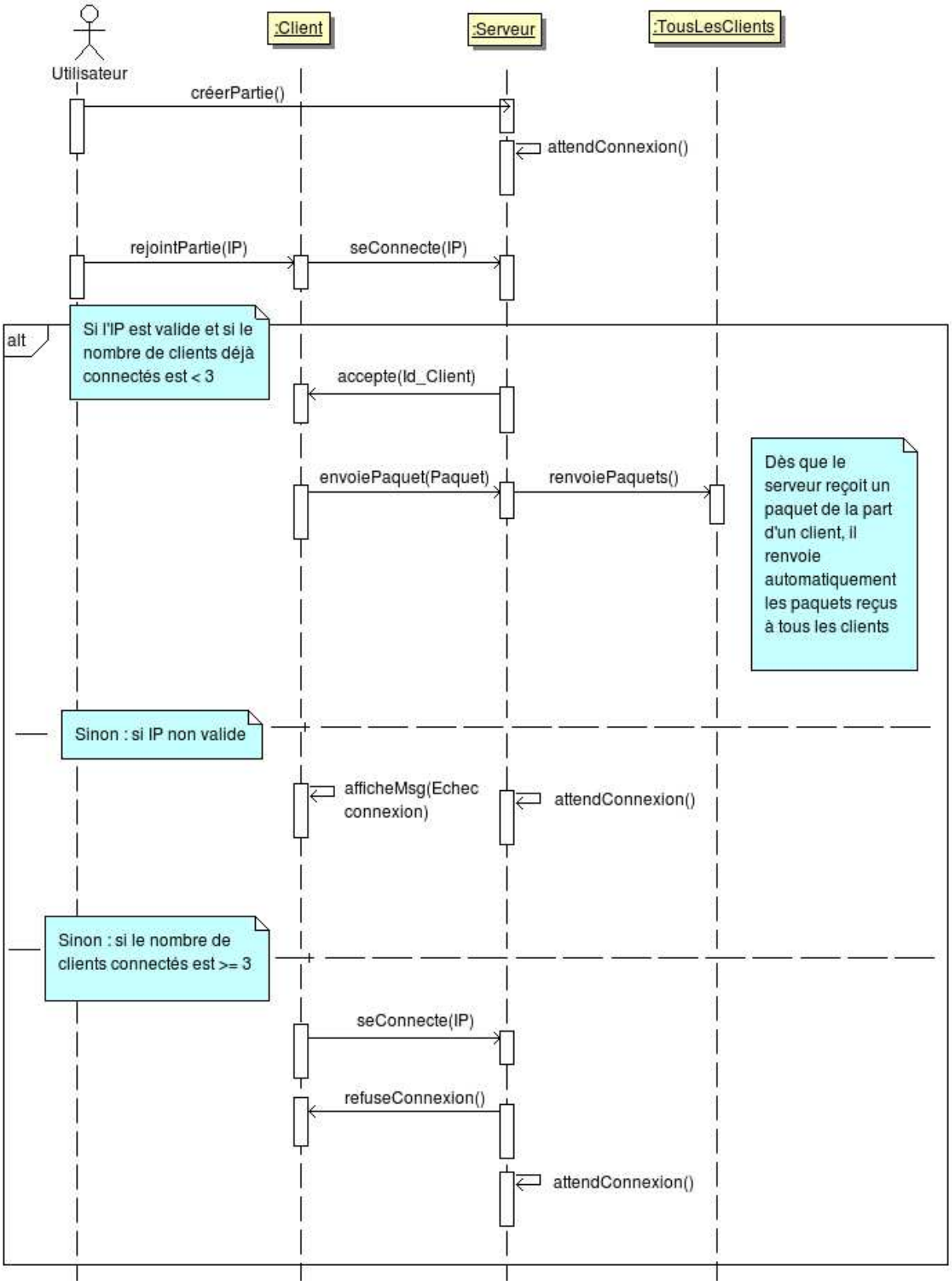


FIG. 16 – Diagramme de séquence « jouer en réseau ».

22 Diagramme métier

Afin d'avoir un meilleur aperçu des différentes classes constituant l'application, nous avons choisit de représenter certaines classes d'une certaine couleur :

- en rose sont représentées les deux classes destinées à modéliser le réseau,
- en vert clair, il s'agit des classes qui concernent le graphisme de l'application,
- en jaune, les autres classes nécessaires à l'application.

Comme nous pouvons le lire sur le diagramme métier qui se situe à la page suivante, l'application est constituée de sept classes :
Les classes Client, Serveur servant à gérer le réseau de l'application, la classe Map qui représente la carte représentant l'espace de jeu, les classes Wall et Gravity qui représentent respectivement les zones de plein sur la carte, et les points de gravité, la classe Player qui modélise le joueur, plus précisément son vaisseau, et enfin la classe Shoot qui permet de représenter les tirs d'un joueur.

▷ Classes gérant le réseau dans l'application

Les classes Client et Serveur modélisent le réseau dans l'application.
Le client ne peut pas exister sans la map, car il récupère, en passant par la map, les données du joueur (informations sur les points de gravité, les tirs, ainsi que les positions, vitesses, accélérations, puissance, rotation) qui vient constituer le paquet a envoyer au serveur à un intervalle de temps régulier. De plus, il est clair que le client ne peut pas non plus exister sans le serveur.
Pour ce qui est du serveur, il ne se préoccupe pas des données de la map, il se contente seulement de recevoir des paquets de la part de tous les clients et des les renvoyer a tous les clients, de façon a ce que la map soit mise a jour par chaque client.
Le réseau implémenté est un réseau multi-joueurs limité à 3 joueurs, ce qui explique les cardinalités de 3 dans le diagramme.

▷ Classes gérant le graphisme dans l'application

Les classes Map, Wall et Gravity font partie des classes qui sont fortement liées au graphisme de l'application.
En effet, les méthodes les constituant sont essentiellement des méthodes destinées au graphisme du jeu.

▷ Autres classes de l'application

Enfin, les classes Player et Shoot concernent plus précisément le joueur : les différents attributs de la classe Player représentent les informations du joueur, quant aux méthodes, elles concernent directement ces informations. Par ailleurs, un joueur fait partie du graphisme de la map, ce qui explique le lien entre les classes Player et Map.
Pour ce qui est des tirs, ils ne peuvent exister sans qu'ils ne soient associés à un joueur.

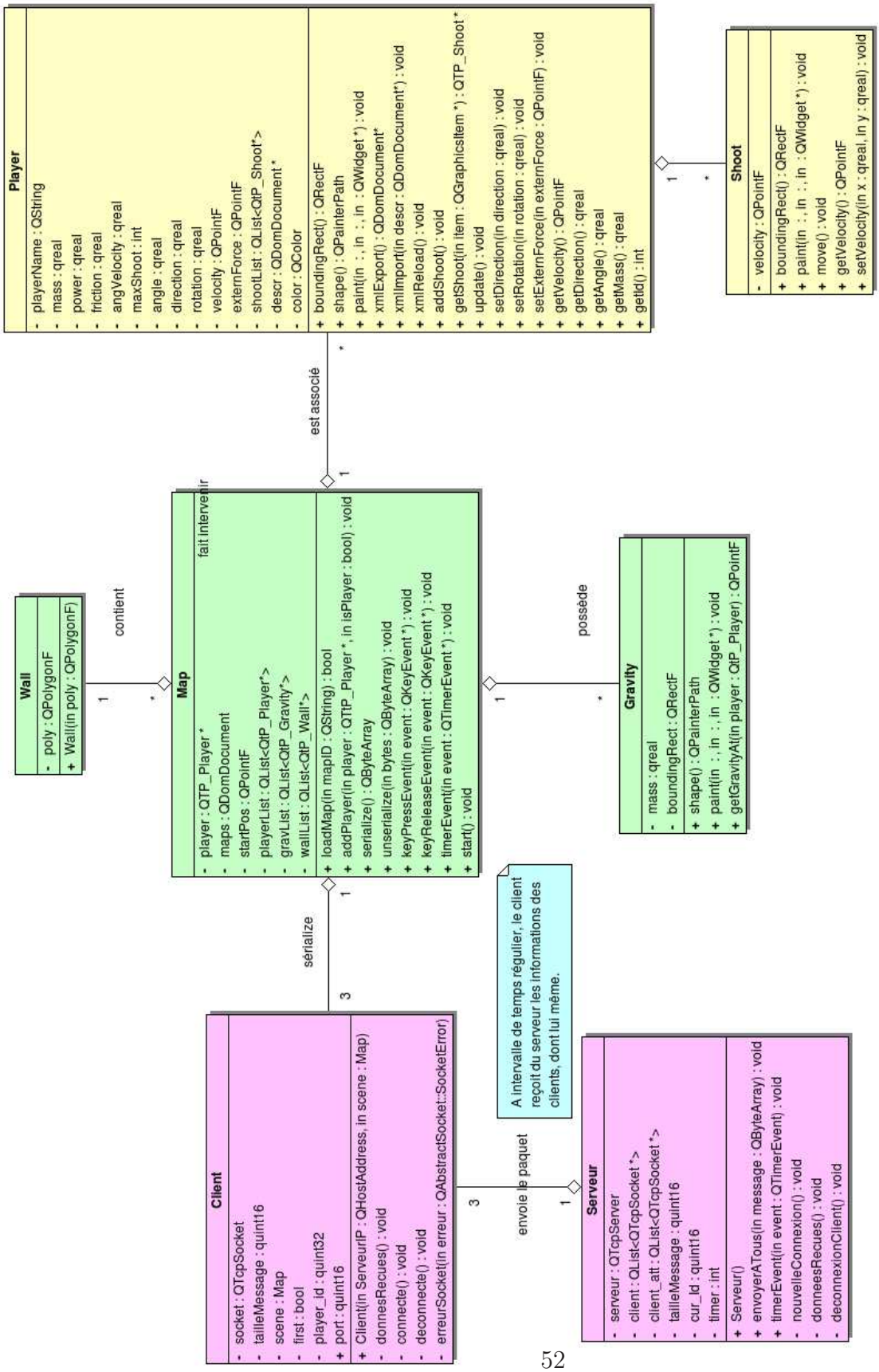


FIG. 17 – Diagramme métier.

23 Diagramme des classes

Après avoir détaillé le diagramme métier de l'application, voici ci-après le diagramme des classes de l'application.

Les classes représentées en vert foncé correspondent en fait à des interfaces Qt desquelles nos classes représentant l'application vont hériter.

Cela concerne essentiellement les classes destinées au graphisme de l'application (en vert clair sur les diagrammes), mais aussi les classes Player et Shoot.

Les classes Wall, Player et Shoot héritent de la classe **QGraphicsItem** : les méthodes qui sont surchargées sont les méthodes : boundingRect(), shape() et paint().

La classe Map quant à elle hérite de la classe **QGraphicsScene**, qui elle-même hérite de la classe QObject. Les méthodes KeyPressEvent et KeyRelease de QObject ont été surchargées.

