

## **Livrable n°2 : Cahier des fonctionnalités**



### **Conception et Programmation Objet - Gestion de projet**

---

**Destinataires :** M. Xavier **Gregut**, M. Neeraj **Singh**, M. Loic **Thierry**

**Numéro de l'équipe :** 4

**Membres de l'équipe :**

| <u>Nom</u>       | <u>Prénom</u> | <u>Rôle</u>                       |
|------------------|---------------|-----------------------------------|
| <b>François</b>  | Yoann         | Développeur - Responsable IHM     |
| <b>Pradier</b>   | Corentin      | Développeur - Responsable BackEnd |
| <b>Loppinet</b>  | Stéphane      | Développeur - Responsable QA      |
| <b>Silvestre</b> | Thomas        | Développeur - Architecte logiciel |
| <b>Ziuzin</b>    | Nikita        | Développeur - UI/UX               |
| <b>Chaveroux</b> | Pierre        | Développeur - Chef de projet      |

## Historique d'édition :

| <u>Numéro de version</u> | <u>Date</u>          | <u>Modification</u>            |
|--------------------------|----------------------|--------------------------------|
| V0.1                     | Vendredi 9 Mai 2025  | Rédaction des fonctionnalités  |
| V0.2                     | Dimanche 11 Mai 2025 | Création du document livrable  |
| V0.3                     | Du 12 au 15 Mai 2025 | Complétion du livrable         |
| <b>V1.0</b>              | Jeudi 16 Mai 2025    | Relecture et dépôt du livrable |

## Sommaire

|  |           |
|--|-----------|
| <b>1. Contexte et objectif général.....</b>            | <b>1</b>  |
| <b>2. Fonctionnalités du jeu.....</b>                  | <b>2</b>  |
| 2.1. Fonctionnalités principales :.....                | 2         |
| 2.2. Fonctionnalités secondaires :.....                | 3         |
| 2.3. Fonctionnalités qui font la différence.....       | 3         |
| <b>3. Interfaces utilisateurs (maquette).....</b>      | <b>4</b>  |
| 3.1. Vue globale.....                                  | 4         |
| 3.2. Menu principal.....                               | 4         |
| 3.3. Lancement d'une partie.....                       | 5         |
| 3.3.1. Choix des personnages.....                      | 5         |
| 3.3.2. Choix de la carte et du nombre des manches..... | 5         |
| 3.4. Partie de Bomber7.....                            | 5         |
| 3.5. Autres.....                                       | 6         |
| 3.5.1. Menu pause.....                                 | 6         |
| 3.5.2. Paramètres.....                                 | 6         |
| 3.5.3. Configuration des touches.....                  | 6         |
| <b>4. Cas d'usage.....</b>                             | <b>7</b>  |
| 4.1. Jeu solo.....                                     | 7         |
| 4.2. Jeu multijoueur local.....                        | 9         |
| <b>5. Annexes - Gestion de projet.....</b>             | <b>10</b> |
| 5.1. Phase de définition.....                          | 10        |
| 5.1.1. Matrice SWOT.....                               | 10        |
| 5.2. Phase de montage.....                             | 11        |
| 5.2.1. Matrice RACI.....                               | 11        |
| 5.2.2. Liste des livrables.....                        | 12        |
| 5.2.3. Diagramme de GANTT.....                         | 13        |
| 5.3. Analyse des risques.....                          | 14        |
| 5.3.1. Matrice des risques.....                        | 14        |
| 5.3.2. Matrice des objectifs.....                      | 14        |

