

Plan de développement

Groupe 1 Agora V3-1 : Partie bibliothèque de jeux

12 mars 2024

Version	5
Date	12 mars 2024
Rédigé par	BINGINOT Etienne CAUCHOIS Niels DUCROQ Yohann KHABOURI Izana MAZUY Axelle MONTAGNE Erwann THIBERVILLE Malvina VAN LIEDEKERKE Florian

Mises à jour du document

Version	Date	Modification réalisée
1	08 décembre 2023	Création du document
2	22 décembre 2023	Modification des rôles, calcul des charges et redéfinition des échéances de sprint
3	24 janvier 2024	Mise à jour après les retours de l'enseignant de gestion de projet
4	25 février 2024	Mise à jour des responsabilités après le sprint 2 Mise à jour du tableau des tâches
5	12 mars 2024	Mise à jour des responsabilités après le sprint 3 Mise à jour du tableau des tâches
6	26 mars 2024	Mise à jour des responsabilités après le sprint 4 et 5 Mise à jour du tableau des tâches
7	15 avril 2024	Mise a jour du tableau des tâches après le sprint 6

Table des matières

1	Contexte du projet	4
1.1	Contexte de développement	4
1.2	Objet du projet	4
1.3	Principaux acteurs	4
1.4	Objectifs poursuivis	4
1.5	Documents de références	4
2	Méthodologie de développement	5
3	Organisation et responsabilités	6
4	Organigramme des tâches	8
5	Evaluation du projet et dimensionnement des moyens	9
5.1	Evaluation globale de la charge et répartition par phase	9
5.2	Besoins en moyens et ressources	9
6	Planning général	9
6.1	Organisation des sprints	9
6.2	Détail des tâches et répartition	11
6.2.1	6 qui prend	11
6.2.2	Splendor	12
6.2.3	Glenmore	13
6.2.4	Myrmes	14
6.2.5	Myrmes	15
6.3	Diagramme de Gantt	17
7	Procédés de gestion	18
7.1	Gestion de suivi des tâches	18
7.2	Gestion de la documentation	18
7.3	Gestion des configurations	18
8	Revue et points clefs	18
8.1	Objectifs	18
8.2	Dates clés	18
9	Procédure de suivi d'avancement	19
9.1	Suivi interne	19
9.2	Compte-rendus d'avancement externe	19
9.3	Réunions prévues	19

1 Contexte du projet

1.1 Contexte de développement

Nous réalisons ce projet dans le cadre de notre formation en Master Informatique, Génie de l'Informatique Logicielle à l'université de Rouen Normandie. Ce projet constitue le module "Application Informatique" se déroulant tout au long de l'année, dans lequel les étudiants doivent réaliser une application informatique par groupes d'environ 6 étudiants.

1.2 Objet du projet

Dans le cadre du projet annuel, nous avons pour mission de développer quatre jeux sur une nouvelle plateforme en cours de conception. Cette plateforme permet aux différents joueurs de jouer à plusieurs jeux de plateau sur un site en ligne hébergé par l'université.

Le projet AGORA a débuté en 2010. Chaque année, un groupe d'étudiants s'occupait d'implémenter de nouveaux jeux à la plateforme. Le projet se voyant vieillissant de par son architecture et de ses technologies, il est donc nécessaire de refondre la plateforme pour répondre aux normes actuelles.

1.3 Principaux acteurs

Voici la liste des parties prenantes externes à l'équipe du projet :

- Client du projet : Pascal CARON, enseignant-chercheur au département Informatique de l'université Rouen
- Référent technique serveur : Bruno MACADRE, administrateur systèmes et réseaux à l'université de Rouen
- Référent technique Symfony : Stéphane BURESI, chef de projet technique chez Proxiad Axe Seine
- Expert gestion de projet : Karim ABDELLAH GODARD, Directeur du Centre de Services chez Proxiad Axe Seine

L'équipe aux commandes du projet est composée de :

- BINGINOT Etienne
- CAUCHOIS Niels
- DUCROQ Yohann
- KHABOURI Izana
- MAZUY Axelle
- MONTAGNE Erwann
- THIBERVILLE Malvina
- VAN LIEDEKERKE Florian

1.4 Objectifs poursuivis

Il est attendu du projet de développer une bibliothèque de code pour l'implémentation des jeux fonctionnelle, qui permette notamment l'implémentation de 4 jeux : le 6 qui prend, le Splendor, Glenmore et Myrmes.

L'objectif principal est donc d'assurer à l'utilisateur la jouabilité de ces quatre jeux, tout en assurant aux prochains intervenants sur ce projet une prise en main rapide du code.

1.5 Documents de références

Voici les listes des documents de références :

- Spécification Technique des Besoins :
 - **STB_GENERAL**
 - **STB_6QUIPREND**
 - **STB_GLENMORE**

- **STB_SPLENDOR**
- **STB_MYRMES**
- Document d'Architecture Logicelle :
 - **DAL_GENERAL**
 - **DAL_6QUIPREND**
 - **DAL_GLENMORE**
 - **DAL_SPLENDOR**
 - **DAL_MYRMES**
- Cahier Des Recettes :
 - **CDR_GENERAL**
 - **CDR_6QUIPREND**
 - **CDR_GLENMORE**
 - **CDR_SPLENDOR**
 - **CDR_MYRMES**
- Analyse Des Risques :
 - **ADR_PROJET_AGORA**
- Ensemble des documents d'archives des anciennes versions d'AGORA

2 Méthodologie de développement

Dans le cadre de ce projet, nous avons choisi d'employer la méthodologie Agile SCRUM. Cette méthodologie nous permettant d'avoir une plus grande efficacité, des livrables réguliers tout en ayant un retour client assuré.