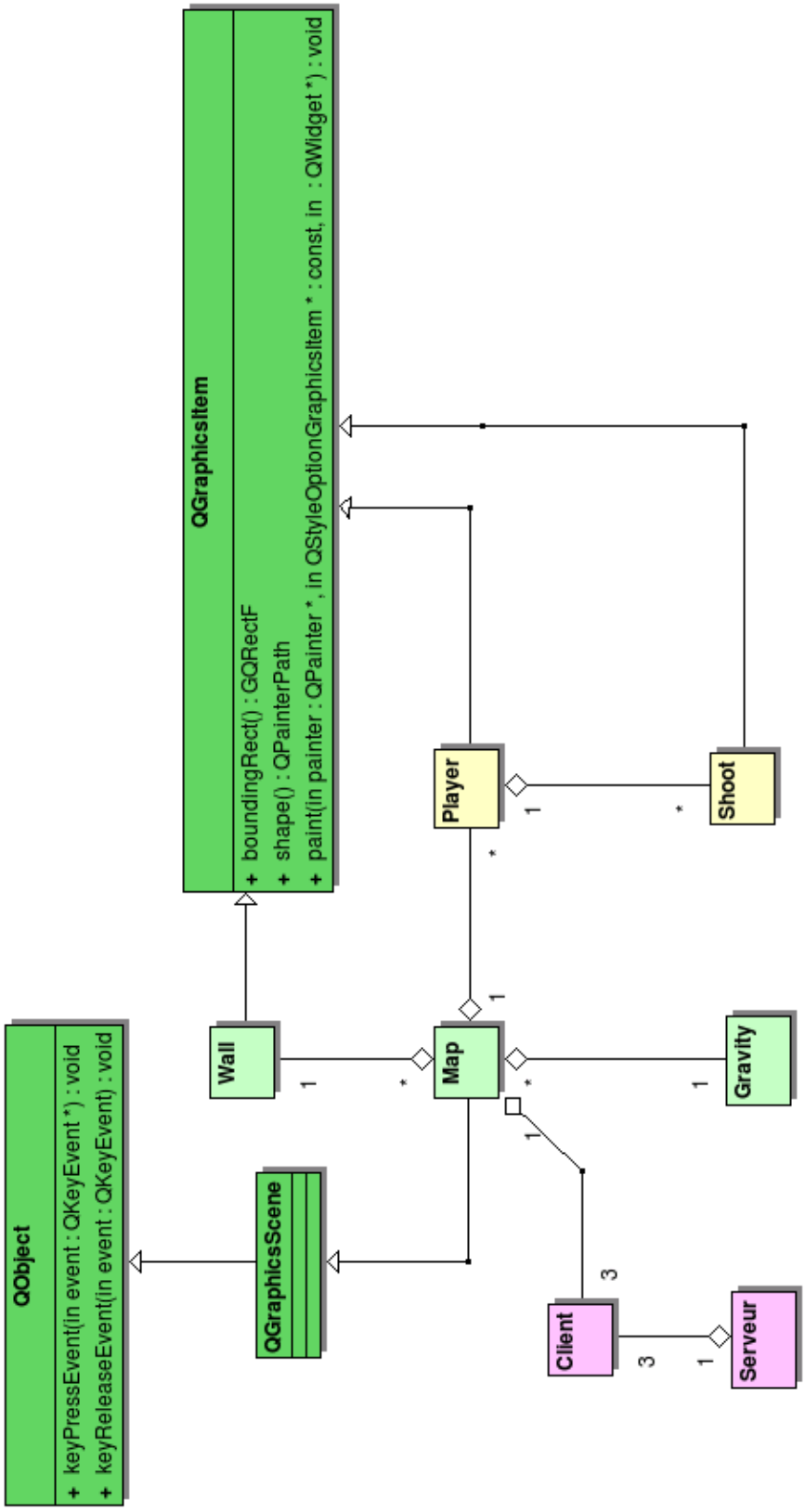


FIG. 18 – Diagramme des classes.

24 Diagramme de navigation

Voici le diagramme de navigation de l'application, qui représente l'IHM (Interface Homme-Machine) de l'application :

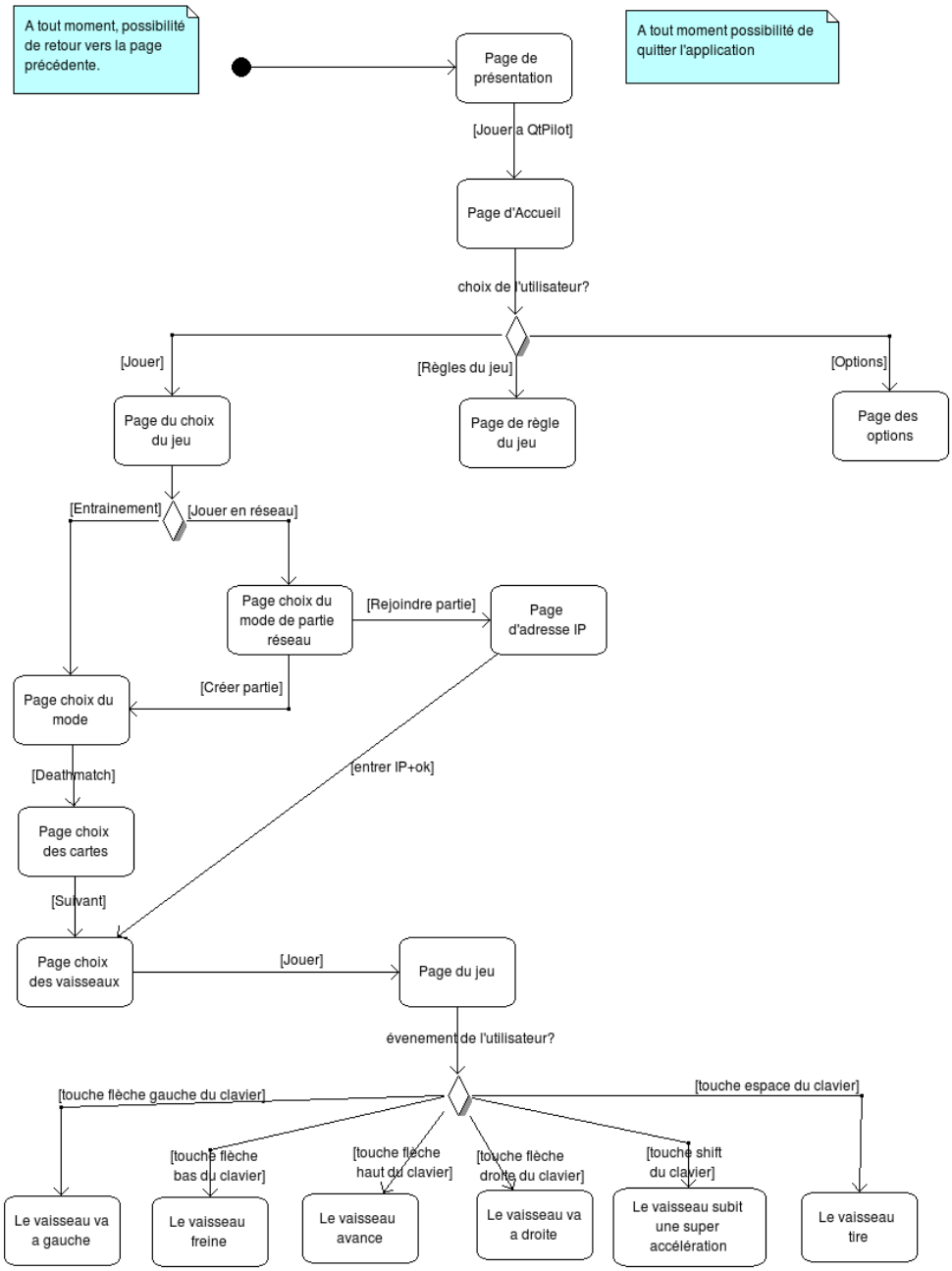


FIG. 19 – Diagramme de navigation.

25 Diagramme de déploiement

Voici le diagramme de déploiement de l'application :

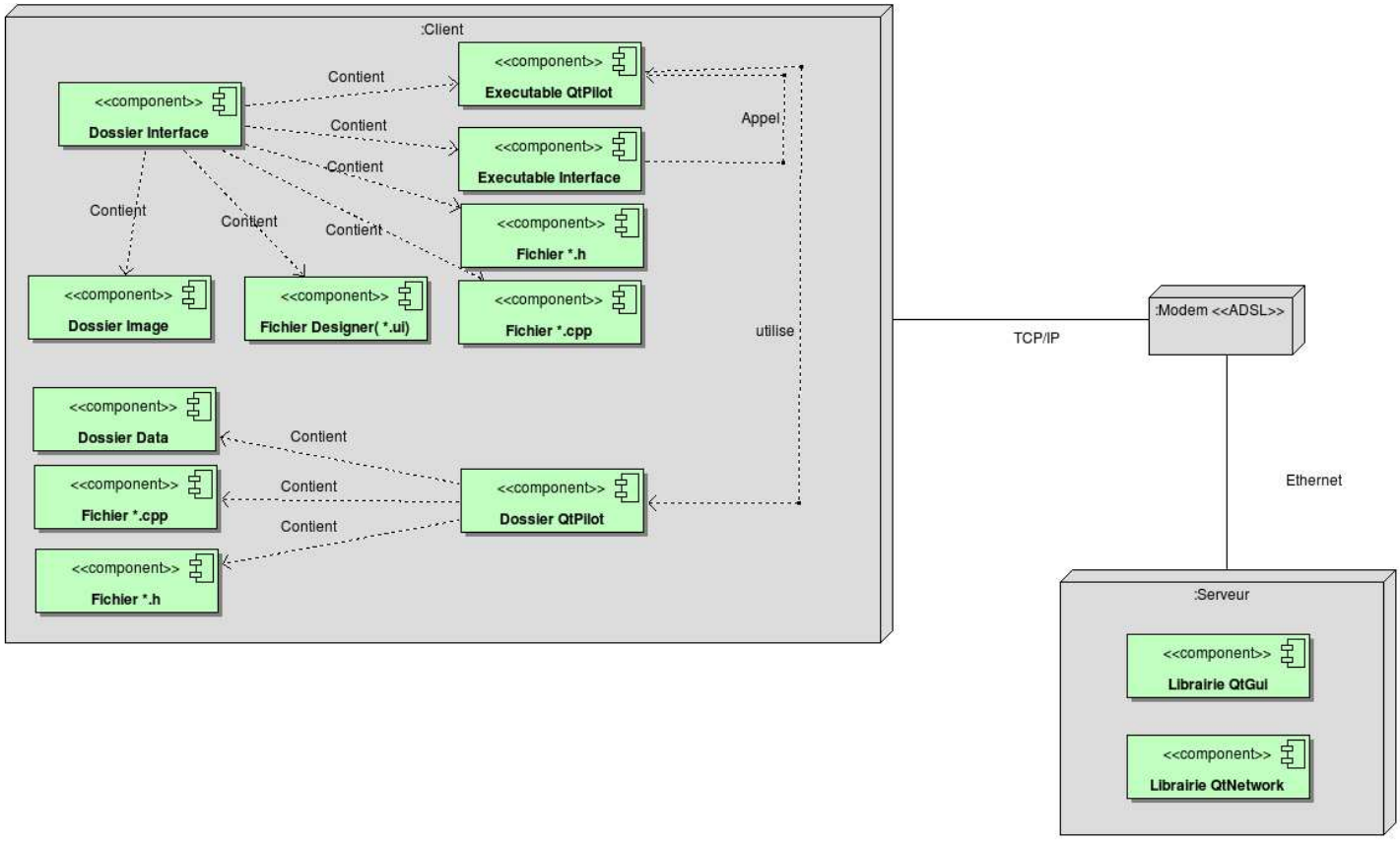


FIG. 20 – Diagramme de déploiement.

Le diagramme de déploiement ci-dessus montre la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels.

Les deux nœuds *Client* et *Serveur* y sont représentés, tous deux connectés via le réseau Internet par l'intermédiaire d'une connexion TCP/IP.

Le client a deux dossier importants : l'un comportant son interface graphique et l'autre comportant le jeu QtPilot à proprement parlé. Après avoir été compilé dans son propre dossier, l'exécutable de QtPilot va dans le dossier interface.
Quand le joueur lance son interface et quand il crée une partie, il crée un serveur auquel les clients pourront se connecter grâce aux bibliothèques Qt et à l'éxécutable de l'application ainsi déplacé dans le dossier.

Neuvième partie

Dossier d'Analyse Technique

Objet : L'objectif du Dossier d'Analyse Techique est de justifier les choix techniques (matériel ou logiciel) qui ont été faits au cours du développement du système.

Auteurs :

Auteurs	**Approbateurs**	**Validation**
Fouzi BENMAKHLOUF Stéphane GRANGE	Zeina BAALBAKI Christophe DEVIN	Stephanie MALAKIAN
Livré le : *12/12/08*	Approuvé le : *12/12/08*	Validé le : *12/12/08*

Diffusion :

Diffusion	*Interne*
À :	Stephanie MALAKIAN (chef de projet)
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Michael REZAC.