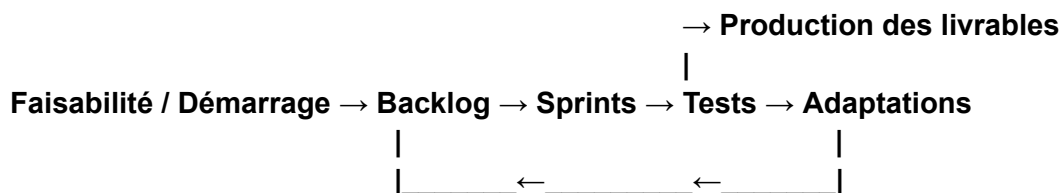
## 1. Contexte et objectif général

Bomber7 est un projet imaginé par notre équipe dont l'objectif est de créer un jeu multijoueur local compétitif en 2D inspiré de Bomberman, un jeu rétro apparu en 1983 sur MSX et ZX Spectrum. Les joueurs s'affrontent de 1 à 4 joueurs sur des cartes de jeux 2D représentatives de notre école, l'ENSEEIH, où l'objectif est de faire exploser les adversaires (monstres et joueurs humains/IA).

Les mécanismes de notre jeu ont pour objectifs d'être complets afin de proposer une expérience utilisateur agréable et différence entre chaque partie : PowerUps, cartes différentes, aléatoires dans le déplacement des ennemis. Plusieurs modes de jeu seront possibles en fonction de la difficulté des IA et monstres, de facile à Chaos.

L'interface utilisateur adoptée sera épurée, avec des menus à compréhension simple : boutons, sliders, menus défilants... La musique de jeux et les sons d'animation (explosions, marche, bruits des monstres, etc) seront libres de droits. L'utilisation des assistants IA (Dallee, ...) sera utilisée pour la génération des images (logos, "sprites", etc, ...), l'aspect graphique sortant du cadre d'évaluation de ce projet.

Dans notre projet, nous tâcherons d'adopter une méthodologie de gestion de projet par méthodes agiles, afin d'obtenir des résultats tangibles fidèles à ce cahier de fonctionnalités. Ainsi, nous adopterons de courtes itérations du cycle de développement suivant :



Notre objectif dans l'adoption de cette méthodologie agile est de pouvoir s'extraire des contraintes de la gestion de projet classique : faible réactivité, effet tunnel, détection des problèmes tardive et documentation excessive.

Nous fonctionnerons sur un format de *sprints* d'une semaine. Chaque jeudi midi sera dédié à un *weekly* permettant de faire le point sur l'avancée respective des membres du groupe vis-à-vis de leurs tâches attribuées. Un jour en fin de semaine sera quant à lui consacré au passage en revue du *backlog* : on raffinerait certaines tâches si nécessaire pour ensuite discuter de la répartition de ces dernières dans le *sprint* suivant. Le but sera de morceler au mieux le projet tout en s'assurant de la cohésion du groupe dans le processus de développement tout en nous permettant de rapidement identifier les points bloquants.

Afin de compléter notre cahier de fonctionnalités, vous trouverez en annexe les livrables de gestion de projet standards en phase de faisabilité et de démarrage : une analyse SWOT, une matrice RACI et un diagramme de GANTT prévisionnel. Bien que les diagrammes de séquence et UML ont déjà été produits par le groupe pour avancer sur ce projet, ces derniers ne seront pas fournis en accords avec les consignes de ce livrable.

## 2. Fonctionnalités du jeu

L'objectif de la seconde partie de ce cahier de fonctionnalités est de détailler ces dernières. Nous avons fait le choix de les scinder en trois parties distinctes, en fonction de leur importance et de la chronologie d'implémentation dans notre jeu.

Dans un souci de lisibilité, nous détaillerons les fonctionnalités à l'aide d'une liste à puce.

### 2.1. Fonctionnalités principales :

- **Carte de jeu :** Grille 2D avec cases destructibles et indestructibles
- **Direction artistique :** Les cartes de jeu sont représentatives des salles de l'ENSEEIH
- **Logique du personnage principal :**
  - Déplacement dans les 4 directions 2D de l'espace (haut, bas, gauche, droite)
  - Pose de bombes
  - Collisions avec les murs et obstacles
- **Logique des bombes :**
  - Explosion après un délai donné (En fonction des PowerUps)
  - Rayon d'explosion variable (En fonction des PowerUps)
  - Destruction des blocs destructibles uniquement
- **Logique des ennemis :**
  - Déplacement aléatoire des monstres
  - Déplacement semi-intelligent des joueurs artificiels
    - Algorithme basé sur l'aléatoire (Facile)
    - Algorithme basé sur le calcul de distances (Intermédiaire)
    - Algorithme avec adaptation du comportement (Difficile, avec possible implantation de Machine Learning)
  - Élimination par les explosions