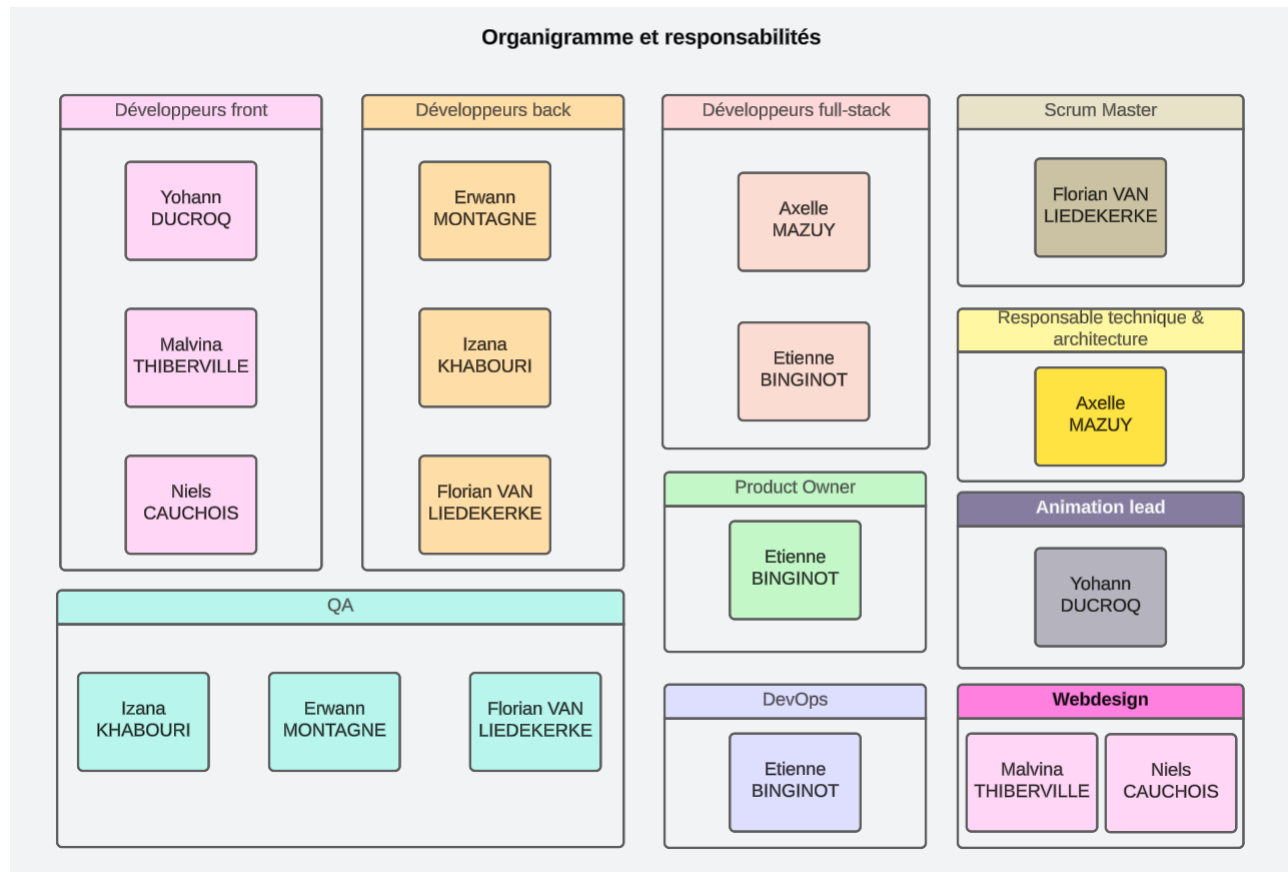
Ce processus fonctionne par itération que l'on appelle sprint. Chaque sprint comporte le même fonctionnement :

- **Organisation et planification** : Cette phase permet l'évaluation des tâches à réaliser durant le sprint et leur attribution aux différents membres de l'équipe.
- **Phase de développement** : Cette phase consiste en la réalisation des tâches définies précédemment, leurs tests et suivi. Cette phase est également ponctuée de réunions quotidiennes concernant l'avancement du projet, mais aussi d'un tableau de tâches permettant le suivi de la réalisation de celles-ci.
- **Revue de sprint et livraison** : C'est à cette phase que la partie du produit finie est rendue au client, mais aussi qu'une revue de l'itération passée est effectuée, permettant d'analyser les retours du client, les points à améliorer au prochain sprint concernant le code et le contenu pressenti pour le prochain sprint.
- **Rétrospective** : Cette phase consiste davantage à une réunion d'équipe, tenue entre la fin d'un sprint et le début d'un autre, durant lequel l'équipe fait un retour sur ce qu'il s'est produit durant le sprint d'un point de vue expérience acquise, les points à améliorer concernant notamment la communication ou la planification, et permettre une amélioration des performances dans les prochains sprints.

Nos sprints seront d'une durée de 2 semaines, à la fin desquels seront rendus au client des livrables. Ces livrables seront majoritairement des fonctionnalités de l'application attendue concernant la partie bibliothèque de jeux. On pourra également retrouver de la documentation concernant les jeux ou encore la façon d'implémenter un jeu, qui seront agrémentées au fil du développement.

L'objectif est de réussir à produire la base d'un jeu fonctionnel en un à deux sprints, la preuve de concept étant sur le jeu 6 qui prend, la première itération commencera donc par terminer son développement.