Historique :

Version du document	**Date de modification**	**Modifications apportées**	**Auteurs des modifications**
V1.0	05/11/08	Création du document	Fouzi BENMAKHLOUF, Stéphane GRANGE
V1.1	12/12/08	Finalisation du document	Zeina BAALBAKI, Michael REZAC

26 Choix matériels

26.1 Les outils de travail

Voici la liste des différents outils utilisés tant bien pour l'organisation et le développement que pour le suivi du projet :

- **Systèmes d'exploitation** : GNU Linux 2.6, Windows XP, Windows Vista.
  Le travail de développement au niveau de l'implémentation a pu être réalisé sous ces trois systèmes d'exploitation, ce qui garantit une certaine liberté d'utilisation du jeu « QTpilot », quelque soit le système d'exploitation de l'utilisateur du jeu. Liens : www.gnu.org/gnu/linux-and-gnu.fr.html, www.microsoft.com/france/windows/

- **Planification** : Planner 0.14.
  Ce logiciel libre nous a permis la gestion de la planification du projet, avec notamment la création des diagrammes de Gantt ainsi que la gestion des ressources. Lien : live.gnome.org/Planner

- **Modélisation UML** : BOUML.
  Nous avons considéré cet outil de modélisation UML comme étant stable, offrant l'autocomplétion et qui de plus permet la génération automatique de code, ce qui en fait donc un outil efficace pour la phase d'analyse du projet. Lien : bouml.free.fr

- **Génération de documentation de code** : DOXYGEN.
  Cet outil libre, nous a permis, à partir des commentaires que nous avons écrits dans nos codes sources, de générer automatiquement la documentation de notre code. De plus, cet outil est facilement paramétrable, et offre ainsi de nombreuses spécificités pour la documentation (comme par exemple la génération automatique des graphes de dépendances ou des graphes d'héritage). Lien : www.doxygen.org

- **Gestion des versions** : SVN (Subversion).
  Un repository (ou dépôt) a été créé via Google afin que tous les membres de l'équipe puisse accéder ou modifier les divers documents (documentation, code) qui constituent le projet. Lien : subversion.tigris.org

- **Mise en page du rapport** : LaTeX.
  La puissance de LaTeX (qui est également un outil libre) a été exploitée pour mettre en page la documentation accompagnant l'application en elle-même. Lien : www.latex-project.org

27 Choix logiciels

27.1 Interface graphique du menu

L'interface graphique correspondant au menu permettant de lancer le jeu a été réalisée avec Qt, version 4.4.x.
Les différents écrans de l'application sont représentés par des **QDialog**.

27.2 La gestion du réseau

L'envoi des différentes informations de chaque client connecté (position du vaisseau, vitesse, accélération, tirs) au serveur, et de la réponse de celui-ci qui renvoie les informations à tous les clients afin de les mettre à jour) est géré via des **QTcpSocket**.
De plus, le seveur gère des listes de clients, grâce à l'utilisation de **QList**.
Pour gérer les paquets, nous avons choisit d'utiliser des **QByteArray** que nous avons jugé assez pratique (étant donné que l'on peut y mettre les objets que l'on veut).

27.3 Le graphisme de l'application

27.3.1 Création des divers objets graphiques de l'application

Pour créer les différents objets graphiques de *QTpilot* tels que le vaisseau ou les zones de plein, nous avons utilisé l'interface **QGraphicsItem** notamment dans nos classes QtP_Wall, QtP_Player ainsi que QtP_Shoot.

27.3.2 Affichage des divers objets graphiques

Pour afficher les objets graphiques, nous utilisons l'interface **QGraphicsScene** dans notre classe QtP_Map.

27.3.3 Gestion des cartes

D'autre part, en ce qui concerne les différentes cartes possibles du jeu, elles sont gérées via un fichier xml dont la structure est définie par un schéma xsd. Ainsi, chaque map est représentée par un élément, et peut contenir des zones de plein (qui sont en fait des polygones, pour lesquels la couleur de fond ou de bord est paramétrable), mais également des points de gravité. La structure du fichier xml est décrite dans le manuel de maintenance (**partie X** à la **page 60**).

27.4 Version et composants livrés

La version de l'application livrée lors de la livraison finale est « QtPilot-vfinale » : cette version correspond à un jeu de base (un mode de jeu, réseau multi-joueurs, tirs et collisions).
Le dossier de suivi du projet « qtpilot-doc-vfinale.pdf » correspond à la version finale de la documentation complète accompagnant le jeu *QTpilot*.

Dixième partie

Manuel de Maintenance

Objet : L'objectif du manuel de maintenance est de mettre l'accent sur tout ce qui concerne l'implémentation de l'application (au niveau du code, et non plus au niveau de la conception).

Auteurs :

Auteurs	**Approbateurs**	**Validation**
Fouzi BENMAKHLOUF Michael REZAC	Zeina BAALBAKI Stéphane GRANGE	Stephanie MALAKIAN
Livré le : *12/12/08*	Approuvé le : *12/12/08*	Validé le : *12/12/08*

Diffusion :

Diffusion	*Interne*
À :	Stephanie MALAKIAN (chef de projet)
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Michael REZAC.

Historique :

Version du document	**Date de modification**	**Modifications apportées**	**Auteurs des modifications**
V1.0	05/11/08	Création du document	Fouzi BENMAKHLOUF, Michael REZAC
V1.2	19/11/08	Rédaction partie graphisme	Fouzi BENMAKHLOUF, Michael REZAC
V1.3	05/11/08	Rédaction partie réseau	Zeina BAALBAKI, Stephanie MALAKIAN

Voici les différentes classes constituant l'ensemble de notre code source et pour lesquelles nous expliquons certains de nos choix de codage. Une présentation plus détaillée du code source est disponible via la documentation automatiquement générée (par DOXYGEN) à partir de notre code source, dans le répertoire DOC livré lors de la livraison finale du produit.

28 Convention de codage

Toutes les classes constituant le projet « QTpilot » commencent par les lettres : QtP_ .

29 Partie graphisme

29.1 La classe QtP_Map.cpp

Cette classe permet de créer une carte sur laquelle se deplaceront les joueurs. Voici les méthodes principales de cette classe :

- loadMap : permet de charger une carte depuis son ID, cet ID est defini dans le fichier xml contenant les cartes. La méthode récupère le noeud contenant les informations de la carte, et la dessine à l'aide des outils **QPainter**. La méthode se termine en retournant *true* en cas de chargement réussi, *false* sinon.

- addPlayer : cette méthode permet d'ajouter un joueur à la carte. Il s'agit principalement d'ajouter l'instance de **QGraphicsItem** représentant le joueur à la **QGraphicsScene** représentant la carte. L'argument supplémentaire permet de déterminer si le joueur ajouté est contrôlé par la personne derrière le clavier ou s'il s'agit d'une autre personne en ligne. Dans le cas ou il s'agit de la personne présente, la carte va mémoriser le pointeur vers ce joueur.

- serialize : cette methode renvoie une représentation des informations du joueur.

- unserialize : cette méthode récupère un représentation des informations d'un joueur et lui transmet, de manière à ce qu'il puisse se mettre a jour.

- keypressevent / keyreleaseevent : comme leur nom l'indique, ces méthodes récuperent les évènements claviers afin d'appliquer un mouvement au joueur.

- start : démarre la mise à jour de la carte.

- timerEvent : cette méthode se déclenche à intervalle régulier et effectue plusieurs opérations pour maintenir la carte a jour :
  - détection des collisions à l'aide des méthodes implementées par **QGraphicsItem**
  - mise a jour des joueurs « réseaux » depuis leur représentation
  - déplacement du joueur.

29.2 La classe QtP_Player.cpp

Le constructeur de cette classe génère une instance de joueur à partir de son nom, id, et du nom du vaisseau. Bien que cela ne soit pas encore implementé, le nom du vaisseau permet d'utiliser des vaisseaux ayant différentes caractéristiques physiques. Voici les principales méthodes de cette classe :

- boundingRect : définit la zone de dessin du joueur.
- shape : définit la forme du vaisseau (pour les collisions).
- paint : dessine le vaisseau.
- xmlExport : renvoit un pointeur vers une description xml du vaisseau (position, vitesse, direction, tirs).
- xmlImport : reçoit et stocke une description xml, si une description était déjà stockée, elle est supprimée car plus à jour.
- xmlReload : applique la descrition xml enregistrée, si elle existe, puis la supprime.
- addShoot : crée un tir pour le joueur, à condition que celui-ci n'ait pas déjà atteint son nombre maximal de tirs simultanés.
- getShoot : retourne un pointeur vers un des tirs du joueurs, le même que celui passé en paramètre, à condition que celui-ci appartienne bien au joueur. Cette méthode peut etre utilisée pour verifier l'appartenance d'un tir à un joueur (et permet en meme temps un cast du pointeur fourni en entrée).
- update : applique les différentes forces au joueur (acceleration, gravité, répulsions des murs,...), calcule sa nouvelle vitesse, et enfin, le fait se déplacer.
- updateShoot : fait se déplacer chacun des tirs du joueur en fonction de leur vitesse.

29.3 La classe QtP_Gravity.cpp

Le constructeur de cette classe crée un point de gravité aux coordonnées passées en argument, et lui affecte une masse, passée elle aussi en argument. Voici les principales méthodes de cette classe :

- boundingRect : définit la zone de dessin.
- shape : définit la forme (vide ici, pour aucune collision).

– paint : dessine le point de gravité, un simple cercle uni pour le moment, dont la couleur dépend de la masse.

– getGravityAt : retourne la force qui s'applique au joueur passé en argument, qui dépend de la masse des deux corps ainsi que de la distance les séparant.

29.4 La classe QtP_Shoot.cpp

Via le constructeur de cette classe, le tir peut etre créé de deux manières, soit en fournissant une position, un angle et une vitesse, soit une position et une vitesse pour chaque composante (verticale et horizontale). Voici les principales méthodes de cette classe :

– boundingRect : définit la zone de dessin.

– shape : définit la forme.

– move : met à jour le tir en le faisant se déplacer.

29.5 La classe QtP_Wall.cpp

Cette classe n'est rien de plus qu'une implémentation de **QGraphicsItem**, son constructeur prend un polygone en entrée, polygone qui est utilisé seulement dans la détection des collisions, l'affichage s'étant fait dans QtP_Map : :loadMap.

29.6 Gestion des maps via un fichier xml / schéma xsd

Le schéma suivant décrit la structure du fichier xml qui permet la gestion des maps de l'application :

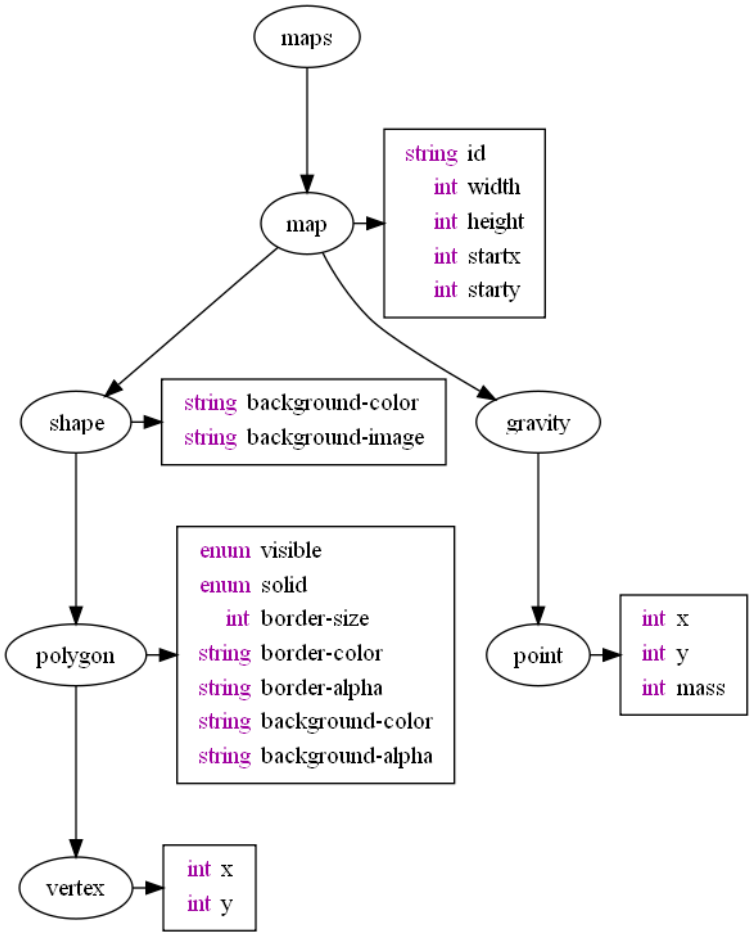


FIG. 21 – Structure du fichier xml décrivant les cartes du jeu.

30 Partie réseau

30.1 Le serveur : classe QtP_ Reseau_serv.cpp

Voici les principales méthodes de cette classe :

- nouvelleConnexion : ajoute un nouveau client à la liste des clients.
- donneesRecues : gère les données reçues. Tout d'abord, on détermine quel client envoie le message (recherche du QTcpSocket du client). Après avoir vérifié que le message a été reçu en entier (en testant la taille), alors, on appelle la méthode envoyerATous afin d'envoyer le paquet qui vient d'être reçu, à tous les clients.

– deconnexionClient : cette méthode détermine quel est le client qui vient de se déconnecter, et le retire de la socket.
– envoyerATous : envoi du paquet préparé à tous les clients connectés au serveur.