## 2.2. Fonctionnalités secondaires :

- **PowerUp** : blocs permettant d'obtenir des pouvoirs
  - Gilet pare-balle (insensible aux bombes pendant 30 secondes)
  - Plus de portée sur les bombes
  - Poser 1 bombe supplémentaire
  - Obtention d'une vie supplémentaire
  - Glisse sur Rollers : permet de découpler la vitesse de déplacement (x2)
  - PowerUp avancées :
    - Chevaucher un "Yoshi" : vitesse de déplacement accrue (x4), saut de blocs et vue supplémentaire
    - Bombe à détonation manuelle (choix du timing d'explosion)
    - Malus aléatoire : commandes inversées, ralentissement...
- **Interface Graphique Utilisateur** :
  - Animations des personnages : "sprites" (mouvement des jambes, bras, ...)
  - Menu principal : choix du nombre de joueurs, du nombre de robots, de la carte de jeu, ...
  - Menu pause : paramètres de jeu, quitter, retour, ...
- **Multijoueur local** :
  - Jeu sur un seul clavier de un à quatre joueurs
  - Bindings de touches personnalisables pour chaque joueur
- **Mode de jeu en équipe (2 contre 2)** :
  - Les participants d'une même équipe choisissent une couleur
  - Ces derniers ne peuvent pas s'entre-tuer et chercheront à gagner ensemble

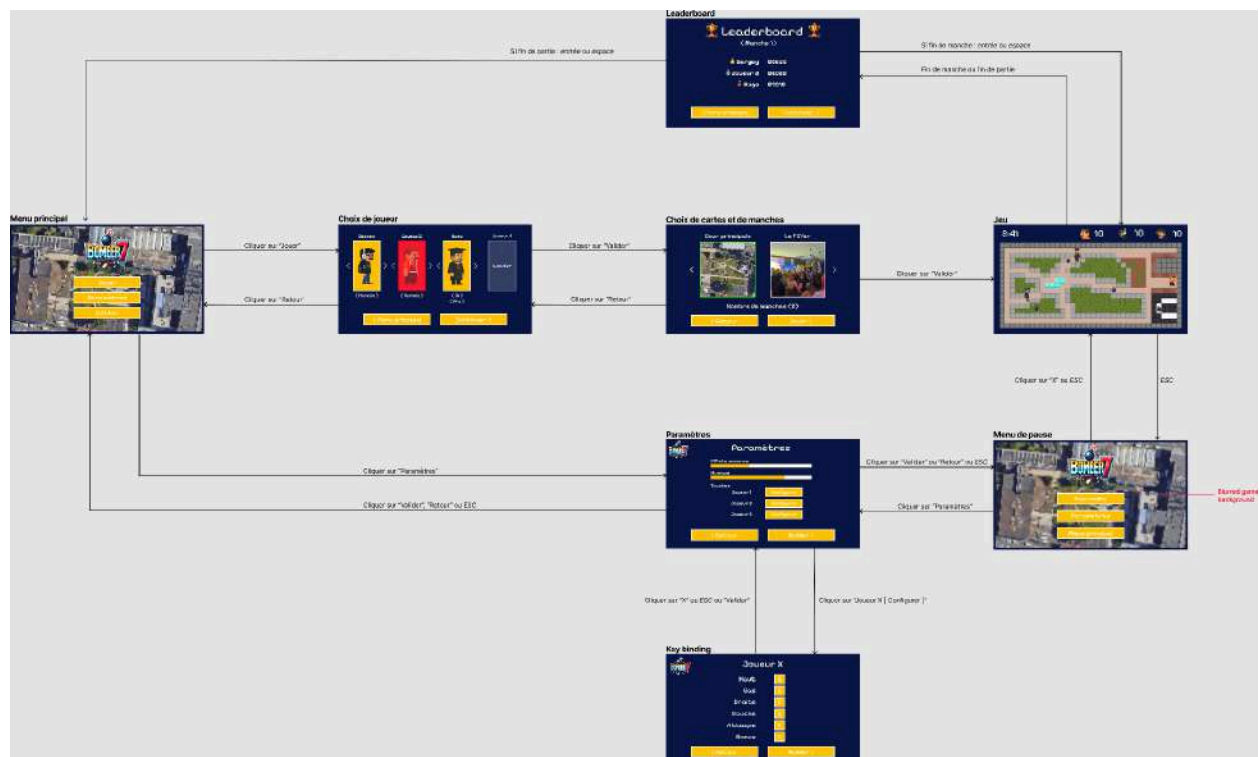
## 2.3. Fonctionnalités qui font la différence

