

Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0



CAHIER DES CHARGES FONCTIONNEL

- **Structure émettrice**
Info1 - Groupe B
- **Emetteurs**
 - CLEMENT Ian ian.clement@etu.univ-nantes.fr
 - DECAMP Grégoire gregoire.decamp@etu.univ-nantes.fr
 - DIAS Fabien fabien.dias@etu.univ-nantes.fr
 - GOUREAU Nicolas nicolas.goureau@etu.univ-nantes.fr
- **Destinataires**
 - TAMZALIT Dalila dalila.tamzalit@uni-nantes.fr
 - MOTTU Jean-Marie jean-marie.mottu@univ-nantes.fr

1. Présentation générale du problème	3
1.1. Projet	3
1.1.1. Finalités.....	3
1.1.2. Espérance de retour sur investissement.....	3
1.2. Contexte.....	3
1.2.1. Études déjà effectuées	3
1.2.2. Études menées sur des sujets voisins	4
1.2.3. Nature des prestations demandées	4
1.2.4. Caractère confidentiel s'il y a lieu	4
1.3. Énoncé du besoin	4
1.4. Environnement du produit recherché	4
1.4.1. Listes exhaustives des éléments et contraintes.....	4
2. Expression fonctionnelle du besoin Diagramme de cas d'utilisation	5
<i>Fonctions Principales</i>	<i>5</i>
<i>Fonctions Complémentaires</i>	<i>6</i>

Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0



Fonctions Techniques	7
3. Cadre de réponse	8
3.1. Pour chaque fonction	8
3.1.1. Solution proposée	8
3.1.2. Part du prix attribué à chaque fonction	9
3.2. Pour l'ensemble du produit	9
3.2.1. Prix de la réalisation de la version de base	9
3.2.2. Options et variantes proposées non retenues au cahier des charges	9
3.2.3. Mesures prises pour respecter les contraintes et leurs conséquences économiques	9
3.2.4. Outils d'installation, de maintenance	10
3.2.5. Décomposition en modules, sous-ensembles	10
3.2.6. Prévisions de fiabilité	10
4. Annexes	10
4.1. Diagramme de classe général	10
4.2. Diagramme de Cas d'utilisation	11
4.3. Diagrammes Bête à cornes	12
4.3.1. FC1	12
4.3.2. FC2	13
4.3.3. FC3	14
4.3.4. FC4	15
4.3.5. FP1	16
4.3.6. FP2	17
4.3.7. FP3	18
4.3.8. FT1	19
4.3.9. FT2	20

Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0



1. Présentation générale du problème

1.1. Projet

1.1.1. Finalités

Notre projet a pour but de développer un mastermind qui sera accessible à tous. Pour cela nous essayerons de faire une interface graphique qui conviendra aux jeunes mais aussi aux personnes plus âgées. Celle-ci sera aussi simple d'utilisation et correspondra aux attentes que l'on peut avoir dans le développement d'un tel jeu. De plus, nous rajouterons de petits éléments qui rendront notre projet unique. Notre jeu comprendra une intelligence artificielle contre laquelle le joueur pourra jouer mais aussi un mode 1 contre 1 pour que deux joueurs distincts puissent s'affronter. L'intelligence artificielle sera aussi capable de compter les points, de générer des combinaisons aléatoires et aussi de préciser au joueur recherchant la combinaison, quels pions sont bien placés et lesquels ne le sont pas ou du moins ne sont pas à la bonne place, tout cela avec un code couleur spécifique.

1.1.2. Espérance de retour sur investissement

Comme cité précédemment notre Mastermind sera destiné à tout public. Nous avons plus précisément pour but de combler toutes les personnes de 7 à 77 ans. En effet, le jeu basé sur un pattern MVC (Modèle vue contrôleur) va nous permettre de réaliser les souhaits de nos futurs utilisateurs avec simplicité. Ce pattern permet de combiner interface homme machine(IHM)/code grâce à un contrôleur qui fera la transition entre les deux. Nous voulons viser un public assez large car le Mastermind est un jeu qui permet de développer sa réflexion, ce qui est bon pour les jeunes personnes mais aussi de conserver une bonne concentration et réflexion ce qui est aussi parfait pour les utilisateurs un peu plus âgés. En outre ce jeu permet aussi de rassembler des gens autour d'une partie tout cela dans une bonne ambiance. En tant qu'analyste programmeur, notre investissement sera récompensé par une expérience évidemment et aussi par une ouverture au concept objet.

1.2. Contexte

1.2.1. Études déjà effectuées

Nos études se résument au fonctionnement du jeu en lui-même, et à la résolution de sa complexité algorithmique. En effet, il nous semble intéressant, en plus du mode joueur contre joueur, de développer un IA de plus ou moins haut niveau. Nous nous sommes intéressés aux codes de déchiffrement et de cassage du Mastermind, comme celui proposé par Knuth, qui permettent de mieux appréhender les enjeux et la complexité du jeu en lui-même.

Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0



1.2.2. Études menées sur des sujets voisins

Nous avons étudié les moyens de rendre le jeu modulable tant dans son « gameplay » (difficulté) que dans sa construction (pattern). Ainsi, nous nous sommes informé sur le pattern Modèle-Vue-Contrôleur, qui nous semble une bonne option dans le développement d'un code objet modulable. Ensuite, pour les perspective d'IHM, une librairie proposé sur Internet, et compatible avec Java nous as interpellé.

1.2.3. Nature des prestations demandées

Logiquement, notre projet doit se résumer au développement du Mastermind en langage objet, de manière a d'abord pouvoir jouer, joueur contre joueur. Il faut aussi que notre code soit modulable et efficace évidemment. Il nous faudra aussi ajouter une interface graphique à tout cela. Elle sera ajoutée plus tard, d'où l'importance de la modularité du code.

1.2.4. Caractère confidentiel s'il y a lieu

Notre code sera notre propriété, donc il devra rester plus ou moins confidentiel, d'où l'utilisation d'un dépôt Svn, qui permet de garder privé toutes les sources de notre projet. A part cela, notre projet n'as pas lui d'être entouré d'un secret trop épais, il ne représente pas une technologie ou une nouveauté qui puisse être convoitée par d'autre entreprise. Il n'empêche que des précautions seront à prendre quand a la diffusion des sources, surtout si le programme est en Java.

1.3. Énoncé du besoin

La finalité du projet est de parvenir à une maquette de code capable de craquer une combinaison proposé par un utilisateur. C'est en cela que le projet est intéressant algorithmiquement. L'algorithme sera testé et choisi en fonctions des performances affichées. De plus, l'interface graphique sera aussi un point important du développement puisqu'elle consistera en l'aspect ludique du jeu.

1.4. Environnement du produit recherché

1.4.1. Listes exhaustives des éléments et contraintes

Le produit final sera intuitif et portable. En effet, pour permettre l'adhésion du plus grand nombre, il faudra que l'interface soit lisible et claire. De plus, le format java permettra la portabilité du système. Les prérequis seront un système informatique, et une machine java a jour.

Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0



2. Expression fonctionnelle du besoin Diagramme de cas d'utilisation

Ce tableau regroupe toutes les fonctions de notre projet. On y retrouve les **Fonctions Principales**, les **Fonctions Complémentaires**, ainsi que les **Fonctions Techniques**. C'est à partir de ce tableau que nous allons pouvoir concevoir les différents diagrammes de classe de chaque fonction.