– timerEvent : envoi à chaque client de son ID.

30.2 Le serveur : classe QtP_ Reseau_client.cpp

Voici les principales méthodes de cette classe :

– timerEvent : permet l'envoi d'un message au serveur.

– donneesRecues : gère les données reçues de la part du serveur. Tout d'abord, l'ID qui est attribué au client est récupéré, le client est alors créé et ajouté à la map.

– connecte : cette méthode est appelée lorsque la connexion au serveur a réussi.
– deconnecte : cette méthode est appelée lorsque le client est déconnecté du serveur.

– erreurSocket : cette méthode est appelée lorsqu'il y a une erreur (si le serveur a coupé la connexion, ou bien si l'adresse IP est invalide).

Onzième partie

Manuel de déploiement

Objet : L'objectif du Manuel de déploiement est de détailler les étapes nécessaires à l'installation du produit.

Auteurs :

Auteurs	Approbateurs	Validation
Stephanie MALAKIAN Michael REZAC	Zeina BAALBAKI Fouzi BENMAKHLOUF	Laurent TICHIT
Livré le : *12/12/08*	Approuvé le : *12/12/08*	*En attente de validation*

Diffusion :

Diffusion	*Externe*
À :	Laurent TICHIT
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Stephanie MALAKIAN, Michael REZAC.

Historique :

Version du document	Date de modification	Modifications apportées	Auteurs des modifications
V1.0	05/11/08	Création du document	Stephanie MALAKIAN, Michael REZAC
V1.1	19/11/08	Mise à jour	Zeina BAALBAKI, Christophe DEVIN
V1.2	05/12/08	Mise à jour	Fouzi BENMAKHLOUF, Christophe DEVIN
V2.0	12/12/08	Version finale du déploiement	Stéphane GRANGE, Stephanie MALAKIAN

31 Pré-requis d'utilisation

Pour pouvoir installer et utiliser « QTpilot » correctement, il est nécessaire de posséder une version 4.4.x de Qt.

32 Téléchargement de « QTpilot »

Lors de la réception du produit, vous avez reçu un dossier compressé nommé ***QTpilot-Vfinale.tar.qz***. Il vous faut dans un premier temps décompresser ce dossier en utilisant les commandes suivantes dans un terminal (ou invite de commande Windows) :

```
gunzip QTpilot-Vfinale.tar.gz
tar xvf QTpilot-Vfinale.tar.gz
```

Le dossier alors créé contient lui-même quatre répertoires :

- DOC : qui correspond à la documentation du code source de l'application et que vous pouvez visualiser dans un navigateur Web en y ouvrant le fichier nommé index.html,
- QtPilot,
- InterfaceLinux-vfinale,
- InterfaceWin-vfinale.

33 Installation de « QTpilot » sur Windows

Placez-vous dans le dossier QtPilot.
Tapez les commandes suivantes dans l'invite de commandes Windows afin de créer le fichier exécutable QtPilot.exe sur votre machine :

```
qmake -project "QT += xml network"
qmake
make
```

Copiez alors l'exécutable QtPilot.exe dans le dossier InterfaceWin-vfinale. Déplacez-vous dans le dossier InterfaceWin-vfinale, et tapez les commandes suivantes dans l'invite de commandes Windows afin de terminer l'installation de *QTpilot* sur votre machine :

```
qmake -project
qmake
make
```

Un nouveau fichier exécutable InterfaceWin-vfinale.exe s'est créé. Il ne vous reste plus qu'à double-cliquer sur ce fichier pour lancer le jeu.

34 Installation de « QTpilot » sur GNU Linux

Placez-vous dans le répertoire QtPilot.
Si cela n'a pas déjà été fait, tapez les commandes suivantes dans un terminal afin de créer l'exécutable QtPilot sur votre machine :

```
qmake -project "QT += xml network"
qmake
make
```

Copiez alors l'exécutable QtPilot dans le répertoire InterfaceLinux-vfinale. Déplacez-vous dans le répertoire InterfaceLinux-vfinale, et tapez les commandes suivantes dans un terminal afin de terminer l'installation de *QTpilot* sur votre machine :

```
qmake -project
qmake
make
```

Un nouveau fichier exécutable InterfaceLinux-vfinale.exe s'est créé. Il ne vous reste plus qu'à lancer cet exécutable en ligne de commandes pour lancer le jeu :

```
./InterfaceLinux-vfinale
```

Douzième partie

Manuel d'utilisation de « QTpilot »

Objet : L'objectif de ce document est d'expliquer comment faire pour jouer à *QTpilot*.

Auteurs :

Auteurs	Approbateurs	Validation
Zeina BAALBAKI Fouzi BENMAKHLOUF	Christophe DEVIN Michael REZAC	Laurent TICHIT
Livré le : *12/12/08*	Approuvé le : *12/12/08*	*En attente de validation*

Diffusion :

Diffusion	*Externe*
À :	Laurent TICHIT
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Stephanie MALAKIAN, Michael REZAC.

Historique :

Version du document	Date de modification	Modifications apportées	Auteurs des modifications
V1.0	05/11/08	Création du document	Zeina BAALBAKI, Fouzi BENMAKHLOUF
V1.1	05/12/08	Mise à jour	Zeina BAALBAKI, Christophe DEVIN
V2.0	10/12/08	Finalisation du document	Stéphane GRANGE, Stephanie MALAKIAN

35 Faire déplacer son vaisseau dans la carte

Pour faire déplacer son vaisseau dans la carte, il suffit d'utiliser les flèches du clavier.

35.1 Avancer

La flèche du haut  sert à faire avancer le vaisseau. Elle permet également d'accélérer. Plus longtemps la touche est enfoncée, plus la vitesse du vaisseau augmente.

35.2 Tourner

Les flèches gauche  et droite  servent à diriger la tête du vaisseau vers la gauche ou vers la droite.
L'appui sur les flèches de direction droite (ou gauche) provoque un changement de direction du vaisseau : le vaisseau tourne sur lui même. Ainsi, laisser la touche  enfoncée fera faire un tour au vaisseau.
Une fois la direction choisie, appliquer une accéleration (avec la touche haut ) fera avancer le vaisseau dans cette direction.

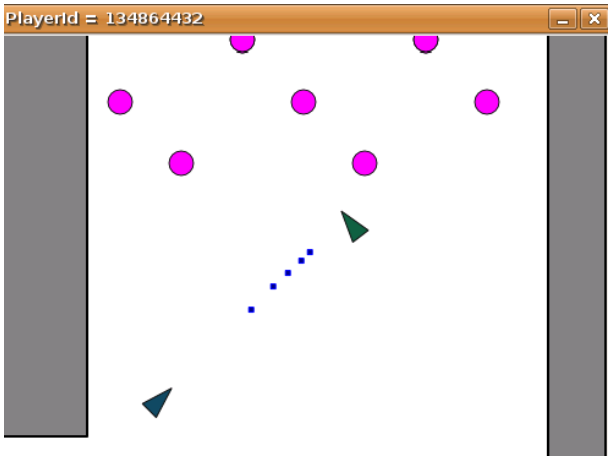
35.3 Reculer et freiner

La flèche du bas  permet de reculer lorsque le vaisseau est à l'arrêt. Si le vaisseau possède une certaine vitesse non nulle, alors le fait de presser la flèche du bas va le freiner (sa vitesse va diminuer peu à peu, puis si la touche est toujours enfoncée, alors le vaisseau va reculer).

35.4 Augmenter sa vitesse

La touche Shift permet d'accroître la vitesse du vaisseau : pour avancer très rapidement, au lieu d'utiliser la touche directionnelle Haut, il suffit de maintenir enfoncée la touche Shift.

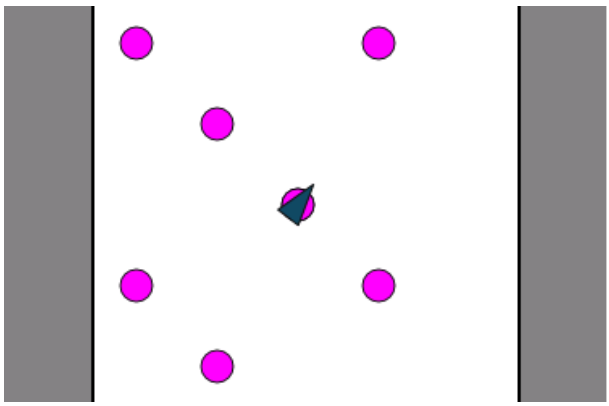
36 Tirer



La touche ESPACE permet au vaisseau de tirer sur ses adversaires.

37 Éviter les zones de plein ou les points de gravité...

D'autre part, la carte présente un certain nombre de points de gravité : si le vaisseau s'approche de ceux-ci, alors soit il sera repoussé (si le point de gravité est répulsif), soit il sera attiré et « bloqué » à l'intérieur d'un des points (si celui-ci est attractif), comme l'illustre la capture d'écran ci-contre : pour en sortir, le joueur doit utiliser la touche SHIFT.



Treizième partie

Dossier d'Évolution

Objet : L'objectif du dossier d'évolution est de détailler les évolutions possibles du jeu *QTpilot*.

Auteurs :

Auteurs	**Approbateurs**	**Validation**
Christophe DEVIN Michael REZAC	Zeina BAALBAKI Stéphane GRANGE	Stephanie MALAKIAN
Livré le : *12/12/08*	Approuvé le : *12/12/08*	Validé le : *12/12/08*

Diffusion :

Diffusion	*Interne*
À :	Stephanie MALAKIAN (chef de projet)
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Michael REZAC.

Historique :

Version du document	**Date de modification**	**Modifications apportées**	**Auteurs des modifications**
V1.0	05/11/08	Création du document	Fouzi BENMAKHLOUF, Michael REZAC
V2.0	12/11/08	Finalisation du document	Stephanie MALAKIAN, Michael REZAC

Voici les différentes options du jeu *QTpilot*, ainsi que leur durée et coût estimés, qui pourront être implémentées par la suite, dans le cadre d'une évolution future du jeu.
N. B. : Les évolutions présentées ci-dessous suivent une certaine logique de chronologie.

38 Suppression de la map des vaisseaux des joueurs déconnectés

☞ **Temps estimé** : 10 heures - **Coût estimé** : 1000€

Lorsqu'un joueur se déconnecte, son vaisseau disparaîtrait de la carte.

Techniquement : En utilisant la méthode `deconnexionClient()` du serveur, il est possible de connaître le moment où un joueur se déconnecte. À ce moment, le serveur doit envoyer une information à chacun des joueurs pour lui dire de supprimer le joueur de la carte (et pourquoi pas afficher un message). En pratique, cela pose un probleme majeur, le flux réseau n'etant pas constant, des paquets sont perdus régulierement, ce qui implique qu'il faudrait implémenter un systeme de vérification des paquets reçus par les clients.

39 Introduction des points de vie pour chaque joueur

☞ **Temps estimé** : 8 heures - **Coût estimé** : 700€

Chaque joueur possèderait un nombre fixé de points de vie en début de partie. À chaque collision soit par une zone de plein, soit par un tir adverse (une pondération serait introduite), alors le joueur perdrait un certain nombre de points de vie. Dès lors que ce nombre devient égal à zéro, alors la partie se termine pour ce joueur.

Techniquement : Cela consisterait à ajouter un membre `private` aux joueurs, et lui affecter une valeur de départ. Les collisions étant deja gérées, on peut facilement intégrer dans la méthode d'update, la gestion des dégâts en cas de tirs ou de collisions avec un mur. La partie la plus compliquée sera de gérer la destruction d'un joueur, on pourrait envisager de bloquer les commandes clavier et de couper le moteur un certain temps pour un joueur ayant perdu toute sa vie. Au bout d'un certain temps, à déterminer, la vie du joueur remonterait alors.

40 Possibilité de tirs arrières et sur les côtés

☞ **Temps estimé** : 10 heures - **Coût estimé** : 1000€

Chaque joueur aurait la possibilité de tirer sur ses adversaires soit par des tirs arrières, soit par des tirs sur les côtés.

Techniquement : Il serait nécessaire de réimplémenter complètement le constructeur de `QtP_Shoot`

pour lui permettre de créer des tirs multi-directionnels. Il faudrait également modifier la gestion des touches dans QtP_Map pour permettre d'assigner une touche à chaque type de tir.

41 Création d'une mini-map « radar »

☞ **Temps estimé** : 8 heures - **Coût estimé** : 700€

Pour le mode de jeu Grand-Prix, chaque joueur pourrait connaître la position de ses adversaires par rapport à sa propre position grâce à une mini-map « radar », qui correspondrait en fait à une petite carte (qui serait une reproduction réduite de la carte du jeu) située en bas dans un des coins de l'écran, qui indiquerait en vert la position en temps réel du joueur, et en rouge toutes les positions des joueurs adverses.