3 Organisation et responsabilités



Présentation des rôles

Dans l'organigramme précédents, on peut retrouver les rôles suivants pour les intervenants internes :

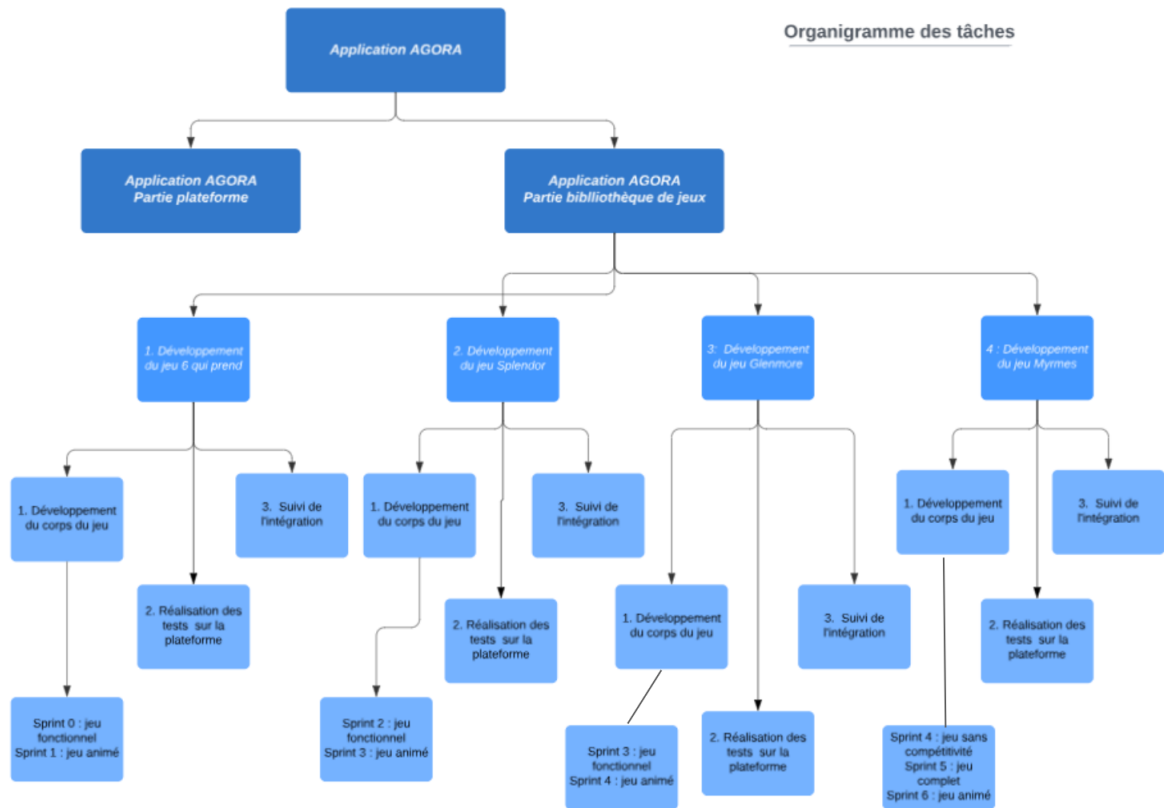
- **Développeur** : Chaque membre de l'équipe est développeur. Le développeur a pour tâche de développer les fonctionnalités en faisant attention à respecter une qualité de code minimale.
- **Développeur back** : Il s'agit d'un développeur s'occupant de la partie métier du code.
- **Développeur front** : Il s'agit d'un développeur s'occupant de la partie visuelle de l'application.
- **Développeur full-stack** : Il s'agit d'un développeur s'occupant de la partie visuelle de l'application autant que la partie métier du code. Ces développeurs permettent également de faire le lien entre la vue et la partie métier.
- **QA** : Il s'agit d'un développeur s'assurant du bon fonctionnement de l'application. Pour cela il va appliquer les tests et vérifier leur bon fonctionnement, tout en s'assurant de la couverture de tests réalisée par les développeurs.
- **DevOps** : Il s'agit de la personne en charge du déploiement du projet sur chacune des machines des développeurs en s'assurant de l'automatisation du code et des tests.
- **Product Owner** : Il est en charge de la communication entre l'équipe de développement et le client, il s'assure de la tenue de réunions et interprète les besoins du client.

- **Responsable architecture** : Il s'agit de la personne vérifiant la cohérence entre l'architecture définie dans la DAL et l'implémentation du code.
- **Responsable technique** : Il s'agit de la personne s'assurant de la bonne utilisation des technologies au sein du projet.
- **Scrum Master** : Il s'agit de la personne s'assurant de la tenue des échéances et de la bonne organisation au sein de l'équipe.
- **Webdesign** : Il s'agit de la personne en charge de la réalisation des maquettes de vue avant le développement du code.
- **Animation lead** : Il s'agit de la personne en charge de l'architecture des animations et de leur implantation. C'est le référent en matière d'animation, qui permet de donner vie à un jeu et peut faciliter la compréhension des utilisateurs.

Dans notre organisation, on retrouve également des intervenants externes. Ces intervenants ne font pas partie de l'organigramme mais sont essentiels au bon fonctionnement du projet :

- **Référent technique** : Il s'agit d'une personne ayant des connaissances techniques et une certaine expérience dans un domaine ou une technologie. Dans notre cas, nous avons deux types de référents techniques : un référent technique concernant Symfony, et concernant les serveurs.
- **Client** : Il s'agit de la personne exprimant un besoin à l'équipe du projet. Maintenir une bonne communication avec le client permet de répondre à sa demande et de satisfaire le besoin exprimé.