- **Cartes (et objets) avec pour thème l'ENSEEIH** :
  - Différentes cartes seront jouables :
    - Cour principale de l'N7 (Churros)
    - Le Foyer Étudiant (Foy)
    - Restaurant universitaire (Crous)
    - Salle des diplômes (Halle aux grains)
  - Les cartes se débloquent dans l'ordre, par niveau.

### 3. Interfaces utilisateurs (maquette)

Dans le but de produire une visualisation de l'interface utilisateur de bonne qualité pour notre projet, il a été décidé d'adopter l'outil Figma. Fonctionnant sur le même principe que Canva, Il est possible de consulter la vue globale directement sur le lien suivant : [Figma](#).

#### 3.1. Vue globale



#### 3.2. Menu principal



Depuis ce menu, l'utilisateur peut :

- Lancer une partie
- Aller dans paramètres
- Quitter le jeu

### 3.3. Lancement d'une partie

#### 3.3.1. *Choix des personnages*



Depuis ce menu, l'utilisateur peut :

- Continuer vers le choix de carte
- Retourner au menu principal

#### 3.3.2. *Choix de la carte et du nombre des manches*



Depuis ce menu, l'utilisateur peut :

- Lancer une partie
- Retourner au choix des personnages

### 3.4. Partie de Bomber7



Ici, l'utilisateur peut :

- Jouer
- Aller dans le menu de pause

### 3.5. Autres

#### 3.5.1. *Menu pause*



Depuis ce menu, l'utilisateur peut:

- Reprendre le jeu
- Aller dans paramètres
- Retourner au menu principal

#### 3.5.2. *Paramètres*



Depuis ce menu, l'utilisateur peut:

- Paramétrer le volume
- Paramétrer les touches des joueurs
- Retourner au menu de pause

#### 3.5.3. *Configuration des touches*



Depuis ce menu, l'utilisateur peut:

- Paramétrer les touches des joueurs
- Retourner aux paramètres



## 4. Cas d'usage

La quatrième partie de ce rapport permet de mettre en perspective deux cas d'usages afin de comprendre avec un exemple les mécaniques de jeu. Nous détaillerons donc deux cas d'usage : l'un simulant une partie en solo, et un second simulant une partie en multijoueur.

### 4.1. Jeu solo

- Menu principal

Description : L'utilisateur a la possibilité de commencer à initier une partie, à modifier les paramètres du jeu ou de quitter l'application.

Déclencheur : L'utilisateur décide de modifier les paramètres des touches de jeu. Il clique sur le bouton Paramètres.

Étape : -

Résultat : L'utilisateur passe dans le menu Paramètres.

- Menu paramètres

Description : L'utilisateur modifie la puissance des effets sonores et de la musique. Il clique sur les paramètres du joueur n°1 (lui-même).

Déclencheur : La sélection de "paramètres" depuis le menu principal ou bien au sein de l'écran de pause après avoir pressé *echap*.