Techniquement : Il faudrait ajouter un *overlay* (une image de premier plan) sur la carte, qui serait mise à jour en même temps que la map principale, mais donc les objets ne seraientt représentés que par de simples points, et sans dessins de la carte.

42 Introduction des objets

☞ **Temps estimé** : 12 heures - **Coût estimé** : 1500€

Divers objets (qui correspondraient en fait à des images se déplaçant suivant une certaine trajectoire à travers la carte) pourraent être introduits, tels que :

- des objets bonus : à chaque contact du vaisseau d'un joueur avec un bonus, ce-dernier disparaîtrait de la carte, et le joueur dont le vaisseau est concerné gagnerait soit des points de vie, soit sa vitesse serait fortement augmentée durant un petit laps de temps.

- des objets malus : idem que pour les bonus, sauf qu'en cas de contact, le joueur concerné perdrait des points de vie, ou verrait sa vitesse très fortement ralentie.

- des boucliers : à chaque passage au-dessus d'un tel objet, le joueur verrait son vaisseau doté d'une invincibilité durant un petit laps de temps.

Techniquement : Cela consisterait en la création d'une nouvelle classe pour représenter les objets et leurs effets. À la detection d'une collision avec un objet, récuperer son effet et l'appliquer au joueur, qui doit donc avoir été modifié pour se « souvenir » quel effet est en train de s'appliquer. Il doit également se souvenir des objets qu'il est en train de porter, afin de pouvoir influer sur son poids.

43 Création d'un harpon pour chaque vaisseau

☞ **Temps estimé** : 6 heures - **Coût estimé** : 500€

Chaque vaisseau aurait la possibilité d'attraper certains objets spécifiques par un harpon, qu'il n'aurait le droit d'utiliser qu'un certain nombre de fois par partie.

Techniquement : La gestion des objets doit être présente. Cela consisterait alors en la modification de la gestion des touches pour ajouter la possibilité de déclencher un « champ de gravité » de forte puissance, mais de rayon très faible, et n'affectant que les objets. Le code des points de gravité etant assez similaire, la complexité du travail s'en trouve simplifiée.

44 Choix des maps

☞ **Temps estimé** : 8 heures - **Coût estimé** : 700€

Lorsqu'un joueur créé une partie, il a la possibilité de choisir sa map parmi celles proposées. De plus, il aurait la possibilité de créer ses propres maps (en définissant autant de zones de plein ou de points de gravité souhaité).
Il faudrait alors, que lorsqu'un joueur se connecte à une partie déjà existante, le serveur envoie la map utilisée pour cette partie au client sur lequel joue l'utilisateur afin que la carte soit la même pour tous.

Techniquement : Il faudrait intégrer la lecture du fichier xml a l'interface pour afficher la liste des maps disponibles, pour pouvoir les passer en argument à `loadMap`. Il serait ensuite nécessaire de fournir à tout nouveau joueur le nom de la map à charger, et de vérifier qu'il l'a bien reçu.

45 Création de deux modes de jeu en réseau supplémentaires

☞ **Temps estimé** : 25 heures - **Coût estimé** : 2000€

- **Mode Death-Match** : ce jeu se jouerait soit individuellement, soit en équipe. Le but serait de tuer un maximum d'ennemis.

- **Mode Drapeau** : dans ce cas, le jeu se jouerait en équipe (chaque joueur au moment de se connecter choisit dans quelle équipe il souhaite faire partie). Le but serait pour chaque équipe, d'attraper un ou plusieurs drapeaux qui se trouveraient à un certain endroit de la carte –semée d'embûches–, et de le ramener dans sa « goal-zone » (chaque équipe possède sa goal-zone sur la carte). La partie serait chronométrée, et à chaque fin de partie, l'équipe ayant le plus de drapeaux dans sa goal-zone gagne.

Techniquement : Cela consisterait en la création d'une nouvelle methode et de membres pour la classe QtP_Map qui vont gérer le déroulement de la partie, compter les points pour chaque equipe. L'implémentation correcte ne peut être déterminée qu'au moment de la création, les autres fonctions affectant fortement le déroulement de la partie (comme la gestion des tirs qui peut être desactivée, ou les points de vie qui sont nécessaires pour le mode deathmatch).

46 Création de cartes par l'utilisateur directement via le menu

☞ **Temps estimé** : 50 heures - **Coût estimé** : 10000€

L'idée serait de creér un editeur WYSIWYG (What You See Is What You Get) permettant de dessiner les elements du décor, ainsi que de placer des points de gravité et de départ des joueurs. Une toute nouvelle classe devrait être créée, qui gérerait les intéractions claviers/souris sur la zone d'edition. Au final, il s'agirait plus ou moins de coder un mini logiciel de dessin vectoriel se limitant à des tracés de polygones.

47 Choix de la configuration des touches ou utilisation d'une manette

☞ **Temps estimé** : 6 heures - **Coût estimé** : 500€

Chaque utilisateur aurait la possibilité de configurer ses propres choix de touches pour telle ou telle action

Techniquement : Il faudrait ajouter des variables contenant les valeurs de chacune des touches, et les utiliser plutôt que d'utiliser des constantes comme c'est le cas en ce moment. Une sauvegarde de ces réglages pourrait se faire dans un simple fichier texte ! ;

48 Création du mode de jeu « Entraînement »contre l'ordinateur

☞ **Temps estimé** : 30 heures - **Coût estimé** : 5000€

Cela correspondrait à la creation d'un intelligence artificielle qui serait capable de contrôler le vaisseau pour lui faire atteindre son objectif (finir premier et/ou détruire ses adversaires).

49 Crash en cas de collision

☞ **Temps estimé** : 5 heures - **Coût estimé** : 500€

Les points de vie doivent être implémentés. De plus, la gestion de dégâts doit etre implémentée au préalable. Lors d'une collsion avec un mur, connaissant la vitesse d'un joueur, on pourrait lui appliquer un certain nombre de points de dégâts, il suffirait alors de donner une valeur limite qui donnerait autant de dégâts que de points de vie, ce qui détruirait le vaisseau.

Quatorzième partie

Bilan final du projet

Au terme de ces 7 semaines de projet, une application permettant un jeu de base et satisfaisant notre client, a été réalisée.

À ce jour, le jeu *QTpilot* offre :
- la possibilité de jouer en réseau multi-joueurs sur des machines distantes,
- un mode de jeu Grand-Prix simplifié,
- les tirs entre vaisseaux sont effectifs,
- les zones de pleins sont intégrées à chaque carte,
- les collisions (que ce soit en ce qui concerne les vaisseaux contre les zones de plein, ou les tirs sur les vaisseaux) sont effectives.

Tout au long du travail d'implémentation de l'application, nous nous sommes efforcés, tout en restant en accord avec les besoins du client, à implémenter les fonctionnalités dites « de base » (telles que le réseau par exemple), en insistant sur le fait qu'elles se devaient d'être fonctionnelles, avant d'envisager l'implémentation de diverses options plus sophistiquées (comme par exemple l'introduction des objets).

Cependant, toutes les options et fonctionnalités supplémentaires définies dans le Cahier des Charges et qui n'ont pu être implémentées (faute de temps), ont été détaillées avec soin dans le Dossier d'Évolution du projet, de façon à ce que, lors d'une évolution future du jeu, de telles fonctionnalités puissent être implémentées de manière rapide et efficace.

Quinzième partie
Annexes

50 Tableau de Bord

Objet : L'objectif de cette section est de présenter le tableau de bord des courriels émis et reçus entre le client GAME-IN et notre société e-nov data, ainsi que les différentes livraisons concernant le projet *QTpilot*.

Auteurs :

Auteurs	Approbateurs	Validation
Zeina BAALBAKI Christophe DEVIN	Stéphane GRANGE Stephanie MALAKIAN	Laurent TICHIT
Livré le : *12/12/08*	Approuvé le : *12/12/08*	*En attente de validation*

Diffusion :

Diffusion	*Externe*
À :	Laurent TICHIT
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Stephanie MALAKIAN, Michael REZAC.

Historique :

Version du document	Date de modification	Modifications apportées	Auteurs des modifications
V1.0	27/11/08	Création du document	Zeina BAALBAKI, Christophe DEVIN
V1.1	12/12/08	Finalisation du document	Zeina BAALBAKI, Christophe DEVIN

50.1 Tableau de bord des courriels émis

Voici le tableau de bord récapitulant les divers courriers électroniques émis dans le cadre du projet *QTpilot* :

N°	Réf.	Date	Émetteur	Destinataire	Objet
1	CR1	22/10/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	Compte rendu externe n°1
2	LIV0	26/10/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	Livraison initiale (n°0)
3	RDV1	02/11/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	- Demande de rendez-vous pour le 05/11/08 - Ordre du jour
4	DEM1	06/11/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	Proposition de rendez-vous pour le jour même pour faire une démonstration du réseau au client
5	LIV1	08/11/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	Livraison n°1
6	CR2	12/11/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	Compte rendu externe n°2
7	LIV2	19/11/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	Livraison n°2
8	RDV2	28/11/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	- Demande de rendez-vous pour le 01/12/08 - Ordre du jour
9	CR3	01/12/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	Compte rendu externe n°5
10	LIV3	05/12/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	Livraison n°3

N°	Réf.	Date	Émetteur	Destinataire	Objet
11	RDV3	05/12/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	- Demande de rendez-vous pour le 08/12/08 - Ordre du jour
12	CR4	09/12/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	Compte rendu externe n°6
13	LIV4	12/12/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	Livraison finale (n°4)

TAB. 6 – Tableau de bord des courriels émis.

50.2 Tableau de bord des courriels reçus

Voici le tableau de bord récapitulant les divers courriers électroniques reçus dans le cadre du projet *QTpilot* :

N°	Réf.	Date	Émetteur	Destinataire	Objet
1	CR1	24/10/08	*GAME-IN* <tichit@lidil.univ-mrs.fr>	*e-nov data*	Réponse CR1 : correction de l'en-tête
2	LIV0	29/10/08	*GAME-IN* <tichit@lidil.univ-mrs.fr>	*e-nov data*	Réponse LIV0 : remarques concernant la disposition des documents
3	RDV1	03/11/08	*GAME-IN* <tichit@lidil.univ-mrs.fr>	*e-nov data*	Réponse RDV1 : confirmation du rendez-vous
4	DEM1	07/11/08	*GAME-IN* <tichit@lidil.univ-mrs.fr>	*e-nov data*	Réponse DEM1 : confirmation de sa présence lors de la démonstration en salle de TP.
5	LIV1	11/11/08	*GAME-IN* <tichit@lidil.univ-mrs.fr>	*e-nov data*	Réponse LIV1 : remarques concernant le découpage en lots/livrables et les outils utilisés.
6	CR2	12/11/08	*GAME-IN* <tichit@lidil.univ-mrs.fr>	*e-nov data*	Réponse CR2 : validation du compte rendu externe n°3.
7	LIV2	20/11/08	*GAME-IN* <tichit@lidil.univ-mrs.fr>	*e-nov data*	Réponse LIV2 : accusé de réception de la livraison n°2.
8	RDV2	29/11/08	*GAME-IN* <tichit@lidil.univ-mrs.fr>	*e-nov data*	Réponse RDV2 : confirmation du rendez-vous.

N°	Réf.	Date	Émetteur	Destinataire	Objet
9	LIV3	08/12/08	*GAME-IN* <tichit@lidil.univ-mrs.fr>	*e-nov data*	Réponse LIV3 : accusé de réception de la livraison n°3.
10	RDV3	08/12/08	*GAME-IN* <tichit@lidil.univ-mrs.fr>	*e-nov data*	Réponse RDV3 : confirmation du rendez-vous.
11	CR4	10/12/08	*GAME-IN* <tichit@lidil.univ-mrs.fr>	*e-nov data*	Réponse CR4 : validation du compte rendu externe n°3.

TAB. 7 – Tableau de bord des courriels reçus.

50.3 Gestion des livraisons

Voici le tableau de bord récapitulant les livraisons des diverses versions de *QTpilot* :

N°	Réf.	Date	Émetteur	Destinataire	Objet
1	LIV0	26/10/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	Livraison n°0 (initiale)
2	LIV1	08/11/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	Livraison n°1
3	LIV2	19/11/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	Livraison n°2
4	LIV3	05/12/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	Livraison n°3
5	LIV4	12/12/08	*e-nov data* <malakian.stephanie@gmail.com>	*GAME-IN* <tichit@lidil.univ-mrs.fr>	Livraison finale (n°4)

TAB. 8 – Gestion des livraisons.