4 Organigramme des tâches



Un organigramme des tâches permet de lister les tâches nécessaires à l'aboutissement d'un produit. Dans notre cas, le produit final est l'application Agora. Toutefois, étant deux groupes sur le projet, notre produit final d'équipe constituera la bibliothèque de jeux de l'application.

On peut voir figurer dans l'organigramme des tâches les grandes tâches nécessaires, à l'aboutissement du projet.

Après la conception des documents, nous commençons la phase de développement, et donc le début de la méthode Agile SCRUM. Les grandes étapes de développement des jeux avec cette méthode sont donc le développement du corps des jeux. Dans l'ordre, le 6 qui prend sera réalisé en version fonctionnelle donc sans animation lors de la POC (sprint 0) - preuve de concept - et finalisée avec les animations et l'ensemble de la bibliothèque et des fonctionnalités comme le chat de discussion au sprint 1.

Nous développerons ensuite le jeu du Splendor au sprint 2 dans sa version fonctionnelle (donc toujours sans les animations) et elle sera finalisée au sprint 3.

Nous rendrons également une version fonctionnelle du jeu Glenmore au 3ème sprint, et ce jeu sera animé au sprint 4.

Lors du sprint 4, nous rendrons une version non compétitive du jeu Myrmes (c'est à dire que les fonctionnalités de base du jeu seront présentes, mais les actions permettant d'aboutir à un système de point et de victoire ne seront pas encore présents). Au sprint 5, le jeu sera complété, donc on retrouvera notamment un système de points et de victoire, et le sprint 6 marquera la fin du jeu Myrmes avec son animation.

Ces jeux sont aussitôt testés via la CI/CD (les tests unitaires étant réalisés dès la phase du développement du corps des jeux par les développeurs), puis une phase de suivi de l'implémentation qui constitue davantage une vérification du maintien des jeux lors du développements des autres jeux, observer s'il existe des bugs et les réparer, ainsi que d'adapter en fonction du retour client.

Il est à noter que la documentation demandée sera rédigée au cours du développement, alimentée par notre expérience.

5 Evaluation du projet et dimensionnement des moyens

5.1 Evaluation globale de la charge et répartition par phase

L'équipe en charge de la bibliothèque des jeux est composée de 8 membres pouvant travailler chacun sur une période de 10 jours pour 2 semaines - 2 semaines correspondant à un sprint - à raison d'une moyenne d'une heure par jour.

Nous avons donc une capacité de 80 jours/Homme par cycle. Etant donné que l'on compte réaliser 6 sprints, nous avons donc 480 jours/Homme pour l'ensemble de la phase développement.

Si l'on additionne la charge évaluée pour la phase de développement selon les tâches, on a alors une charge de travail de 387 jours / Homme, pour 480 jours / Homme de capacité.

La charge évaluée pour les jeux est la suivante :

- 6 qui prend : 85 jours/Homme ;
- Splendor : 59 jours/Homme ;
- Glenmore : 70 jours/Homme ;
- Myrmes : 173 jours/Homme.

La charge moyenne d'un sprint est d'environ 55 jours/Homme.

5.2 Besoins en moyens et ressources

Afin de réaliser ce projet, nous aurons besoin d'un serveur de développement pour le développement des jeux. Ce serveur sera commun avec l'autre équipe Agora afin d'assurer le plus tôt possible le bon fonctionnement de l'application. Nous aurons également un serveur de pré-production et de production, également en commun. Le serveur de pré-production aura des configurations similaires au serveur de production afin de tester en conditions réelles l'application développée.

Nous aurons également un serveur SonarQube pour la réalisation des tests et assurer la CI/CD.

6 Planning général

6.1 Organisation des sprints

Le client nous a demandé de lui fournir 4 jeux dans un ordre de difficulté croissante. Le premier jeu, 6 qui prend, faisant l'objet de notre preuve de concept sera donc réalisé de manière fonctionnelle avant le premier sprint. Le quatrième jeu étant conséquent, il est nécessaire de lui réserver au minimum 2 sprints, c'est pourquoi nous avons prévu ces livrables pour chaque sprint :

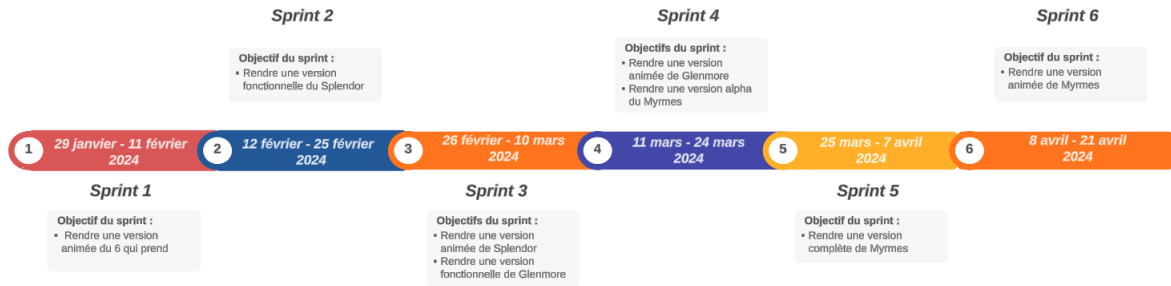
- Sprint 1 : Version animée du 6 qui prend ;
- Sprint 2 : Version fonctionnelle de Splendor ;
- Sprint 3 : Version animée de Splendor et version fonctionnelle de Glenmore ;
- Sprint 4 : Version animée de Glenmore et Myrmes en version alpha ;
- Sprint 5 : Version complète de Myrmes ;
- Sprint 6 : Version animée de Myrmes.