Fonction	Importance	Critères d'appréciation	Niveau d'appréciation
Fonctions Principales			
FP1 : Paramétrage partie.	4	<ul style="list-style-type: none"> - Choix de la couleur des pions. - Nombres d'essais. - Nombre de pions de la combinaison secrète. - Possibilité de réutiliser une même couleur. 	<ul style="list-style-type: none"> - Minimum 6 couleurs, maximum 8 couleurs. - Minimum 10 d'essais et 15 max. - Minimum 4 pions dans la combinaison secrète, Max 6 couleurs. - La combinaison secrète peut avoir ou non plusieurs fois la même couleur.
FP2 : Création de combinaison secrète.	5	<ul style="list-style-type: none"> - L'utilisateur doit choisir le nombre de pions définis préalablement. - L'utilisateur a le choix dans les couleurs disponibles. 	<ul style="list-style-type: none"> - Le nombre de pions dans les combinaisons est égale au nombre de pions choisie préalablement. - Seuls les couleurs disponibles sont utilisés.
FP3 : Proposition et validation de la solution.	5	<ul style="list-style-type: none"> - L'utilisateur doit pouvoir construire une solution ayant le bon nombre de pions et utilisant les couleurs associés à la partie. - Le joueur doit pouvoir soumettre sa combinaison pour la comparer avec la combinaison secrète. 	<ul style="list-style-type: none"> - La solution doit être en format comparable à la combinaison secrète. - Le joueur doit connaître les pions de bonnes couleurs bien placés. - Le joueur doit connaître les pions de bonnes couleurs mal placés.

Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0



Fonctions Complémentaires			
FC1 : Fenêtre graphique de partie.	4	<ul style="list-style-type: none"> - Une fenêtre graphique s'affiche pour modéliser la partie en cours. - La combinaison secrète doit pouvoir être occultée. - La proposition doit être visible et modifiable à partir de cette fenêtre. - Le joueur doit pouvoir savoir combien d'essai il a effectué. 	<ul style="list-style-type: none"> - La fenêtre doit afficher les solutions. - La combinaison secrète n'est pas visible. - La composition de la proposition ce fait via l'interface graphique. - Le nombre d'essai est affiché (explicitement ou implicitement).
FC2 : Fenêtre graphique du menu.	3	<ul style="list-style-type: none"> - Une fenêtre graphique s'affiche pour pouvoir lancer une nouvelle partie ou voir les crédits. 	<ul style="list-style-type: none"> - La fenêtre possède un bouton pour lancer le paramétrage de partie. - La fenêtre possède un deuxième bouton pour lancer la fenêtre de crédit.
FC3 : Fenêtre graphique de paramétrage de partie.	3.5	<ul style="list-style-type: none"> - Une fenêtre graphique doit s'afficher pour que l'utilisateur puisse modifier les paramètres de la partie qu'il va lancer. 	<ul style="list-style-type: none"> - Un objet graphique permet le choix du nombre de couleurs dans les combinaisons. - Un objet graphique permet le choix du nombre de pions dans les combinaisons. - Un objet graphique permet le choix du nombre d'essais disponible aux joueurs. - Un objet graphique permet l'autorisation du choix multiple de couleurs dans une combinaison.
FC4 : Fenêtre graphique des crédits.	2	<ul style="list-style-type: none"> - Une fenêtre graphique s'affiche pour que l'utilisateur affiche les crédits du programme. 	<ul style="list-style-type: none"> - Un texte affiche tous les développeurs et les programmes utilisés

Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0



Fonctions Techniques			
FT1 : Comptage du score.	3	- Le score doit être calculé.	<ul style="list-style-type: none"> - Le score calculé doit être correcte à 100%. - Le score est calculé à partir du nombre d'essais effectués.
FT2 : Switch entre les joueurs.	5	- Le programme doit pouvoir gérer le tour des joueurs.	<ul style="list-style-type: none"> - Le programme ne commet aucune erreur sur le tour de jeux. - Le 1^{er} joueur choisit une combinaison secrète. - Puis le 2^{ème} joueur émet une proposition jusqu'à ce que le nombre essais soit épuisé ou que la proposition soit valide. - Le 1^{er} joueur compare à chaque proposition du 2^{ème} joueur. (→ voir pour proposer comparaison automatique).

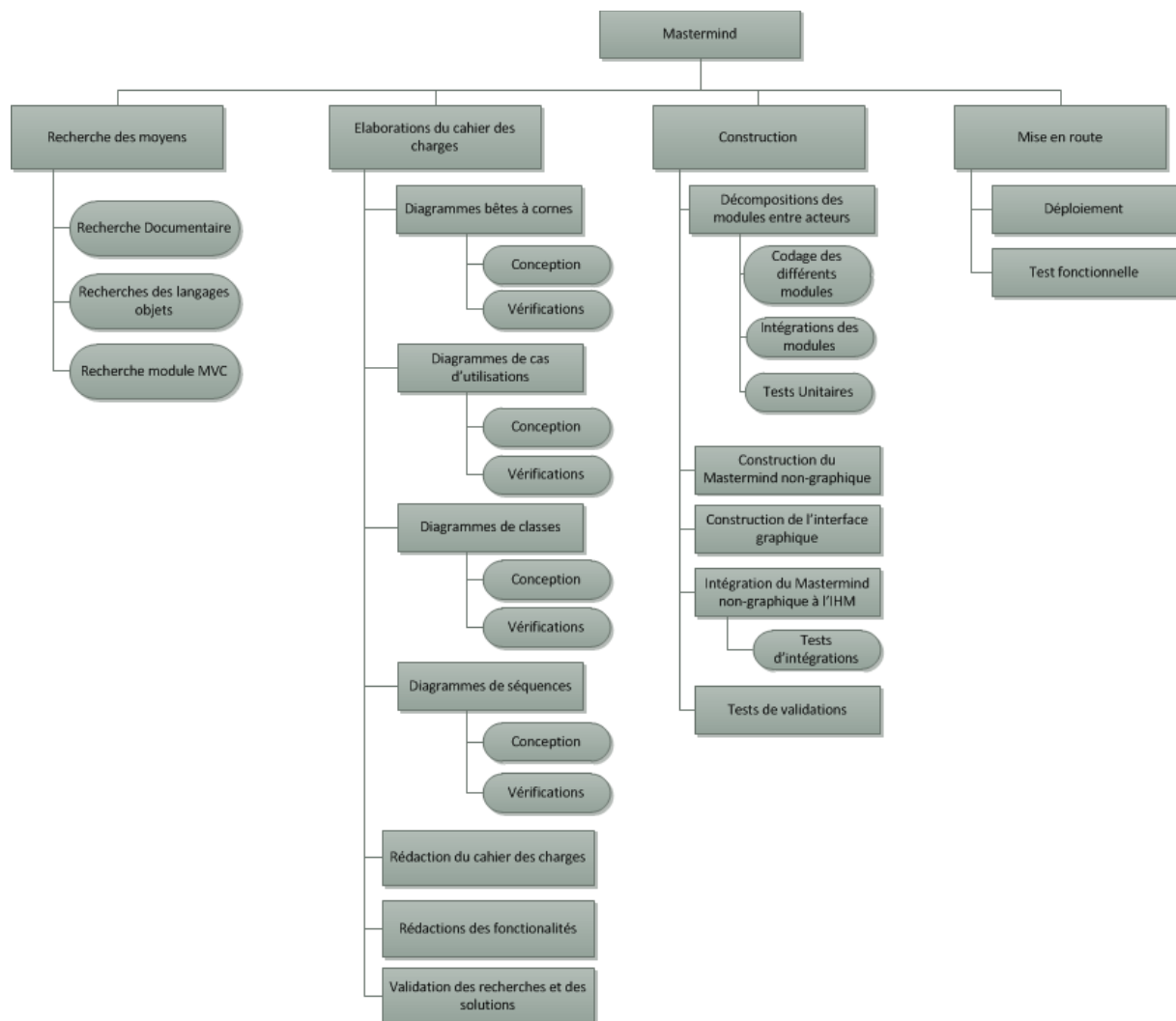
Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0