Étape : -

Résultat : Les puissances sont modifiées en conséquence.

- Paramétrage des touches

Description : L'utilisateur souhaite modifier les touches qui lui permettent d'effectuer des actions dans le jeu (se déplacer, poser une bombe.)

Déclencheur : La sélection de "configuration" à côté du joueur concerné.

Étape : -

Résultat : L'utilisateur peut ensuite bénéficier de son nouveau réglage une fois en jeu.

- Choix de joueur

Description : Ce menu permet de choisir le nombre de joueurs, leur apparence, ainsi que leur type et leur nom. Un joueur peut être humain ou un robot de difficulté variable : débutant, intermédiaire, expert.

Déclencheur : Le menu s'affiche suite à la sélection de "jouer" depuis le menu principal.

Étape : -

Résultat : Suite à la validation des paramètres, l'utilisateur passe au menu de sélection des cartes et du nombre de manches.

- Choix de cartes et de manches

Description : Ce menu permet de choisir la carte sur laquelle jouer ainsi que le nombre de manches à effectuer. Suite au choix, l'utilisateur peut poursuivre ou bien revenir en arrière sur le menu choix de joueur.

Déclencheur : Différents choix s'offrent à l'utilisateur.

Étape : -

Résultat : Après sélection, l'utilisateur se voit diriger vers un début de partie dans la carte qu'il a sélectionnée.

- Jeu : exemple d'interactions pendant la partie

### 1. Déplacements sur la carte

Description : Le joueur peut se déplacer à partir de sa zone d'apparition au sein de cases vides. Il est contraint dans ses mouvements par des blocs : cassables ou non.

Déclencheur : Commencement d'une partie et lors de son déroulé.

Étapes : -

Résultat : Le joueur se déplace.

### 2. Destruction des obstacles pour progresser

Description : Le joueur doit poser des bombes pour détruire des murs et obstacles. Ces actions permettront au joueur de se déplacer sur la carte à la recherche d'ennemis à vaincre.

Déclencheur : Arrivée dans une zone obstruée par des obstacles destructibles (ex : caisses)

Étapes :

1. Le joueur se déplace près d'un obstacle.
2. Il pose une bombe sur une case vide juxtaposée.
3. Il s'éloigne pour éviter l'explosion d'un rayon donné.
4. L'explosion détruit l'obstacle révélant un passage ou un "PowerUp".

Résultat : Accès à de nouvelles zones ou récompenses.

### 3. Neutralisation des ennemis

Description : Des ennemis patrouillent et attaquent le joueur. Il doit les éliminer.

Déclencheur : Rencontre avec un ennemi dans le niveau.

Étapes :

1. Le joueur esquivé les mouvements de l'ennemi.
2. Il place une bombe sur le trajet de l'ennemi qui paraît le plus prévisible.
3. L'explosion touche l'ennemi.

Résultat : L'ennemi est potentiellement vaincu selon sa position par rapport à la bombe et son nombre de vies.

#### 4. Collecte de bonus

Description: Des bonus (PowerUps) sont disséminés sur la carte ou lâchés par des ennemis. Ils améliorent les capacités du joueur (portée des bombes, vitesse, nombre de bombes...)

Déclencheur : Découverte d'un bonus et passage sûr ce dernier.

Étapes :

1. Le joueur repère le bonus ;
2. Il se dirige vers sa case pour le ramasser ;
3. Le bonus est appliqué, les capacités du joueur sont augmentées.

Résultat : Le joueur, suite à son déplacement, récupère un objet : ses capacités sont donc altérées.

#### 5. Élimination d'un joueur (robot)

Description: L'utilisateur rencontre un joueur. Le but du jeu étant d'éliminer ses adversaires, il va chercher à prendre ce dernier pour cible.

Déclencheur : Rencontre d'un joueur

Étapes :

1. Les joueurs se rencontrent, ils vont chercher à s'éliminer
2. Chacun place des bombes de sorte à coincer son adversaire et l'éliminer tout en essayant de ne pas se saboter soi-même

Résultat : L'utilisateur élimine le joueur ou réciproquement.