Une version alpha signifie que l'on a une version de jeu sans compétitivité, il n'y a pas de système de classement et points.

Une version complète correspond à une version intégrant la compétitivité, et donc un classement et un système de point, ainsi que les moyens de victoire (les dernières phases).

Voici un schéma général de l'organisation des sprints :

Méthodologie de développement



6.2 Détail des tâches et répartition

6.2.1 6 qui prend

Sprint	Code	Tâche	Durée (jours)	Charge (j/h)	Responsables
S0	US01	Démarrage de la partie	1	1	Etienne
S0	US02	Démarrage de la manche	1	1	Etienne
S0	US03	Choix d'une carte à poser par l'utilisateur	3	12	Axelle, Niels, Florian, Etienne
S0	US04	Traitement des cartes des joueurs	4	12	Etienne, Malvina, Erwann
S0	US05	Classement des têtes de boeuf	3	9	Yohann, Izana, Erwann
S0	US06	Fin d'une manche	3	6	Yohann, Florian

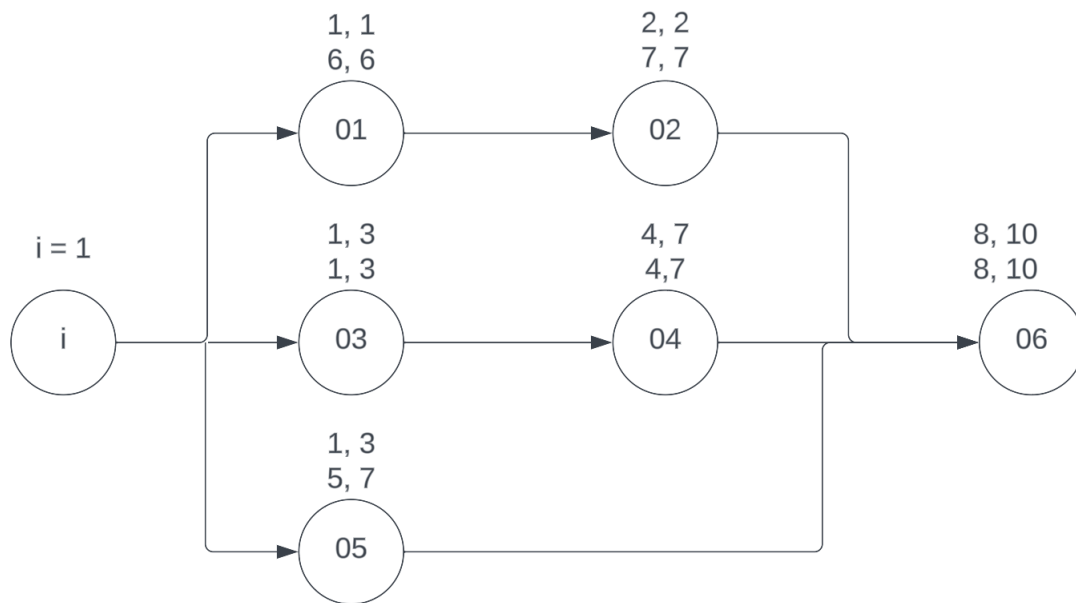


FIGURE 1 – Diagramme de PERT du sprint 0 - POC, 6 qui prend en version fonctionnelle

Sprint	Code	Tâche	Durée (jours)	Charge (j/h)	Responsables
S1	US07	Gestion des spectateurs	5	10	Axelle, Etienne
S1	US08	Animation de dépôt des cartes et classement	6	24	Yohann, Niels Malvina, Etienne
S1	US09	Intégration du chat de discussion des joueurs	3	9	Izana, Erwann, Niels
S1	US10	Intégration à la plateforme	1	1	Etienne
S1	US11	Modifications sur retour client	2	4	Non-assigné
S1	US12	Gestion des logs	4	4	Florian

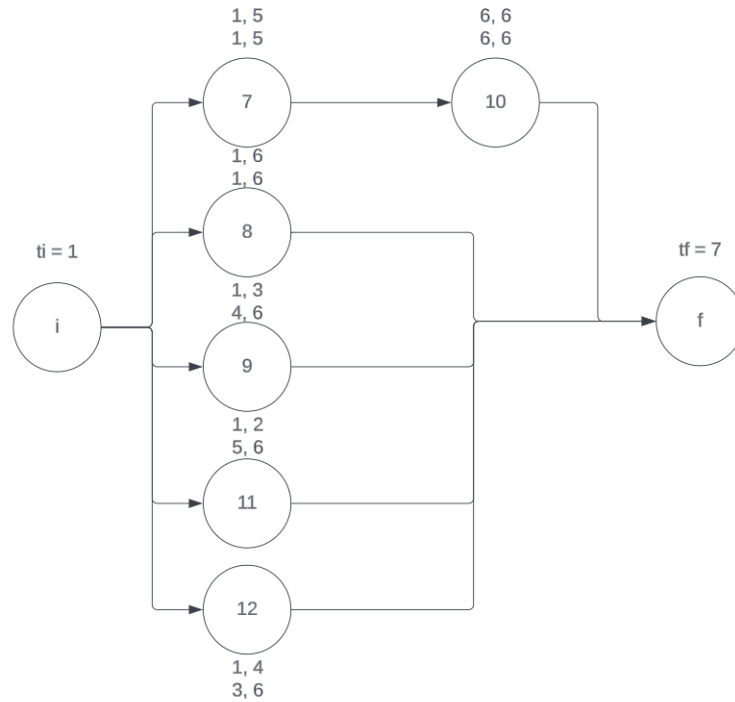


FIGURE 2 – Diagramme de PERT du sprint 1 - 6 qui prend en version animée et fonctionnalités de la bibliothèque commune