3. Cadre de réponse

3.1. Pour chaque fonction

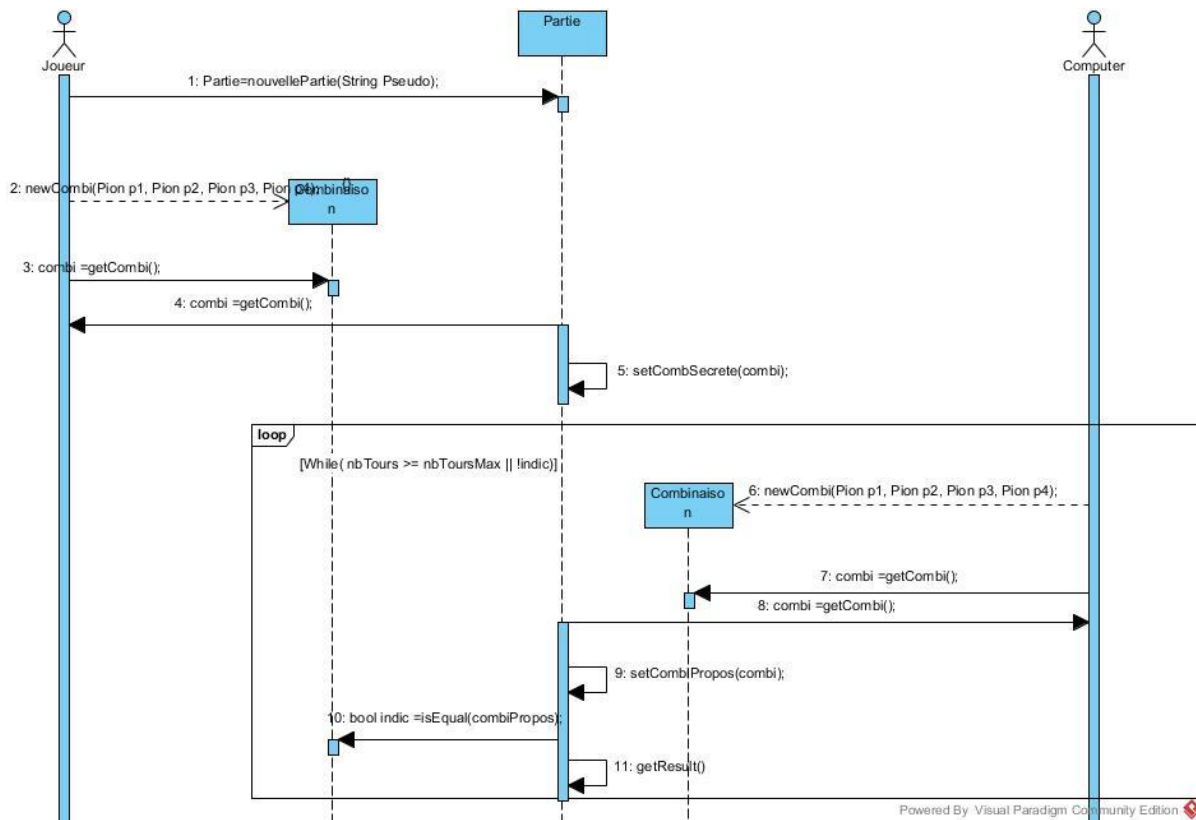
3.1.1. Solution proposée

Diagramme WBS



Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0

Diagramme de séquence



3.1.2. Part du prix attribué à chaque fonction

Chaque tâche aura la même importance, puisque le développement sera linéaire et partagé.

3.2. Pour l'ensemble du produit

3.2.1. Prix de la réalisation de la version de base

Nous ne prévoyons aucun prix pour la réalisation de base du fait que nous utilisons des logiciels gratuits.

3.2.2. Options et variantes proposées non retenues au cahier des charges

Mettre le jeu sur une plateforme de jeu en ligne.

3.2.3. Mesures prises pour respecter les contraintes et leurs conséquences économiques

Notre principal contrainte est le temps c'est pour que cela nous nous efforçons de travailler rapidement et efficacement.

Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0

3.2.4. Outils d'installation, de maintenance

Aucun.

3.2.5. Décomposition en modules, sous-ensembles

Nous décomposerons notre code en plusieurs parties afin qu'il soit lisible et assez simple.

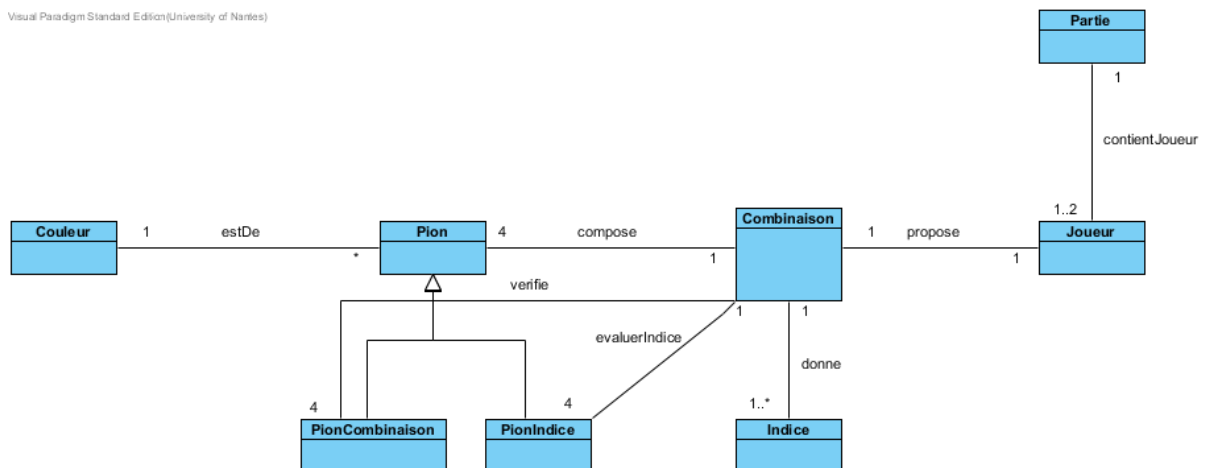
3.2.6. Prévisions de fiabilité

Nous ne pouvons pas nous prononcer pour le moment.