51 Tableau de bord des réunions internes

N°	Réf.	Date	Émetteur	Destinataire	Objet
1	INT1	22/10/08	*Chef de projet e-nov data* <malakian.stephanie@gmail.com>	*équipe e-nov data* <zeinabaalbaki@gmail.com> <fouzi.benmakhlouf@gmail.com> <devinchristophe2@gmail.com> <stephane.grange@dil.univ-mrs.fr> <rezac.michael@tsundere.fr>	Réunion interne n°1
2	INT2	30/10/08	*Chef de projet e-nov data* <malakian.stephanie@gmail.com>	*équipe e-nov data* <zeinabaalbaki@gmail.com> <fouzi.benmakhlouf@gmail.com> <devinchristophe2@gmail.com> <stephane.grange@dil.univ-mrs.fr> <rezac.michael@tsundere.fr>	Réunion interne n°2
3	INT3	19/11/08	*Chef de projet e-nov data* <malakian.stephanie@gmail.com>	*équipe e-nov data* <zeinabaalbaki@gmail.com> <fouzi.benmakhlouf@gmail.com> <devinchristophe2@gmail.com> <stephane.grange@dil.univ-mrs.fr> <rezac.michael@tsundere.fr>	Réunion interne n°3
4	INT4	26/11/08	*Chef de projet e-nov data* <malakian.stephanie@gmail.com>	*équipe e-nov data* <zeinabaalbaki@gmail.com> <fouzi.benmakhlouf@gmail.com> <devinchristophe2@gmail.com> <stephane.grange@dil.univ-mrs.fr> <rezac.michael@tsundere.fr>	Réunion interne n°4

TAB. 9 – Tableau de bord des réunions internes.

52 Comptes rendus des réunions externes

Objet : L'objectif de ce document est de rendre compte des différentes réunions que nous avons eu avec le client.

Auteurs :

Auteurs	**Approbateurs**	**Validation**
Stephanie MALAKIAN Zeina BAALBAKI	Christophe DEVIN Stéphane GRANGE	Laurent TICHIT
Livré le : *12/12/08*	Approuvé le : *10/12/08*	*En attente de validation*

Diffusion :

Diffusion	*Externe*
À :	Laurent TICHIT
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Stephanie MALAKIAN, Michael REZAC.

Historique :

Version du document	Date de modification	Modifications apportées	Auteurs
V1.0	22/10/08	Création du document	Zeina BAALBAKI, Stephanie MALAKIAN
V1.1	23/10/08	Ajout du compte rendu de la réunion externe n°1 du 22/10/08	Zeina BAALBAKI, Stephanie MALAKIAN
V1.2	07/11/08	Ajout du compte rendu de la réunion externe n°2 du 05/11/08	Stéphane GRANGE, Stephanie MALAKIAN
V1.3	18/11/08	Ajout du compte rendu de la réunion externe n°3 du 12/11/08	Zeina BAALBAKI, Michael REZAC
V1.4	19/11/08	Ajout du compte rendu de la réunion externe n°4 du 19/11/08	Christophe DEVIN, Stephanie MALAKIAN
V1.5	05/12/08	Ajout du compte rendu de la réunion externe n°5 du 01/12/08	Christophe DEVIN, Stephanie MALAKIAN
V2.0	10/12/08	Ajout du compte rendu de la réunion externe n°6 du 08/12/08	Stéphane GRANGE, Michael REZAC

52.1 Compte rendu de la réunion externe n°1 du 22/10/08

Cette première réunion en externe fut l'occasion d'établir une première approche en ce qui concerne les besoins du client. Voici donc les questions que nous avons pu lui poser afin d'obtenir des précisions concernant ses attentes :

▷ **Comment se déroule le jeu ?**

QTpilot est un jeu multi-joueurs. Chaque joueur appartient à une équipe et pilote un vaisseau à travers une carte.
Des objets peuvent être capturés, et ainsi faire gagner des points (augmentation du nombre de tirs, after-burners (c'est-à-dire un gain de puissance pendant un petit laps de temps), cloaking devices (qui permet de rendre invisible) etc...).
De plus, un certain nombre de paramètres peuvent intervenir au cours du jeu (densité de l'atmosphère, gravité etc...).
Trois modes de jeu sont possibles : le mode « *capturer le drapeau* » , le mode « *death-match* » et le mode « *grand prix* ».

▷ **A quoi correspondent ces trois modes de jeu ?**

Le mode « *capturer le drapeau* » consiste à récupérer le drapeau au centre (que l'on capture avec le harpon) et que l'on doit ramener dans la goal-zone, sachant que ce drapeau a une certaine masse (ce qui va donc nécessiter plus ou moins de puissance pour le ramener dans la goal-zone).
Le mode « *death-match* » correspond à l'affrontement de deux équipes, chacune essayant de tuer un maximum d'adversaires de l'autre équipe.
Le mode « *grand prix* » est une course (circuit ou bien en 8) avec les autres joueurs. Dans ce cas, les tirs ne seront pas activés.

▷ **En ce qui concerne l'aspect visuel du jeu, préfèrez-vous qu'il y ait une vue sur le côté ou vue d'en haut ?**

La vue d'en haut serait plus adaptée.

▷ **Dans ce cas, comment intervient la gravité ?**

Deux possibilités : la gravité centrale qui correspond à l'attirance vers un point de la carte, et la gravité axiale qui correspond à l'attirance vers un des bords de la carte.

▷ **Que se passe-t-il si l'utilisateur ne presse aucune touche pendant un certain moment ?**

S'il n'y a pas de gravité, alors le vaisseau poursuit sa trajectoire.
S'il y a de la gravité, la vitesse du vaisseau va diminuer petit à petit, puis lorsqu'il s'arrête, le vaisseau sera attiré par le centre (ou le bord) de la gravité. Par ailleurs, si le vaisseau avançait dans le même sens que la gravité, alors il s'en rapprochera plus vite.

▷ **Dans quelles circonstances intervient la masse des différents objets ?**

Chaque objet (bouclier anti-choc, anti-canon, drapeau etc...) a une masse. Plus le vaisseau capture d'objets, plus sa propre masse augmente, ce qui fait qu'il aura besoin de plus de puissance pour pouvoir se déplacer (et donc il aura besoin de plus de carburant).

▷ **Comment évolue la vitesse du vaisseau ?**

Tant que le joueur appuiera sur la touche, la vitesse du vaisseau augmentera à l'infini.
De plus, plus le vaisseau va vite, plus la résistance de l'air sera forte.

▷ **Que se passe-t-il si un vaisseau rencontre une zone de « plein » ?**

Cela peut dépendre de la vitesse du vaisseau : si celle-ci est très élevée, alors le vaisseau s'écrase.
Si au contraire, la vitesse du vaisseau est faible, alors le vaisseau perdra des points de vie, mais il ne s'écrasera pas.
De plus, on pourrait aussi envisager que le vaisseau puisse faire des rebonds lorsqu'il entre en contact avec une zone de plein et que sa vitesse n'est pas trop élevée.

52.2 Compte rendu de la réunion externe n°2 du 05/11/08

Cette seconde réunion avec le client s'est déroulée le mercredi 05/11/08. Voici l'ordre du jour :

- ▷ Présentation de la première version de la maquette,
- ▷ Présentation du premier diagramme des cas d'utilisations,
- ▷ Présentation d'une version initiale « de base » du jeu,
- ▷ Détails du premier livrable et mise au point de la première livraison.

▷ Présentation de la première version de la maquette

Cette réunion nous a tout d'abord permis de présenter au client une première version de la maquette de l'application. D'un point de vue général, le client en est satisfait. Il nous a cependant faire remarqué concernant les boutons « Quitter » qui permettent de quitter l'application : ils étaient présents sur chaque écran. Selon le client, il est préférable de ne proposer ce choix que sur la page d'accueil du jeu seulement. Ainsi, si l'utilisateur souhaite quitter l'aaplication alors qu'il se trouve sur un des autres écrans, il devra cliquer sur le bouton de retour jusqu'à ce qu'il se retrouve sur l'écran d'accueil, où il aura la possibilité de quitter l'application.

▷ Présentation du premier diagramme des cas d'utilisations

Nous avons également pu présenter au client un premier diagramme des cas d'utilisations.
Le client et nous avons alors pu apporter une précision concernant ses attentes : le cas d'utlisation *Options* doit correspondre au cas où l'utilisateur souhaite choisir ses préférences ou bien connaître les commandes du jeu :

- Choisir ses préférences : cela correpond aux préférences du joueur (comme modifier son pseudo par exemple, ou bien encore connaître les meilleurs scores)
- Connaître les commandes du jeu : si le joueur souhaite savoir quelle touche est associée à quelle action.

▷ Présentation d'une version initiale « de base » du jeu

Cette réunion fut également l'occasion de présenter au client une première version « en l'état » de l'application (c'est-à-dire une partie du lot n°1) : cette version consiste simplement en un vaisseau qui se déplace sur une carte. Le vaisseau peut avancer, reculer, freiner et accélerer.
La gestion du réseau n'étant pas encore au point, nous avons convenu d'éventuellement prendre un rendez-vous ultérieur avec le client afin de lui en faire une rapide démonstration une fois que le réseau à 2 joueurs sera mis en place.

▷ **Détails du premier livrable et mise au point de la première livraison**

Enfin, nous avons fixé, avec le client, le contenu précis de l'itération n°1 (en cours, et qui se termine le vendredi 05/11/08) ainsi que le contenu de la livraison n°1, qui est prévue pour le vendredi 07/11/08. Le premier livrable est constitué des lots n°1 et 2, comme prévu, et devront être livré au vendredi 07/11/08.

52.3 Compte rendu de la réunion externe n°3 du 12/11/08

Cette troisième réunion avec le client s'est déroulée le mercredi 12/11/08. Son ordre du jour a consisté en les points suivants :

- ▷ Tout d'abord, nous avons fait le point avec le client en ce qui concerne l'itération en cours,
- ▷ Enfin, nous avons convenu d'un changement du contenu de la prochaine livraison.

▷ Mise au point de l'itération en cours (itération n°2)

Lors de cette réunion avec le client, nous avons pu faire un état d'avancement concernant l'itération en cours (itération n°2) : cela a consisté principalement à expliquer au client les diverses caractéristiques déjà intégrées à l'application et de présenter celles que nous avons l'intention de développer pour la prochaine livraison, prévue pour le vendredi 14/11/08.

▷ Modification du contenu de la prochaine livraison

Étant donné que les diverses caractéristiques correspondant à l'itération n°3 (et devant être livrés pour le 24/11/08) comme la vitesse et l'accélération des vaisseaux, ont déjà été codées, intégrées à l'application et sont fonctionnelles, nous avons décidé d'un commun accord avec le client d'inverser (en grande partie) les contenus des livraions n°2 et 3 : ainsi, l'implémentation des zones de plein dans la carte et la gestion des collisions sur ces zones de plein sont reportées à l'itération n°3 (qui débutera le lundi 17/11/08), tandis que la gestion des tirs est maintenue pour l'itération actuelle, à laquelle s'ajoute la gestion de la densité de l'atmosphère ainsi que de la gravité centrale.

52.4 Compte rendu de la réunion externe n°4 du 19/11/08

Cette réunion avec le client s'est déroulée le mercredi 12/11/08. Voici les deux points qui y ont été abordés :

- ▷ Nous avons présenté au client la toute dernière version de l'application,
- ▷ Puis nous avons fait le point concernant la prochaine itération.

▷ Présentation de la toute dernière version de l'application

Cette réunion fut l'occasion de faire une démonstration au client de la toute dernière version de l'application, qui correspond à l'aboutissement de l'itération n°2 (qui a été terminée avec quelques jours de retard) : les tirs ainsi que la gravité centrale ont été implémentés, et sont fonctionnels. À la demande du client, la démonstration s'est faite sur deux ordinateurs distants (serveur + client) et suite à cette démonstration, le client fut satisfait de l'avancement du développement de l'application.

▷ Mise au point de la prochaine itération : itération n°3

Nous avons également pu, avec l'avis du client, aborder le sujet de la prochaine itération : initialement prévue de commencer le lundi 17/11/08 (et étant donné du retard accumulé lors de l'itération n°2), nous avons convenu ensemble que la prochaine livraison (livraison n°3) serait reportée du vendredi 21/11/08 au mercredi 26/11/08 (qui correspondra également à la date de fin de l'itération n°3).
Par ailleurs, le client nous a questionné en ce qui concerne le contenu de la prochaine livraison, celle-ci contiendra essentiellement la version du jeu présentant les zones de plein, et permettant la gestion des collisions (collision contre ces zones / collision lors des tirs).