#### 6. Finir une manche

Description: Seul un joueur reste en vie, la manche se termine suite à l'élimination de l'avant-dernier joueur ; on passe à l'écran affichant le classement avant de passer à la manche suivante ou de terminer la partie.

Déclencheur : Éliminer tous ses adversaires.

Étapes : -

Résultat : La manche est terminée, on passe à l'écran des scores.

#### 4.2. Jeu multijoueur local

*Il convient de mentionner que le mode multijoueur local reprend les mêmes cas d'usage et fonctionnalités que la partie en mode solo. Effectivement, seule la dynamique de jeu évolue sur la même base de fonctionnalité, car la stratégie adoptée pour vaincre ses amis devient centrale. L'objectif demeure identique, être le dernier survivant sur la carte en éliminant l'ensemble des joueurs, qui, pour certains, sont contrôlés par des humains.*

## 5. Annexes - Gestion de projet

### 5.1. Phase de définition

#### 5.1.1. *Matrice SWOT*

La matrice SWOT (strengths, weaknesses, opportunities et threats) de notre projet a pour objectif de valider l'étape de faisabilité de notre projet, ainsi que d'identifier les facteurs externes pouvant l'affecter.

| Origine Interne   |   | Origine Externe   |  |
|---|---|---|--|
| Forces  | Faiblesses  | Opportunités  | Menaces  |
| Connaissance du langage Java par tous les membres du groupe   | Niveaux différents en programmation au sein du groupe : nécessité de monter en compétence pour les BUT RT | Projet au sujet amusant, favorisant donc l'envie de s'investir dans ce projet           | Faible volume de temps pour réaliser le projet (33 heures)   |
| Bonne entraide, pédagogie et dynamique de travail dans le groupe  | Manque de connaissances en lien avec la méthode AGILE   | Utilisation des ressources de la bibliothèque et de l'aide enseignante                  | Les cours de Gestion de projet sur la méthode SCRUM n'ont pas pu être dispensés                                  |
| Complémentarité des connaissances de chaque membre : graphisme, gestion de projet, développement logiciel | /   | Développement en équipe de taille moyenne permettant de s'initier au <i>peer-coding</i> | Les autres matières et partiels peuvent prendre le pas sur l'investissement des membres du groupe dans le projet |
| Autant d'effectifs provenant de BUT INFO que de BUT RT  | /   | /   | /  |

## 5.2. Phase de montage

### 5.2.1. Matrice RACI

Notre RACI permet d'identifier les rôles et responsabilités de chaque membre de l'équipe en charge de la réalisation d'un axe principal de notre projet. Cette matrice nous permet de nous répartir les tâches et de tirer profit des compétences de chacun.

**R : Responsable - A : Accountable - C : Consulted - I : Informed**

| Missions   | Yoann François | Corentin Pradier | Stéphane Loppinet | Thomas Silvestre | Nikita Ziuzin | Pierre Chaveroux |
|--|----------------|------------------|-------------------|------------------|---------------|------------------|
| Conception de l'architecture logicielle                          | C              | R                | C                 | RA               | C             | C                |
| Back-End : Développement des mécanismes de jeu                   | C              | C                | R                 | R                | RA            | R                |
| Back-End : Développement des mécanismes de joueur humain         | R              | RA               | C                 | I                | R             | C                |
| Back-End : Développement des mécanismes des joueurs IA           | C              | C                | RA                | R                | C             | R                |
| Back-End : Développement des PowerUps                            | R              | C                | R                 | R                | C             | RA               |
| Front-End : développement des cartes du jeu                      | R              | R                | I                 | I                | RA            | C                |
| Front-End : développement des menus et liens Front-End, Back-End | RA             | I                | C                 | C                | R             | I                |
| Test et validation des livrables de code                         | I              | C                | RA                | R                | I             | R                |
| Respect des jalons et bonnes pratiques de gestion de projet      | R              | C                | C                 | C                | C             | RA               |