4. Annexes

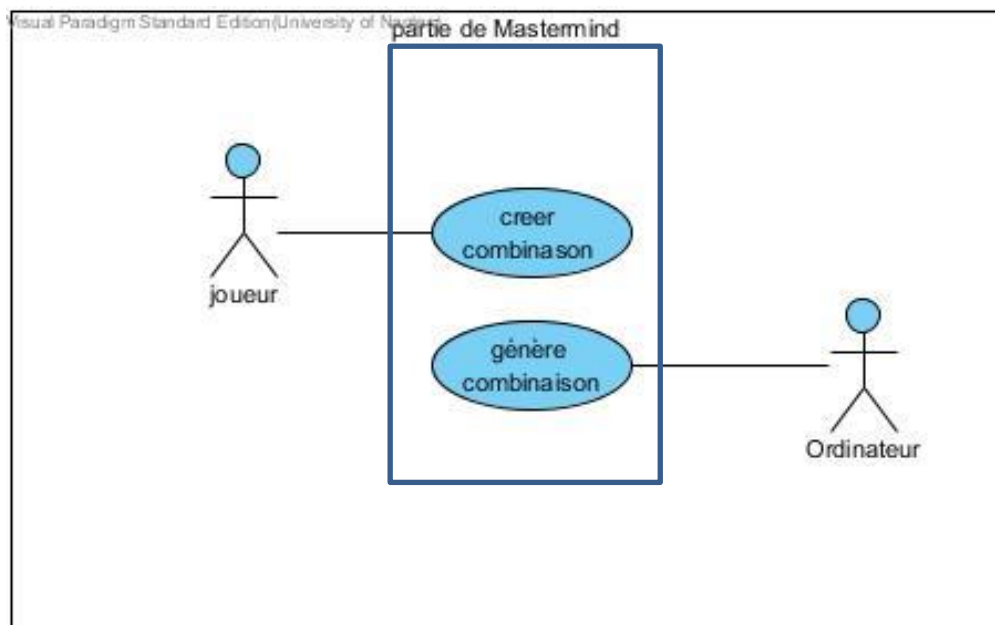
4.1. Diagramme de classe général

Visual Paradigm Standard Edition (University of Nantes)

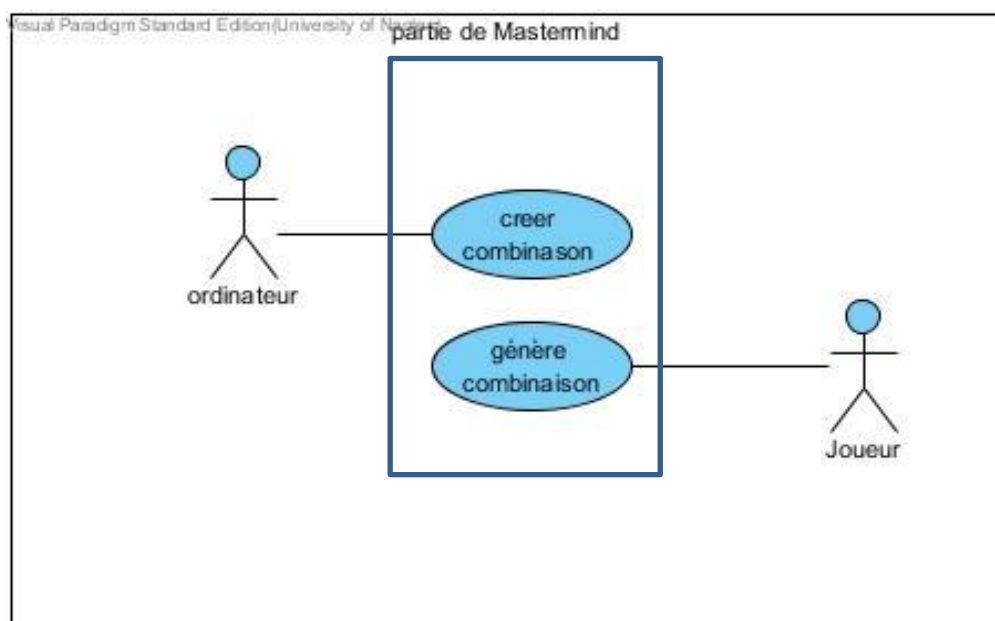


Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0

4.2. Diagramme de Cas d'utilisation



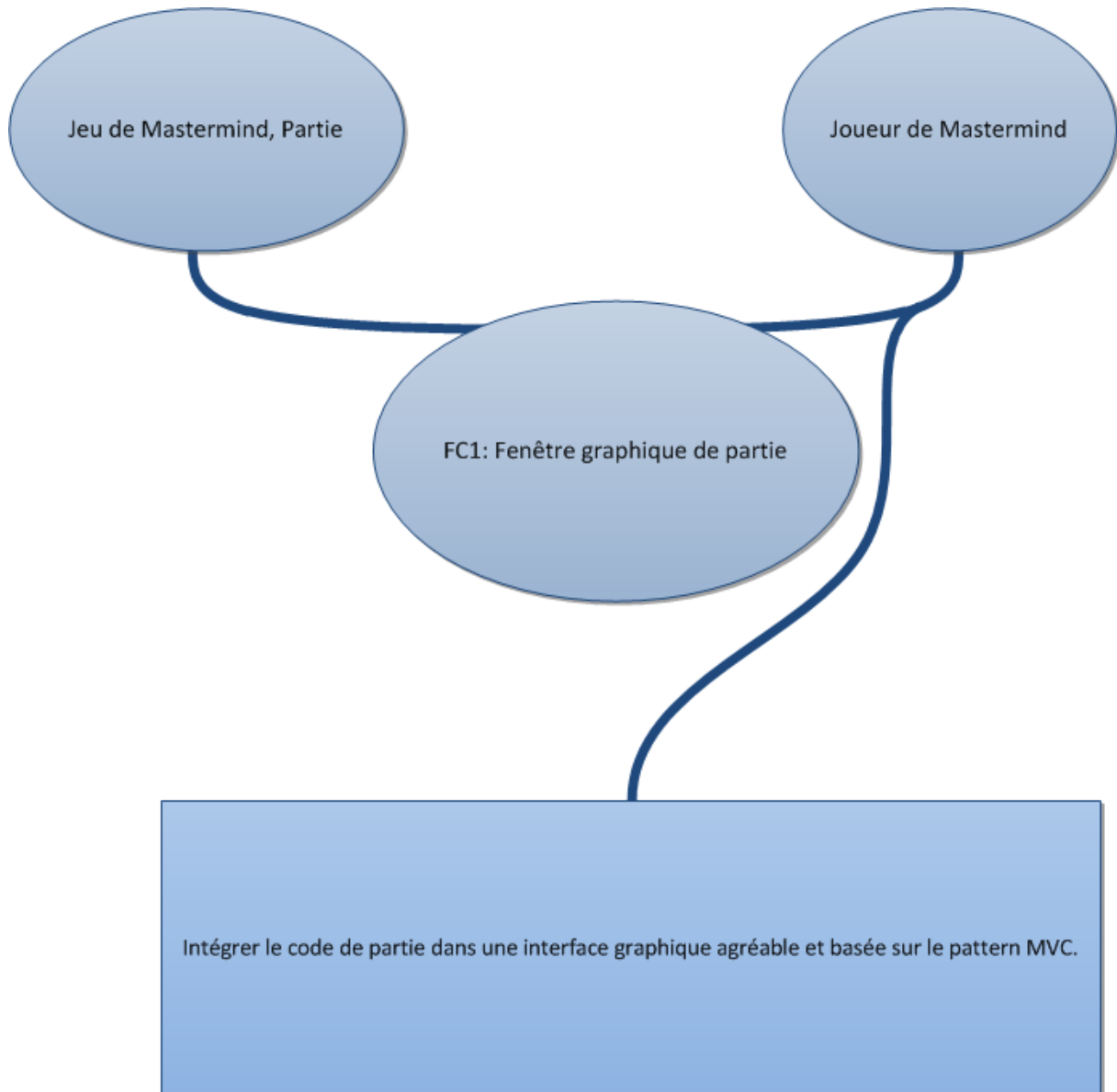
Ces diagrammes de cas d'utilisation nous indiquent que les algorithmes de génération de combinaison secrète diffèrent selon l'acteur.



Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0

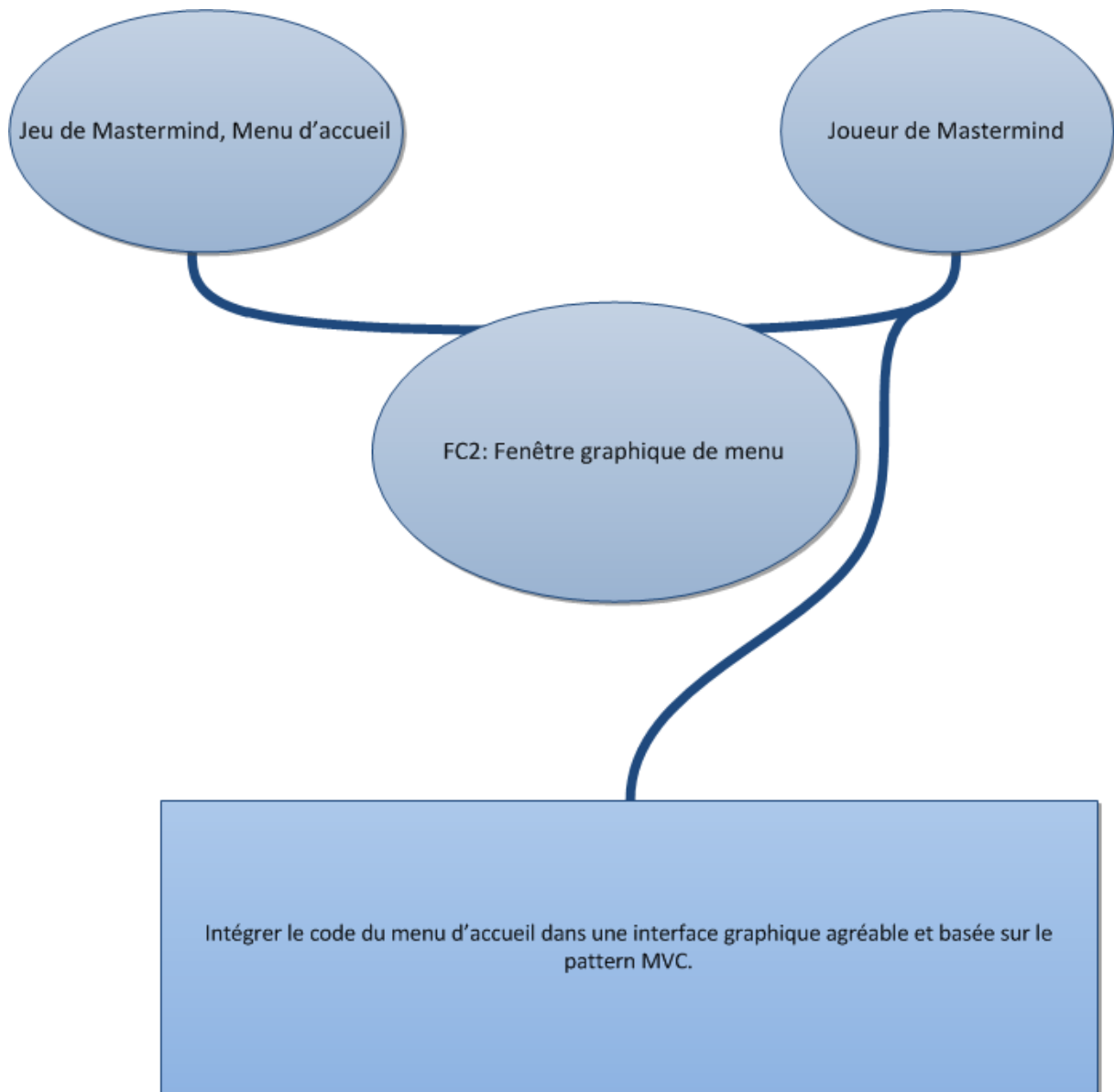
4.3. Diagrammes Bête à cornes

4.3.1. FC1



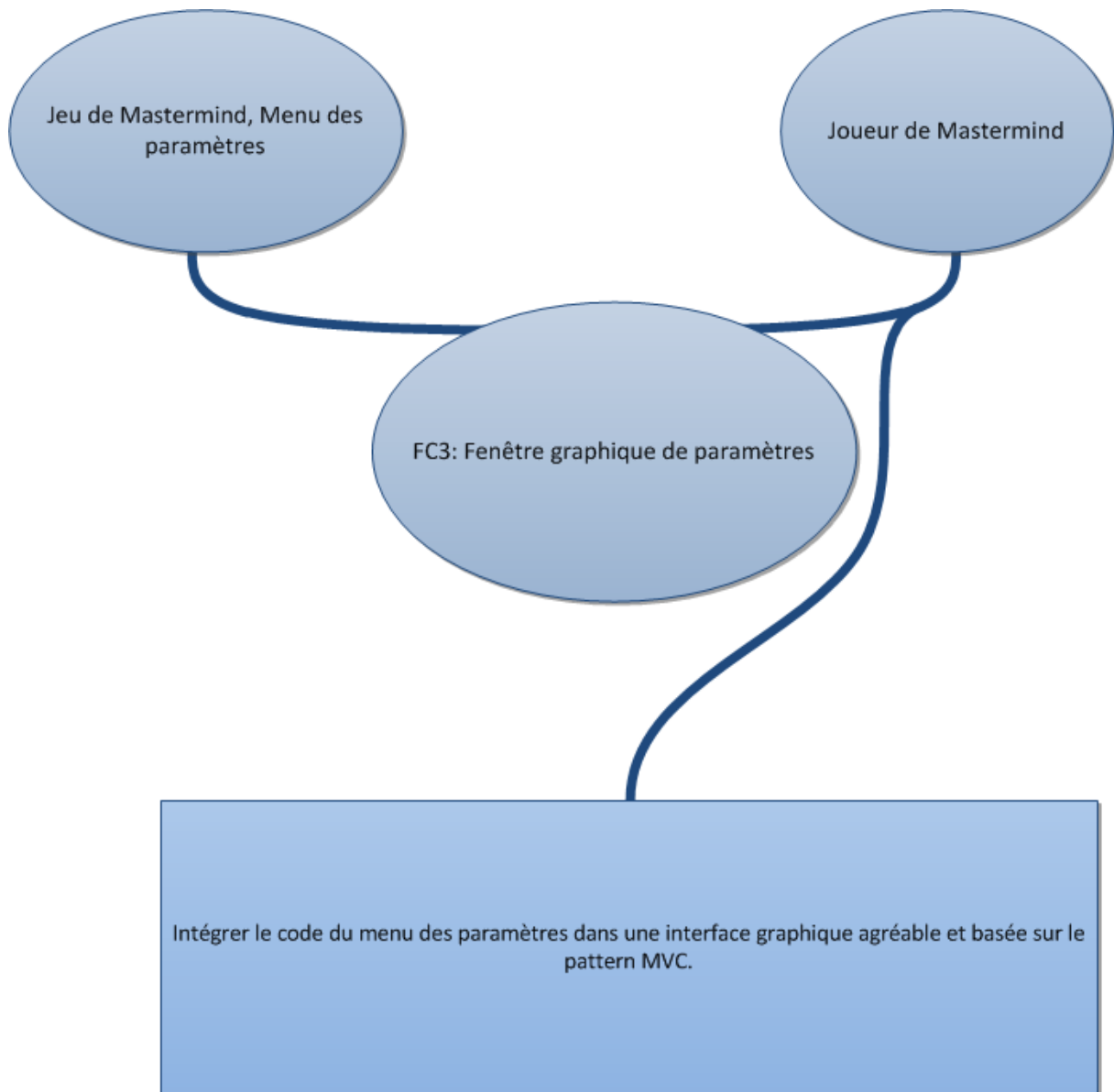
Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0