52.5 Compte rendu de la réunion externe n°5 du 01/12/08

Cette cinquième réunion avec le client a eu lieu durant l'itération n°3, consacrée principalement à l'élaboration des zones de plein et la gestion des collisions. En voici l'ordre du jour :

- ▷ Tout d'abord, nous avons fait la démonstration en l'état de l'application,
- ▷ Puis enfin, nous avons fait un point concernant la prochaine livraison.

▷ Démonstration des zones de plein et des collisions

L'objectif principal de cette réunion fut de présenter au client la toute dernière version de l'application qui implémente des zones de plein sur la carte et la gestion des collisions.
Une zone de plein correspond en fait à un polygone, que l'on peut paramétrer (nombre de côtés, couleur de bord, couleur de contenu, taille) via un fichier xml, gérer par un schéma xsd.
Les collisions concernent les vaisseaux (vaisseau qui entre en collision avec ces zones de plein), mais aussi les tirs (tirs qui sont « stoppés » par ces zones de plein).

Le client a testé l'application, et nous a posé la question de l'angle de collision, chose que nous avons choisi de ne pas traiter. Globalement, il en est satisfait.

▷ Mise au point concernant la prochaine livraison.

L'itération courante (n°3) consiste en plus à modifier notre codage pour le réseau afin de permettre un jeu multi-joueurs (et non plus à deux joueurs seulement, comme ce fut le cas jusqu'à présent). Le codage des collisions a entraîné des modifications au niveau du réseau, que nous n'avons pas encore terminé de mettre à jour. De ce fait, nous n'avons pas pu faire tester au client les collisions / zones de plein dans le cas où il y a plusieurs joueurs en réseau.

Nous avons alors convenu avec le client que le réseau multi-joueurs ainsi que les collisions concernant un jeu en réseau seraient fonctionnels et intégrés à l'application pour la prochaine livraison (livraison n°3), que nous avons fixé avec le client à vendredi 05/12/08.

52.6 Compte rendu de la réunion externe n°6 du 08/12/08

Cette sixième et dernière réunion avec le client a eu lieu durant l'itération n°4, consacrée principalement au débuggage de l'application ainsi qu'à la finalisation de toute la documentation l'accompagnant. En voici l'ordre du jour :

- ▷ Tout d'abord, nous avons fait la démonstration en l'état de l'application,
- ▷ Puis enfin, nous avons fait un point concernant la livraison finale.

▷ Démonstration du réseau en multi-joueurs

L'itération en cours (n°4) est essentiellement consacrée au débuggage de l'application, notamment en ce qui concerne le réseau multi-joueurs. En effet : celui-ci est fonctionnel, mais présente quelques défauts. Cette réunion fut donc pour nous l'occasion d'en faire la démonstration avec le client : nous avons lancé l'application sur quatre machines séparées, le client a alors pu tester le réseau en multi-joueurs : il est presque entièrement fonctionnel (un décalage apparaît entre les différents joueurs au but d'un certain temps). Nous avons alors suggéré au client de limiter le nombre de connexions (peut-être éventuellement à trois), il est d'accord.

▷ Mise au point concernant la livraison finale

L'itération courante (n°4) étant la dernière du projet, nous avons également pu au cours de la réunion faire le point concernant la livraison finale du produit : nous avons récapitulé avec le client les différentes options implémentées dans la dernière version de *QTpilot*, et aussi quelles options nous n'avons pas implémentées, en rappelant que tout ceci était détaillé dans la documentation de référence de *QTpilot*, qui sera également livrée lors de la livraison finale, qui aura lieu comme prévu, vendredi 12/12/08 avant 17h.

53 Comptes rendus des réunions internes

Objet : L'objectif de ce document est de rendre compte des différentes réunions que nous avons eu en interne avec l'ensemble des membres de l'équipe.

Auteurs :

Auteurs	**Approbateurs**	**Validation**
Zeina BAALBAKI Fouzi BENMAKHLOUF	Michael REZAC Christophe DEVIN	Stephanie MALAKIAN
Livré le : *12/12/08*	Approuvé le : *05/12/08*	Validé le : *12/12/08*

Diffusion :

Diffusion	*Interne*
À :	Stephanie MALAKIAN (chef de projet)
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Michael REZAC.

Historique :

Version du document	Date de modification	Modifications apportées	Auteurs
V1.0	25/10/08	Création du document	Zeina BAALBAKI, Stephanie MALAKIAN
V1.1	25/10/08	Ajout du compte rendu de la réunion interne n°1 du 22/10/08	Zeina BAALBAKI, Stephanie MALAKIAN
V1.2	03/11/08	Ajout du compte rendu de la réunion interne n°2 du 30/10/08	Christophe DEVIN, Stephanie MALAKIAN
V1.3	19/11/08	Ajout du compte rendu de la réunion interne n°3 du 19/11/08	Christophe DEVIN, Stephanie MALAKIAN
V1.4	05/12/08	Ajout du compte rendu de la réunion interne n°4 du 26/11/08	Zeina BAALBAKI, Fouzi BENMAKHLOUF

53.1 Compte rendu de la réunion interne n°1 du 22/10/08

Cette première réunion en interne a eu lieu le 22/10/08, juste après le premier entretien avec le client, et elle a réuni l'ensemble des membres de *e-nov data*.
L'ordre du jour a consisté en quatre points principaux :

- ▷ Le premier fut la découverte du projet, de ses enjeux ainsi que des besoins du client,
- ▷ Le second fut de lister les principales tâches constituant le projet,
- ▷ Ensuite, l'organisation de l'équipe fut abordée,
- ▷ Et enfin, la phase de lancement du projet a pu être démarrée.

▶ Découverte du projet et de ses enjeux

Suite à notre première réunion en externe avec le client, nous avons pu découvrir quels étaient les enjeux du projet, ainsi que ses attentes concernant le produit que nous devons réaliser.

▶ Liste des tâches

Nous avons dans un premier temps tous ensemble listé l'ensemble des tâches intervenant dans le projet, dans un ordre plus ou moins chronologique et par niveau de difficulté de réalisation décroissant (du plus difficile au moins difficile), afin d'avoir un aperçu du projet dans son ensemble et surtout par rapport aux contraintes de temps.

▶ Organisation de l'équipe

Les premiers binômes ont ensuite été constitués par le chef de projet, et affectés aux tâches initiales, à savoir :

- la création du planning prévisionnel,
- l'organisation générale du rapport,
- la découverte des outils techniques (C++/Qt pour le réseau et le graphisme).

D'autre part, l'ensemble des membres de l'équipe a décidé d'une fréquence des réunions internes fixée à une semaine (avant le passage à une nouvelle itération), dans le but de s'assurer du bon avancement de chaque tâche ainsi que pour prévenir / ou rattrapper d'éventuels retards.

▶ Démarrage de la phase de lancement

Enfin, la phase de lancement du projet a été fixée : elle consiste principalement en la création du planning prévisionnel et la rédaction du Plan d'Assurance Qualité. Dès lors, la phase de lancement a pu être démarrée.

53.2 Compte rendu de la réunion interne n°2 du 30/10/08

Cette seconde réunion en interne a eu lieu le 30/10/08 et a réuni l'ensemble des membres de *e-nov data*. L'ordre du jour a consisté en les points suivants :

- ▷ Faire un point quant à l'avancement des différentes tâches prévues pour cette semaine,
- ▷ Mettre en commun les différents résultats et avancements de chaque binôme,
- ▷ Préparer la prochaine réunion avec le client.

▶ Mise au point sur l'état d'avancement du projet

Après cette nouvelle semaine, nous avons pu faire le point concernant l'état d'avancement des premières tâches constituant le projet. Globalement, la chronologie des tâches a pour l'instant bien été respectée. Par contre, un remaniement des binômes a été effectué, dans le but de ré-équilibrer le travail de chaque membre de l'équipe. D'autre part, nous avons également pu lister les prochaines tâches importantes qu'il fallait traiter en priorité : il s'agit principalement de la suite de l'analyse fonctionnelle (diagrammes de séquences) et de la gestion du réseau en C++/Qt, dont la formation a dû être renforcée et étendue à tous les membres de l'équipe afin de garantir que la première itération puisse proposer au client un jeu « de base » permettant de jouer à deux en réseau.

▶ Mise en commun des premiers résultats

Cette réunion fut aussi l'occasion pour chacun des binômes de présenter son travail au reste de l'équipe, que ce soit au niveau de l'analyse fonctionnelle ou de la partie technique. Ainsi, chacun des membres de l'équipe a pu donner son avis quant au travail déjà effectué : les diagrammes de cas d'utilisations ont ainsi pu être améliorés, la maquette a été approuvée par l'ensemble de l'équipe, et la toute première version de l'application (un seul vaisseau représenté par un triangle, et qui se déplace dans une map qui défile) a été testée : les remarques faites par certains des membres ont ainsi pu être prises en compte pour la continuation du codage déjà commencé, et pour la suite de l'analyse fonctionnelle.

▶ Préparation de la prochaine réunion externe

Enfin, nous avons fixé l'ordre du jour de la prochaine réunion avec le client. Nous avons décidé de lui présenter :

- ▷ la maquette du jeu,
- ▷ la première version des diagrammes de cas d'utilisation,
- ▷ le diagramme de navigation,
- ▷ la toute première version de l'application.

Nous aurons aussi l'occasion de lui poser quelques questions dans le but de spécifier un peu plus ses besoins quant au jeu.

53.3 Compte rendu de la réunion interne n°3 du 19/11/08

Cette troisième réunion en interne a eu lieu le mecredi 19/11/08 et a réuni l'ensemble des membres de *e-nov data*. L'ordre du jour a consisté en les points suivants :

- ▷ Faire un état d'avancement du projet,
- ▷ Préparer la prochaine itération.

▶ Mise au point sur l'état d'avancement du projet

Cette troisième réunion interne intervient entre la fin de l'itération n°2 et le début de l'itération n°3. Elle a eu pour but entre autre de faire un état d'avancement du projet : du retard a été accumulé en ce qui concerne la livraison n°2, initialement prévue pour le 14/11/08. Nous avons donc fait un point en ce qui concerne l'état d'avancement des lots qui constituent la livraison n°2, à savoir les lots n°5 et 6 (qui concernent essentiellement la gestion des tirs).
Aussi, nous avons pu finaliser la livraison n°2, qui a donc été livrée à ce jour.

▶ Préparer l'itération n°3

Nous avons également pu préparer ensemble la prochaine itération (n°3) : le chef de projet a créé de nouveaux binômes, répartis chacun à une tâche précise.
L'itération n°3 concerne essentiellement la mise en place des zones de vide et de plein dans une carte, et la gestion des collisions contre ces zones. Leur implémentation ayant été jugée comme assez complexe, nous avons décidé tous ensemble d'un « schéma » de codage afin que la mise en commun du travail de chacun des binômes soit facilitée.
De plus, il a été décidé de commencer le codage de l'interface graphique (menu), à partir des fichiers .ui générés via *QtDesigner* lors de la création de la maquette. Nous avons d'ailleurs dû décidé de la façon dont allait être géré la saisie de l'adresse ip du serveur lorsqu'un client veut s'y connecter (ce point n'avait pas été prévu lors de la création de la maquette) : nous avons donc décidé qu'une boîte de dialogue de type « alerte » permettrait la saisie de cette adresse ip.
Par ailleurs, en ce qui concerne l'analyse, nous avons commencé un diagramme de classes, pour lequel quatre des six membres de l'équipe ont été affectés.

53.4 Compte rendu de la réunion interne n°4 du 26/11/08

Cette quatrième réunion en interne a eu lieu le mecredi 26/11/08 et a réuni l'ensemble des membres de *e-nov data*. L'ordre du jour a consisté en les points suivants :

- ▷ Faire un état d'avancement de l'itération en cours (n°3),
- ▷ Prioriser les tâches les plus importantes, et envisager celles qui seront de l'ordre de l'évolution.