6.2.2 Splendor

Sprint	Code	Tâche	Durée (jours)	Charge (j/h)	Responsables
S2	US13	Démarrage de la partie	1	1	Etienne
S2	US14	Choix de jetons par le joueur	5	20	Florian, Niels, Axelle, Erwann
S2	US15	Achat d'une carte de développement par le joueur	3	9	Etienne, Malvina, Yohann
S2	US16	Réservation d'une carte de développement par le joueur	2	6	Yohann, Malvina Izana
S2	US17	Fin d'une partie	3	6	Niels, Florian
S2	US18	Classement	2	4	Erwann, Niels
S2	US19	Intégration dans la plateforme	1	1	Etienne
S2	US20	Modifications sur retour client	3	6	Non-assigné

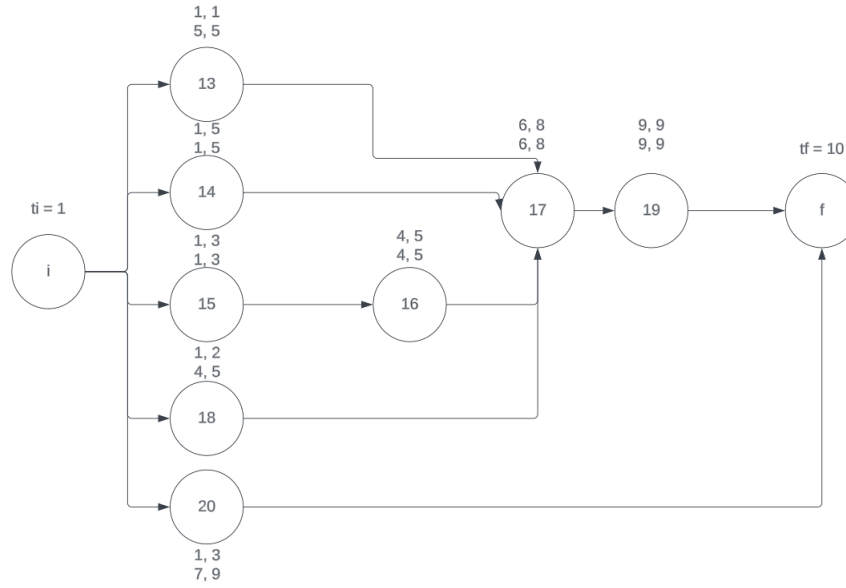


FIGURE 3 – Diagramme de PERT du sprint 2 - Splendor en version fonctionnelle

6.2.3 Glenmore

Sprint	Code	Tâche	Durée (jours)	Charge (j/h)	Responsables
S3	US21	Animation des jetons, cartes, classement de Splendor	5	10	Yohann, Malvina
S3	US22	Démarrage de la partie	2	4	Etienne, Axelle
S3	US23	Gestion des manches et du classement	4	12	Florian, Erwann, Malvina
S3	US24	Déplacement sur une tuile Placement de la tuile	3	12	Etienne, Izana Niels, Erwann
S3	US25	Choix d'activation des tuiles	3	9	Florian, Yohann, Erwann
S3	US26	Gestion des ressources	3	9	Izana, Etienne, Axelle
S3	US27	Intégration dans la plateforme	1	1	Etienne
S3	US28	Modifications sur retour client	5	10	Non-assigné

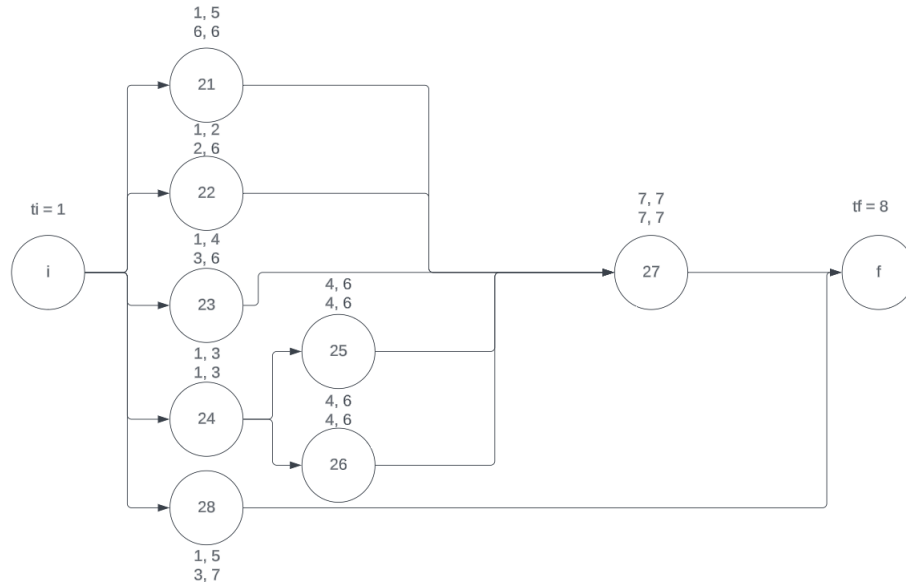


FIGURE 4 – Diagramme de PERT du sprint 3 - Splendor en version animée - Glenmore en version fonctionnelle