4.3.2. FC2



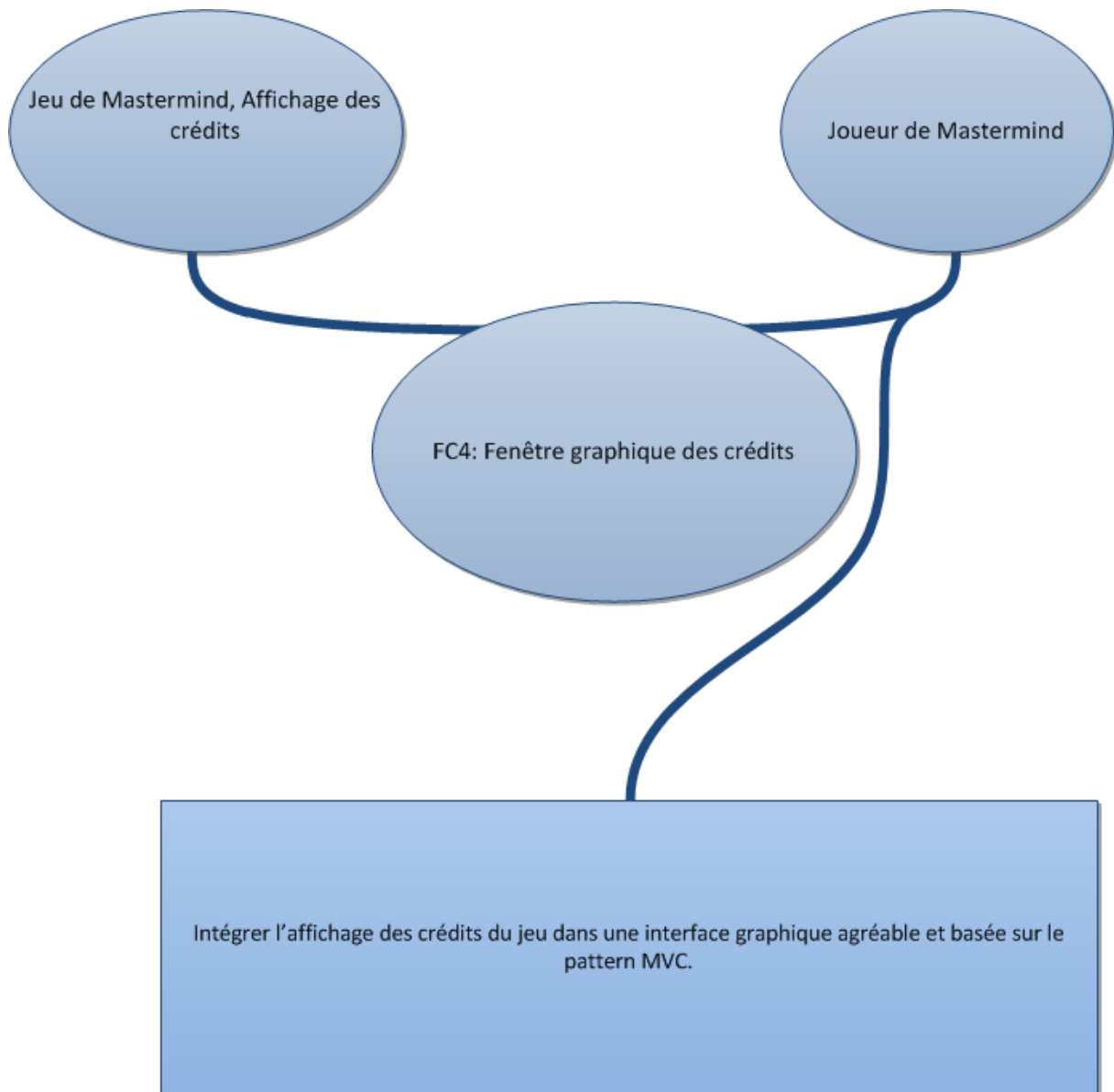
Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0