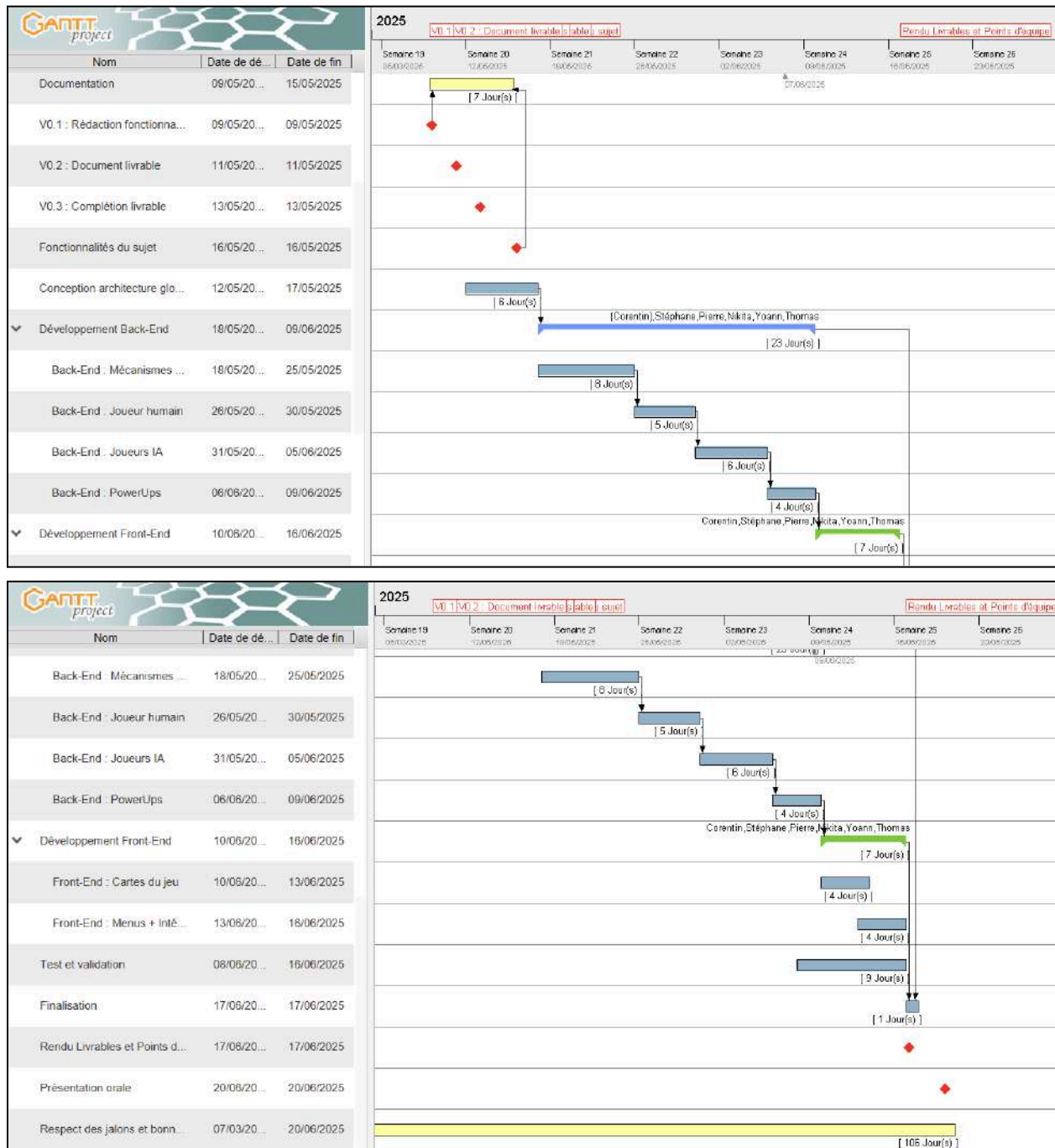
### 5.2.2. Liste des livrables

Cette liste des livrables permet d'avoir une idée claire des jalons à fixer dans le GANTT. Ce tableau vient donc en complément du diagramme.