6.2.4 Myrmes

Sprint	Code	Tâche	Durée (jours)	Charge (j/h)	Responsables
S4	US29	Animation de déplacement des villageois et classement de Glenmore	4	8	Yohann, Malvina
S4	US30	Démarrage de la partie	4	8	Etienne, Niels
S4	US31	Phase évènement	2	4	Florian, Malvina
S4	US32	Phase nourrice	3	12	Erwann, Axelle Niels, Florian
S4	US33	Initialisation phase des ouvrière	3	6	Etienne, Niels
S4	US34	Déplacement des ouvrières	2	4	Izana, Axelle
S4	US35	Placement des objets sur le terrain	5	20	Erwann, Florian, Yohann, Malvina
S4	US36	Intégration dans la plateforme	1	1	Etienne
S4	US37	Modifications sur retour client	5	10	Non-assigné

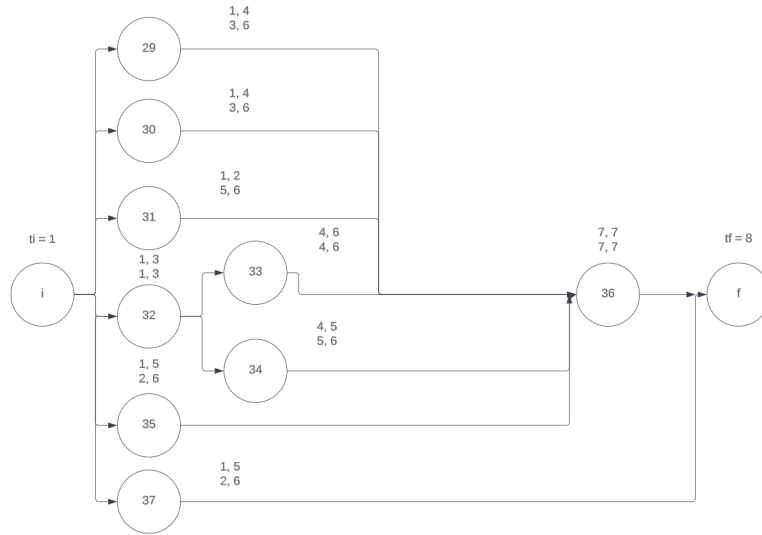


FIGURE 5 – Diagramme de PERT du sprint 4 - Glenmore en version animée - Myrmes en version alpha

6.2.5 Myrmes

Sprint	Code	Tâche	Durée (jours)	Charge (j/h)	Responsables
S5	US38	Phase récolte	2	6	Erwann, Yohann, Niels
S5	US39	Phase atelier	4	16	Florian, Izana, Niels, Axelle
S5	US40	Gestion des objectifs	5	20	Etienne, Yohann, Erwann, Malvina
S5	US41	Fin de manche Phase hiver	3	12	Florian, Izana, Axelle, Niels
S5	US42	Gestion des points	3	3	Malvina
S5	US43	Modifications sur retour client	5	10	Non-assigné

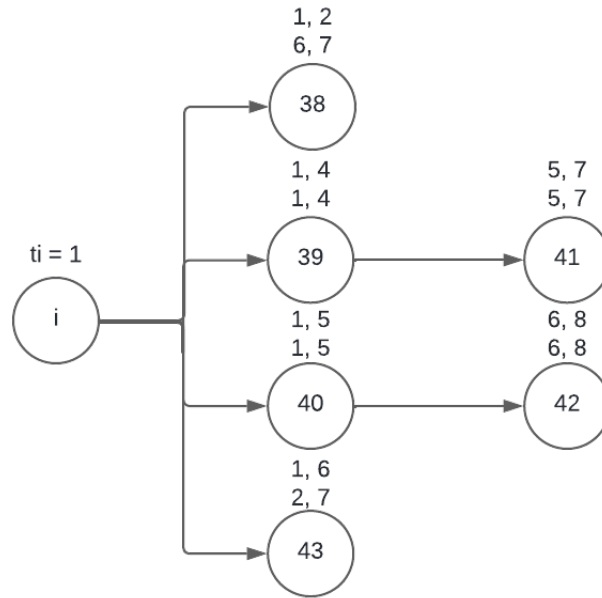


FIGURE 6 – Diagramme de PERT du sprint 5 - Myrmes en version complète

Sprint	Code	Tâche	Durée (jours)	Charge (j/h)	Responsables
S6	US45	Animations Myrmes	6	12	Yohann, Malvina
S6	US46	Gestion des objectifs	3	6	Yohann, Malvina
S6	US47	Phase ouvrière - annulation du coup	3	12	Axelle, Etienne Niels, Florian
S6	US48	Placement automatique sur les pistes de naissance	2	2	Izana
S6	US49	Ajout de la possibilité d'exclure un joueur	2	2	Axelle
S6	US50	Factorisation du code	9	36	Niels, Erwann Florian, Izana
S6	US51	Modifications sur retour client	7	14	Non-assigné