4.3.3. FC3



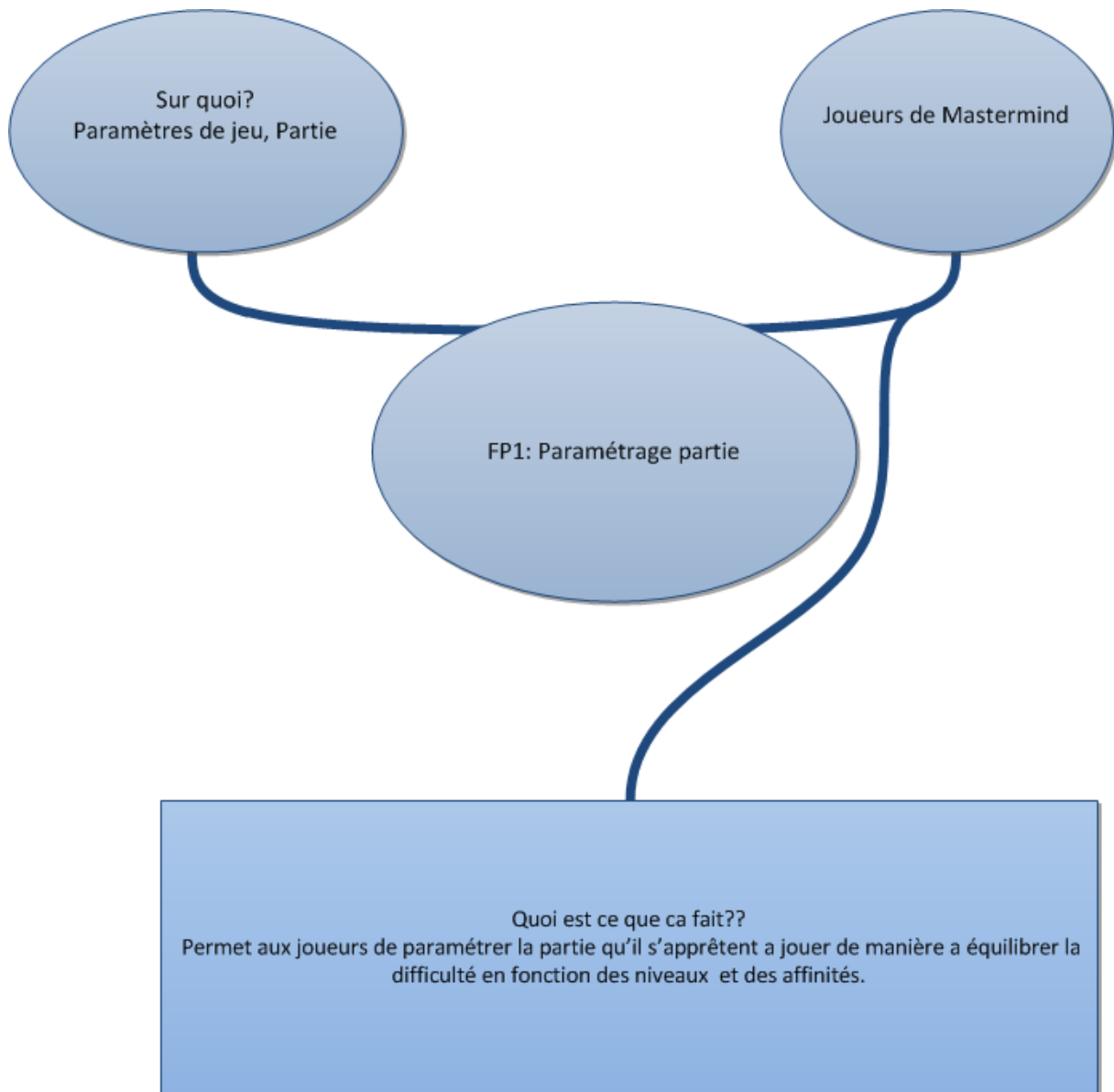
Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0

4.3.4. FC4



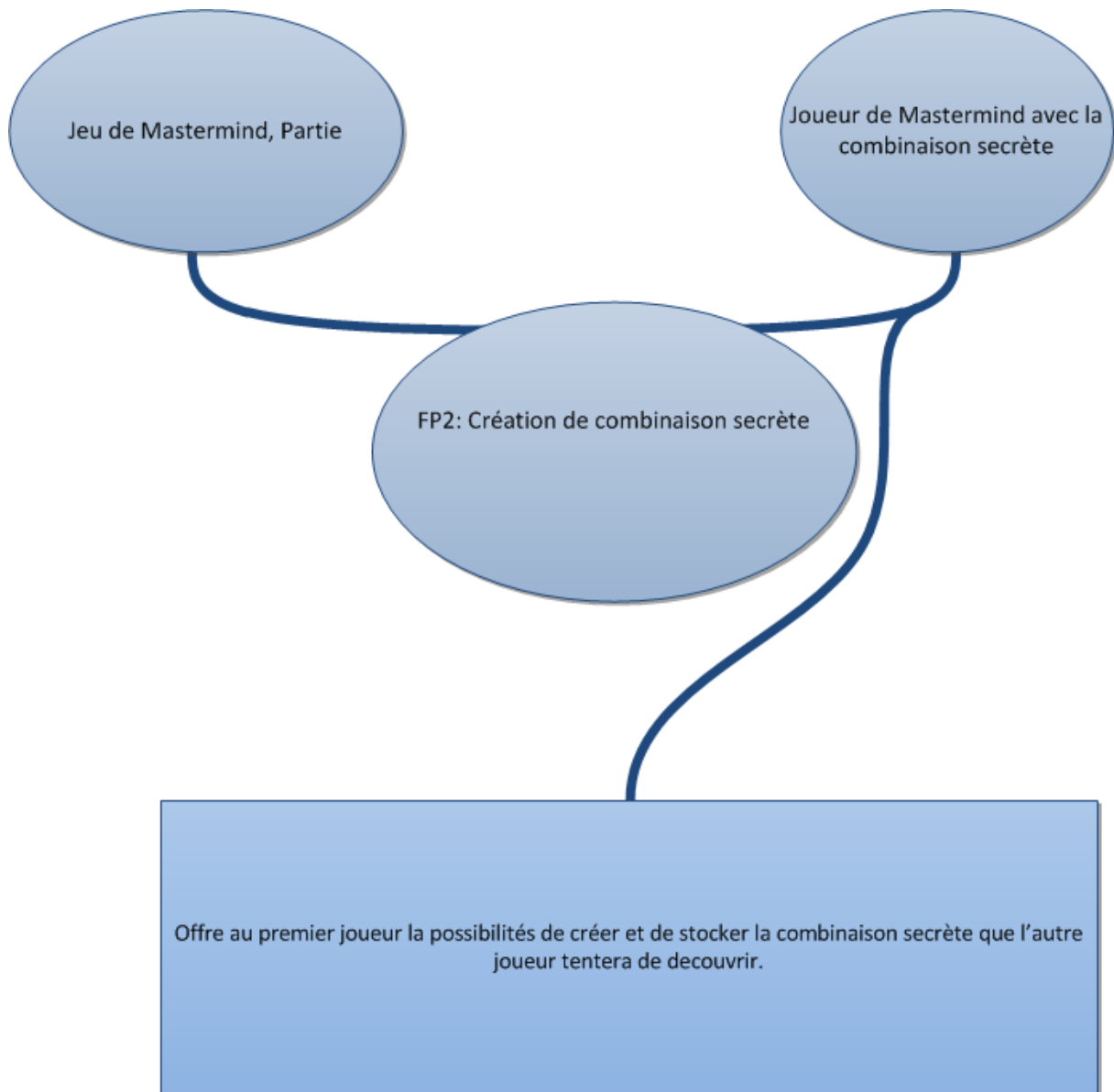
Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0