| Livrables           | Type  | Objectif  |
|---------------------|---|---|
| sujets.pdf          | Document de présentation avec quatre sujets de projet envisagés à faire valider par l'enseignant référent   | Prioriser un sujet afin de le développer, le but étant de convaincre le 'client'  |
| fonctionnalites.pdf | Document condensant fonctionnalités, schémas des interfaces utilisateurs et planification du projet suivant les principes des méthodes agiles   | Rendre-compte des premières idées et de l'ensemble des fonctionnalités de notre projet  |
| rapport.txt         | Rapport regroupant l'ensemble des livrables et explications, qu'ils soient d'ordre technique ou de gestion de projet. Le code source de l'application sera joint à ce document via Hudson | Étayer au maximum ce qui a été réalisé durant le projet (technique et gestion de projet). Valider le cahier de fonctionnalités rédigées |
| utilisateur.pdf     | Manuel utilisateur de l'utilisation du jeu Bomber7  | Rendre accessible le jeu au plus grand nombre en décrivant les actions possibles en jeu   |
| notes-equipiers.txt | Document d'évaluation   | Noter des coéquipiers en fonction de leur investissement dans le projet   |
| presentation.pdf    | Support visuel de type PowerPoint   | Dynamiser et appuyer nos propos lors de la présentation orale   |

### 5.2.3. Diagramme de GANTT

Dans un souci de clarté, le diagramme de GANTT prévisionnel prévu pour ce projet se limite aux points essentiels de ce dernier. Ce diagramme ne comporte donc que les jalons ayant lieu lors de notre dernière période de cours : de la semaine 20 à la semaine 26.



### 5.3. Analyse des risques

#### 5.3.1. *Matrice des risques*

Cette matrice a pour objectif de cerner les points difficiles qu'il est probable de rencontrer au cours de ce projet. Le responsable identifié pour chaque risque doit veiller à tenir ce dernier à distance. S'il ne peut être évité, une réunion de crise doit être organisée.

| Description              | Impact | Probabilité | Criticité | Resp             | Prévention  |
|--------------------------|--------|-------------|-----------|------------------|---|
| Faible volume horaire    | 2      | 2           | 4         | Yoann François   | Être efficace sur les créneaux dédiés                 |
| Cours SCRUM annulés      | 3      | 4           | 12        | Pierre Chaveroux | S'autoformer en ligne                                 |
| Lacunes en Java          | 4      | 1           | 4         | Thomas Silvestre | Compréhension des choix techniques par toute l'équipe |
| Charge de travail élevée | 3      | 3           | 9         | Nikita Ziuzin    | Lisser la charge de travail sur la période            |

#### 5.3.2. *Matrice des objectifs*

À date d'édition de ce document, nos objectifs pour ce projet sont les suivants (possible évolution après chaque itération de développement agile en début de chaque Backlog):

| Réussite     | Bomber7                                | Rapport   | Oral                                      | Gestion Projet                                    |
|--------------|--|---|---|---|
| Insuffisante | Jeu non fonctionnel                    | Choix techniques non justifiés                              | Non-respect du CDC de l'oral              | Livrables non rendus                              |
| Acceptable   | Logique de conception respectée        | Choix techniques et de gestion de projet justifiés          | Temps de parole et CDC de l'oral respecté | Motivation d'équipe constante et livrables rendus |
| Bonne        | MVC respecté et fonctionnel            | Validation des tests des fonctionnalités                    | Démo -> Pres -> Organigramme              | Validation complète du CDC                        |
| Excellente   | Fonctionnalités complètes + IHM fidèle | Mise en conformité du rapport avec les standards du domaine | Réponses correctes aux questions du jury  | Client satisfait des livrables                    |