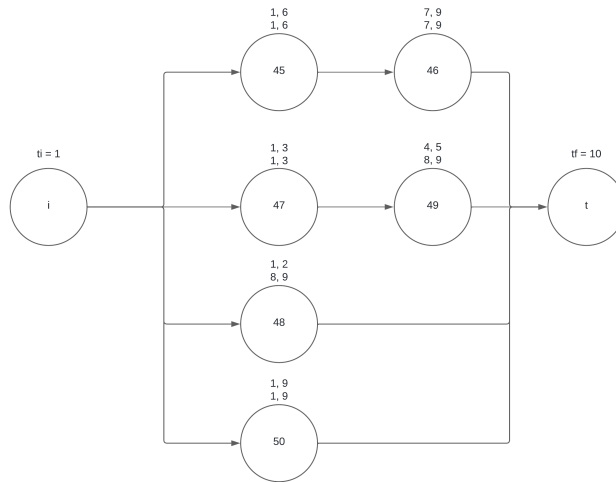


FIGURE 7 – Diagramme de PERT du sprint 6 - Myrmes en version animée

6.3 Diagramme de Gantt

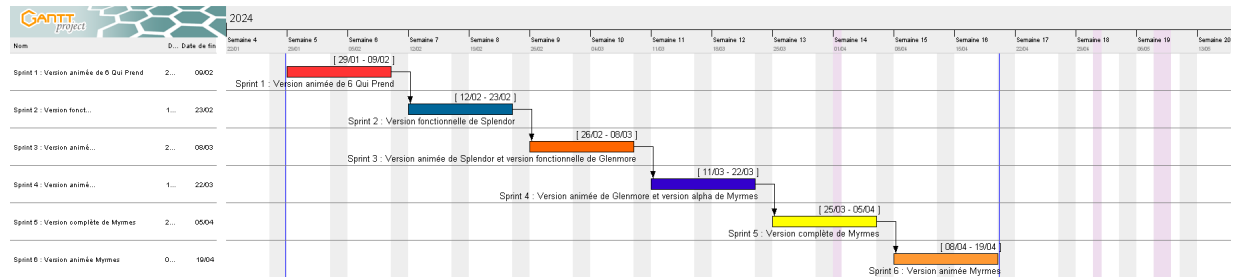


FIGURE 8 – Diagramme de Gantt

7 Procédés de gestion

7.1 Gestion de suivi des tâches

Afin de nous assurer que chaque membre du groupe soit affecté à une tâche et que le projet soit dans les temps, nous allons utiliser l'outil Jira. Cet outil nous permet de visualiser, pour un sprint, l'ensemble des tâches à effectuer avec un suivi sur les personnes et le temps nécessaires à leurs réalisations.

7.2 Gestion de la documentation

Chaque document a fait l'objet d'une analyse et une contribution de groupe.

Les documents spécifiques aux jeux (les DAL et STB) ont été tout d'abord créé d'un premier jet par des binômes en fonction des jeux :

- Splendor : Izana et Yohann
- Glenmore : Florian et Erwann
- Myrmes : Niels et Malvina
- 6 qui prend : Etienne et Axelle

Pour compléter ces analyses, chaque membre a pu expérimenter également les différents jeux à implémenter et ainsi apporter contribution sur les documents, ainsi qu'une autre perspective sur ceux-ci.

Chaque document a donc été relu, rédigé en partie et approuvé par l'ensemble de l'équipe.

7.3 Gestion des configurations

Des règles de configuration pour la sauvegarde des versions ont également été mises en oeuvre :

- Concernant les documents : Etant donné que nous avons rédigé nos documents en LaTeX, nous avons utilisé Overleaf afin de faire de la modification en temps réel et un document partagé avec l'ensemble des membres de l'équipe. Chaque version des PDF ont été archivées dans un dossier dans un Drive Google.
- Concernant le code : Etant donné que nous travaillons sur le même espace de développement que l'autre équipe Agora, il est important de bien séparer en différentes branches les versions de code. Des commits serviront notamment à mentionner clairement et avec une syntaxe précise les fonctionnalités ajoutées.

8 Revues et points clefs

8.1 Objectifs

Comme vue tout au long de ce document, l'objectif principal est d'arriver à implémenter de manière modulaire les 4 jeux demandés en fournissant une documentation détaillée du fonctionnement des modules et de la manière d'en ajouter.

8.2 Dates clés

- Début de la preuve de concept (6 qui prend) : 15 janvier 2024
- Revue de lancement : 26 janvier 2024
- Livraison d'une version animée du 6 qui prend : 11 février 2024
- Livraison d'une version fonctionnelle de Splendor : 25 février 2024
- Livraison d'une version animée de Splendor et d'une version fonctionnelle de Glenmore : 10 mars 2024
- Livraison d'une version animée de Glenmore et une version alpha de Myrmes : 24 mars 2024
- Livraison d'une version fonctionnelle de Myrmes : 7 avril 2024
- Livraison d'une version animée de Myrmes : 21 avril 2024
- Livraison d'une version complète avec retour client : 7 mai 2024

9 Procédure de suivi d'avancement

9.1 Suivi interne

L'équipe se réunit deux fois par semaine pendant 4 à 5 heures afin d'avancer l'analyse du projet, la production des documents et l'avancement des jeux.

De plus, le développement sera suivi par un outil de planification et par les commits, merges sur l'espace Git.

9.2 Compte-rendus d'avancement externe

A l'issu de chaque réunion avec les intervenants externes, des comptes-rendus sont produits et leurs sont envoyés. Cela a pour intérêt de faire un point sur la réunion qui s'est déroulée et rendre compte de ce qui a été dit pour les deux parties.

9.3 Réunions prévues

Voici ci dessous la liste des réunions prévues :

- Daily meeting : des réunions journalières permettant au membres du projet de faire part de leur avancée, leurs éventuelles difficultés et s'assurer que le temps de développement est bien respecté.
- Sprint planning : des réunion de planification de sprint, durant lesquelles les User-stories et les sous tâches sont évaluées par rapport au Plan de développement associé au sprint. Les tâches sont alors attribuées ou réattribuées, la durée peut être revue également en fonction du retour sur les sprints précédents.
- Sprint rétrospective : des réunions à chaque fin de sprint permettant de faire un retour d'expérience sur le sprint, pour être plus efficace sur le sprint suivant.
- Sprint review : des réunions avec le client en fin d'itération pour lui présenter les livrables et l'avancement réalisé, ainsi que mettre en oeuvre la planification pour l'itération suivante.