4.3.5. FP1



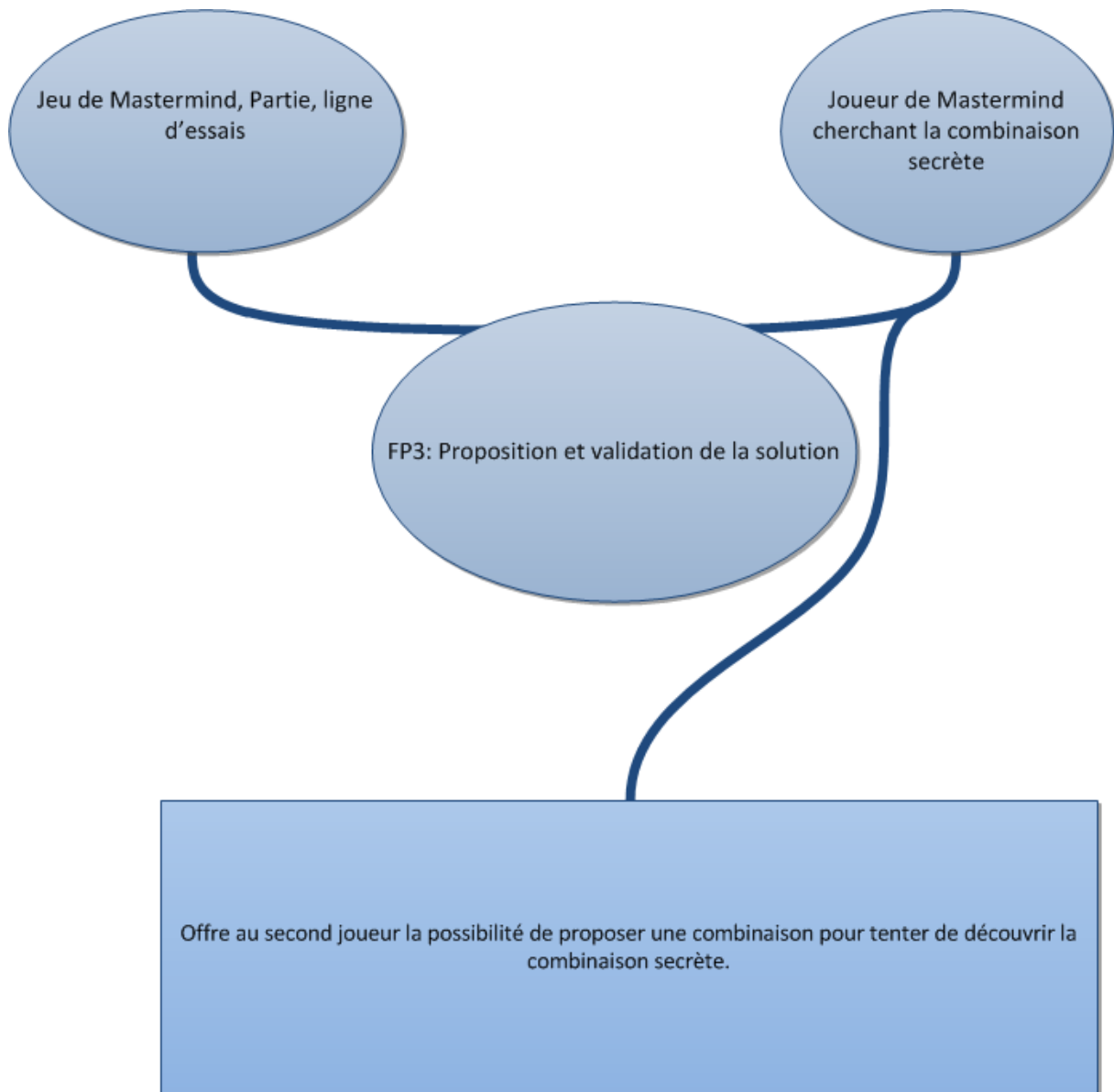
Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0

4.3.6. FP2



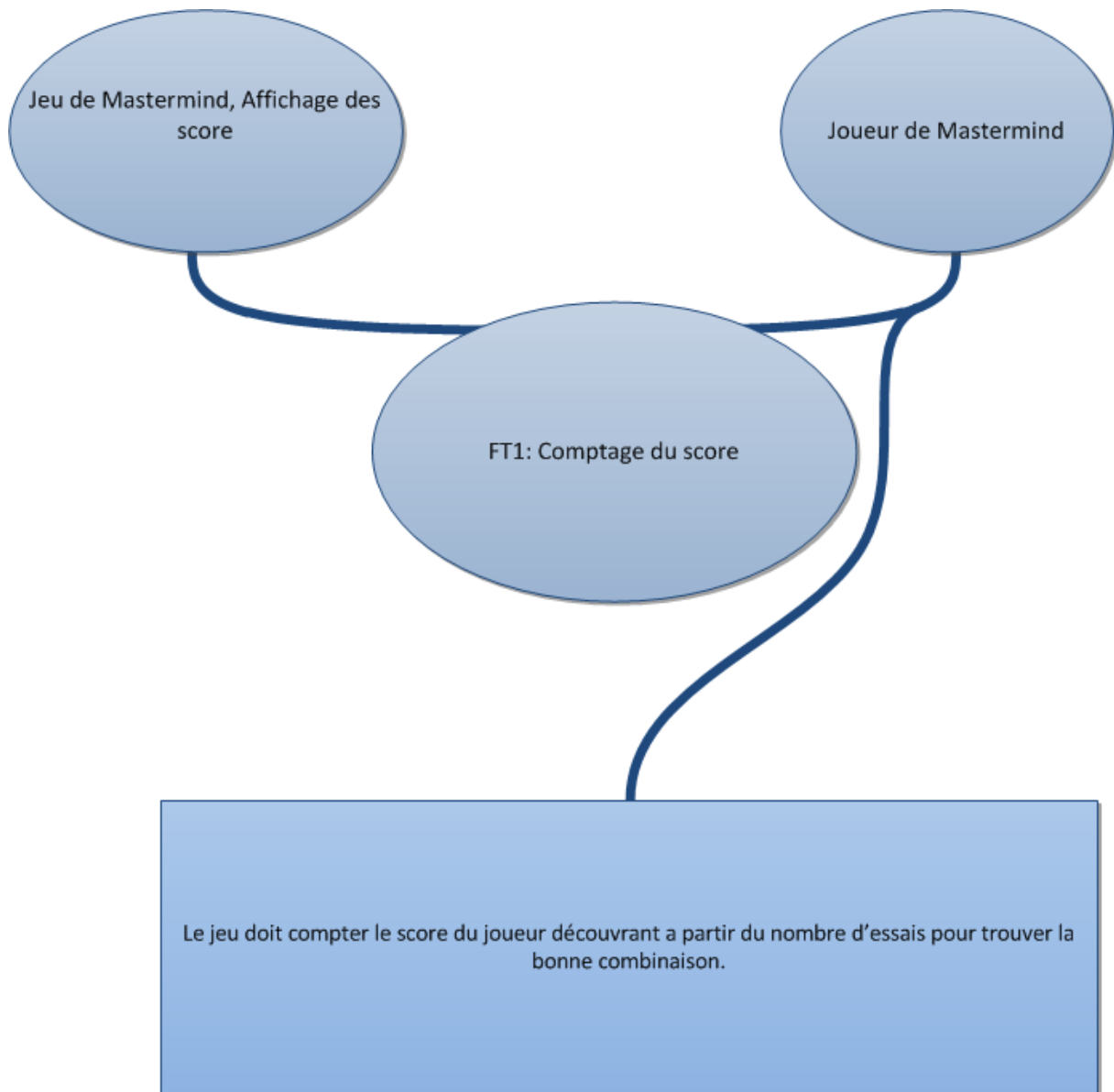
Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0

4.3.7. FP3



Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0

4.3.8. FT1



Réf. Projet	Sujet n°2
Désignation Projet	Mastermind
Date d'émission	03/04/2015
Version	1.0

4.3.9. FT2