► Mise au point sur l'état de l'itération en cours (n°3)

Cette quatrième réunion interne intervient en pleine itération n°3 : celle-ci a été largement retardée (dû notamment à la complexité de l'implémentation des collisions, mais également à cause des autres projets que nous devons gérer). En effet, prévue de se terminer le 28/11/08, nous l'avons prolongée au début de la semaine prochaine.
D'autre part, l'analyse fonctionnelle doit être avancée, car elle aussi subit un certain retard, certains membres de l'équipe ont alors décidé de mettre l'accent sur l'analyse fonctionnelle.
De plus, nous avons fixé une date de rendez-vous à proposer au client en début de semaine prochaine, afin de lui montrer l'état d'avancement de l'itération en cours.
Par conséquent, la livraison n°3 a dû elle aussi être reportée à (au plus tôt) le début de la semaine prochaine.

► Listing des tâches à traiter en priorité

Cette réunion en interne nous a également permi de faire un point concernant les différentes tâches restantes, d'un point de vue codage et documentation.
Étant donné que la date butoire est dans environ deux semaines et au vu du grand nombre d'options restantes à implémenter (collisions (implémentation en cours), captures d'objets, bonus, ravitaillement en carburant, harpon, modes de jeu death-match, drapeau et entraînement, mini-map « radar »), il a fallu discuter ensemble de celles que nous ne traiterons pas de façon certaine (comme : modes de jeu death-match, drapeau et entraînement, mini-map « radar »), et qui seront à détailler dans le manuel d'évolution.

54 Fiches d'itération

Objet : L'objectif de ce document est de décrire chaque itération : les activités les composant, leurs auteurs, leur dates de livraison prévisionnelle et réelle, ainsi que les raisons des éventuels retards.

Auteurs :

Auteurs	**Approbateurs**	**Validation**
Stephanie MALAKIAN Zeina BAALBAKI	Christophe DEVIN Stéphane GRANGE	Laurent TICHIT
Livré le : *12/12/08*	Approuvé le : *12/12/08*	*En attente de validation*

Diffusion :

Diffusion	*Externe*
À :	Laurent TICHIT
Copies à :	Zeina BAALBAKI, Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Stephanie MALAKIAN, Michael REZAC.

Historique :

Version du document	Date de modification	Modifications apportées	Auteurs des modifications
V1.0	23/10/08	Création du document	Zeina BAALBAKI, Stephanie MALAKIAN
V1.1	26/10/08	Ajout de la fiche d'itération n°0 (itération initiale)	Zeina BAALBAKI, Stephanie MALAKIAN
V1.2	03/11/08	Ajout de la fiche d'itération n°1	Zeina BAALBAKI, Christophe DEVIN
V1.3	19/11/08	Ajout de la fiche d'itération n°2	Fouzi BENMAKHLOUF Stéphane GRANGE
V2.0	05/12/08	Ajout de la fiche d'itération n°3	Stephanie MALAKIAN Michael REZAC
V2.1	12/12/08	Ajout de la fiche d'itération n°4	Stephanie MALAKIAN Michael REZAC

54.1 Fiche d'itération n°0 (itération de lancement)

Début de l'itération : 22/10/08
Fin de l'itération : 26/10/08

Activités	Acteurs	Date prévisionnelle de livraison	Date réelle de livraison	Motif si retard
Fiche des acteurs	Zeina BAALBAKI, Stephanie MALAKIAN	26/10/08	26/10/08	-
PAQ	Zeina BAALBAKI, Stephanie MALAKIAN	26/10/08	26/10/08	-
Analyse financière v1.0	Zeina BAALBAKI, Stephanie MALAKIAN	26/10/08	26/10/08	-
Analyse des risques v1.0	Zeina BAALBAKI, Christophe DEVIN	26/10/08	26/10/08	-
Planning prévisionnel v1.2	Zeina BAALBAKI, Stephanie MALAKIAN	26/10/08	26/10/08	-
Cahier des charges v1.0	Zeina BAALBAKI, Stephanie MALAKIAN	26/10/08	26/10/08	-

FIG. 22 – Description de l'itération n°0 (itération de lancement).

54.2 Fiche d'itération n°1

Début de l'itération : 27/10/08
Fin de l'itération : 08/11/08

Activités	Acteurs	Date prévisionnelle de livraison	Date réelle de livraison	Motif si retard
Première version maquette	Christophe DEVIN, Stéphane GRANGE	07/11/08	08/11/08	*retard global livraison*
Diagramme de navigation	Zeina BAALBAKI, Christophe DEVIN	07/11/08	14/11/08	*retard analyse fonctionnelle*
Premiers diagrammes des cas d'utilisation	Zeina BAALBAKI, Christophe DEVIN, Stephanie MALAKIAN	07/11/08	14/11/08	*retard analyse fonctionnelle*
Premiers diagrammes de séquence	Fouzi BENMAKHLOUF, Stéphane GRANGE, Michael REZAC	07/11/08	14/11/08	*retard analyse fonctionnelle*
Compte rendu réunion externe n°2	Stéphane GRANGE, Stephanie MALAKIAN	07/11/08	08/11/08	*retard global livraison*
Modification maquette	Zeina BAALBAKI, Christophe DEVIN	07/11/08	08/11/08	*retard global livraison*
Début codage : un vaisseau se déplaçant dans une map avec défilement de celle-ci	Fouzi BENMAKHLOUF, Michael REZAC	07/11/08	08/11/08	*retard global livraison*
Codage : début du réseau (classe Client)	Stéphane GRANGE, Stephanie MALAKIAN	07/11/08	08/11/08	*retard global livraison*
Génération documentation du code	Michael REZAC, Stéphane GRANGE	07/11/08	08/11/08	*retard global livraison*
Mise à jour planning	Zeina BAALBAKI, Christophe DEVIN	07/11/08	08/11/08	*retard global livraison*
Manuel de déploiement v1.0	Stephanie MALAKIAN, Michael REZAC	07/11/08	08/11/08	*retard global livraison*
Manuel d'utilisation v1.0	Zeina BAALBAKI, Fouzi BENMAKHLOUF	07/11/08	08/11/08	*retard global livraison*

FIG. 23 – Description de l'itération n°1.

54.3 Fiche d'itération n°2

Début de l'itération : 10/11/08
Fin de l'itération : 19/11/08

Activités	Acteurs	Date prévisionnelle de livraison	Date réelle de livraison	Motif si retard
Compte rendu réunion externe n°3	Zeina BAALBAKI, Michael REZAC	12/11/08	12/11/08	-
Diagramme des classes v1.0	Fouzi BENMAKHLOUF, Christophe DEVIN, Stéphane GRANGE, Stephanie MALAKIAN	12/11/08	*ultérieure*	*trop d'erreurs dans l'analyse*
Codage : gestion des tirs	Fouzi BENMAKHLOUF, Michael REZAC	14/11/08	19/11/08	*retard accumulé*
Codage : gestion de la gravité centrale	Fouzi BENMAKHLOUF, Michael REZAC	14/11/08	19/11/08	*retard accumulé*
Tests : codage du test Client	Zeina BAALBAKI, Christophe DEVIN	14/11/08	*ultérieure*	*test à étoffer*
Compte rendu réunion externe n°4	Christophe DEVIN, Stephanie MALAKIAN	19/11/08	19/11/08	-
Fiche de test n°1	Zeina BAALBAKI, Stéphane GRANGE	14/11/08	*ultérieure*	*test à étoffer*
Début codage du menu	Christophe DEVIN Stephanie MALAKIAN	14/11/08	*ultérieure*	*trop peu avancé, priorité aux autres tâches*
Mise à jour planning	Christophe DEVIN, Michael REZAC	14/11/08	19/11/08	*retard global livraison*

FIG. 24 – Description de l'itération n°2.

54.4 Fiche d'itération n°3

Début de l'itération : 24/11/08
Fin de l'itération : 05/12/08

Activités	Acteurs	Date prévisionnelle de livraison	Date réelle de livraison	Motif si retard
Compte Rendu réunion interne n°4	Zeina BAALBAKI, Fouzi BENMAKHLOUF	-	-	-
Diagramme des classes v2.0 Client	Christophe DEVIN, Stephanie MALAKIAN	01/12/08	*ultérieure*	*à terminer*
Diagramme des classes v2.0 Serveur	Zeina BAALBAKI, Christophe DEVIN	01/12/08	*ultérieure*	*à terminer*
Diagramme des séquences	Fouzi BENMAKHLOUF, Stéphane GRANGE	01/12/08	*ultérieure*	*à terminer*
Maquette v2.0	Zeina BAALBAKI, Stephanie MALAKIAN	12/12/08	-	-
Codage interface v1.0	Christophe DEVIN, Stephanie MALAKIAN	12/12/08	-	-
Compte rendu réunion externe n°5	Christophe DEVIN, Stephanie MALAKIAN	01/12/08	01/12/08	-
Création de la classe QtP_Wall	Fouzi BENMAKHLOUF, Michael REZAC	01/12/08	05/12/08	-
Rédaction tableau de bord V1.0	Zeina BAALBAKI, Michael REZAC	12/12/08	-	-
Gestion des maps par fichier XML et schéma XSD	Zeina BAALBAKI, Michael REZAC	01/12/08	05/12/08	*retard global livraison*
Modification du serveur pour gérer le multi-joueurs	Fouzi BENMAKHLOUF, Stéphane GRANGE	01/12/08	05/12/08	*retard global livraison*

Activités	Acteurs	Date prévisionnelle de livraison	Date réelle de livraison	Motif si retard
Modification du client pour gérer le multi-joueurs	Christophe DEVIN, Stephanie MALAKIAN	01/12/08	05/12/08	*retard global livraison*
Intégration du réseau multi-joueurs	Stéphane GRANGE, Michael REZAC	01/12/08	05/12/08	*débuggage nécessaire, retard global livraison*
Mise à jour planning réel	Stéphane GRANGE, Stephanie MALAKIAN	01/12/08	05/12/08	*retard global livraison*
Manuel de maintenance V1.0	Zeina BAALBAKI, Michael REZAC	12/12/08	-	-
Manuel d'évolutions V1.0	Christophe DEVIN, Stephanie MALAKIAN	12/12/08	-	-

FIG. 25 – Description de l'itération n°3.

54.5 Fiche d'itération n°4 (finale)

Début de l'itération : 08/12/08
Fin de l'itération : 12/12/08

Activités	Acteurs	Date prévisionnelle de livraison	Date réelle de livraison	Motif si retard
Compte rendu réunion externe n°6	Stéphane GRANGE, Michael REZAC	09/11/08	09/11/08	-
Débuggage intégration menu	Zeina BAALBAKI, Christophe DEVIN, Stéphane GRANGE, Stephanie MALAKIAN	12/11/08	12/12/08	-
Génération automatique de la documentation du code	Fouzi BENMAKHLOUF, Michael REZAC	12/12/08	12/12/08	-
Rédaction du Dossier d'évolutions V1.0	Zeina BAALBAKI, Christophe DEVIN	12/12/08	12/12/08	-
Rédaction du Dossier d'évolutions V2.0	Stephanie MALAKIAN, Michael REZAC	12/12/08	12/12/08	-
Rédaction du Manuel de Maintenance	Fouzi BENMAKHLOUF, Stéphane GRANGE, Michael REZAC	12/12/08	12/12/08	-
Mise à jour DAT	Stéphane GRANGE, Michael REZAC	12/12/08	12/12/08	-
Mise à jour Manuel de Déploiement	Stéphane GRANGE, Michael REZAC	12/12/08	12/12/08	-
Mise à jour Manuel d'Utilisation	Zeina BAALBAKI, Christophe DEVIN	12/12/08	12/12/08	-
Diagramme de séquence V3.0	Stephanie MALAKIAN, Michael REZAC	12/12/08	12/12/08	-
Diagramme des cas d'utilisations V3.0	Fouzi BENMAKHLOUF, Christophe DEVIN	12/12/08	12/12/08	-
Diagramme métier V3.0	Zeina BAALBAKI, Christophe DEVIN, Stéphane GRANGE	12/12/08	12/12/08	-
Diagramme des classes V3.0	Fouzi BENMAKHLOUF, Stephanie MALAKIAN, Michael REZAC	12/12/08	12/12/08	-

Activités	Acteurs	Date prévisionnelle de livraison	Date réelle de livraison	Motif si retard
Mise à jour DAF	Zeina BAALBAKI, Christophe DEVIN, Stephanie MALAKIAN	12/12/08	12/12/08	-
Mise à jour planning	Zeina BAALBAKI, Fouzi BENMAKHLOUF	12/12/08	12/12/08	-
Planning réel + Bilan planification	Stéphane GRANGE, Stephanie MALAKIAN	12/12/08	12/12/08	-
Rédaction bilan final du projet	Christophe DEVIN, Stephanie MALAKIAN	12/12/08	12/12/08	-

FIG. 26 – Description de l'itération n°4 (finale).