Rapport d’Analyse

1. Méthodologie

Pour ce projet, nous utiliserons la méthode dite **« en V »**.

1.1. Les différentes étapes de la méthode en V

1. Analyse des besoins et faisabilité
2. Spécification
3. Conception architecturale
4. Codage
5. Tests unitaires
6. Tests d’intégration
7. Tests de validation
8. Recette

Les tests seront réalisés **en parallèle de la programmation** afin de détecter les erreurs le plus tôt possible et d’accélérer le projet. Il s'agit d'une technique connue et largement utilisée en ingénierie et en bureau d’études pour **optimiser le temps de développement** grâce à la parallélisation des tâches.

---

2. Objectif du Projet

Le projet consiste en **une adaptation numérique du jeu Carcassonne** sur ordinateur.

---

3. Méthode utilisé

- **Méthodologie** : méthode en V

---

4. Durée du Projet

Le projet débute le **23/01/2025** et doit être rendu le **17/05/2025** à 8h, soit **environ 4 mois** de développement.

---

5. Équipe de Développement

Le projet est réalisé par :

- **Axel**
- **Théo**
- **Valentin**

---

6. Tests et Implémentation

Les tests seront effectués à l'aide du logiciel **Criterion**, qui permet :

- d'isoler les tests
- d'intercepter les sorties standard
- de calculer le **taux de couverture du code** (*coverage*)

La documentation est disponible ici : Criterion Documentation.

---

7. Exigences du Client

L'objectif est de réaliser **une adaptation en C** du jeu de société **Carcassonne** avec les spécificité suivantes :

- **Pas de tuiles « rivière » ni d'abbé.**
- **Les règles officielles du jeu fournies en PDF font loi** en cas de conflit avec d'autres versions du jeu.
- **Possibilité de jouer contre des intelligences artificielle (IA).**

---

8. Architecture du Projet

Le projet est divisé en plusieurs **blocs** :

8.1. Structures de données

- Contient toutes les structures de données essentielles au projet.
- **Dépend de :** Rien.
- **Utilisé par :** Tous les autres blocs.

8.2. Objets du jeu

- Contient la définition des objets du jeu avec leurs caractéristiques propres (ex : tuiles, meeples, etc.).
- **Dépend de :** *Structures de données*.
- **Utilisé par :** *Fonctions du jeu* et *Gestionnaire du jeu*.

8.3. Fonctions du jeu

- Contient toutes les fonctions implémentant les règles du jeu.
- **Dépend de :** *Structures de données* et *Objets du jeu*.
- **Utilisé par :** *Gestionnaire global du jeu*.

8.4. Gestionnaire global du jeu

- Coordonne l’ensemble des blocs pour faire fonctionner le jeu.
- **Dépend de :** Tous les autres blocs.
- **Utilisé par :** Rien.

---

9. Conception Détaillée

1. Structure et composition du jeu

- Le jeu est constitué de 84 tuiles, dont 12 tuiles avec des rivières (non utilisées dans cette version). Il y aura donc **72 tuiles**.
- Chaque joueur a un compteur de points.
- Une tuile est décomposable en 5 morceaux localisés sur la tuile : à **droite**, en **haut**, à **gauche**, en **bas** et au **milieu**.
- Un morceau de tuile peut être :
    - Un **pré**
    - Une **route**
    - Une **ville**
    - Une **ville** avec **blason**
    - Un carrefour (ou **village**)
    - Une **abbaye**
- Chaque tuile peut être représentée comme ceci :

	X	
X	X	X
	X	

X prend l’une des valeurs suivantes : ville, pré, route, ville avec blason, village, abbaye.

2. Règles de pioche et de placement des tuiles

- Seules les tuiles au-dessus de la pile peuvent être piochées, et les tuiles constituant la pile doivent être agencées dans un ordre aléatoire.
- Les tuiles doivent pouvoir être posées sur une grille, les unes à côté des autres.
- La taille de cette grille doit pouvoir être augmentée afin de permettre aux joueurs de poser une tuile adjacente à n’importe quelle autre.
- Lorsque c’est le tour du joueur, lui proposer tous les endroits où il peut placer sa tuile par rotation de 90° soit 4 possibilités maximum par emplacement.

3. Gestion des joueurs et des meeples

- Le nombre maximal de joueurs, IA compris, est de 5, le nombre minimal est de 2.

- Le ou les joueurs doivent également pouvoir jouer avec des IA, et ces dernières doivent pouvoir être différenciées des joueurs par le programme.
- Chaque joueur commence la partie avec 5 meeples, qu'il doit pouvoir dépenser et regagner, mais il ne doit jamais avoir plus de 8 meeples ni moins de 0.
- Le joueur dont c'est le tour doit pouvoir être identifié.
- Un indicateur sur chaque tuile doit représenter si un meeple est actuellement sur la tuile et, si oui, à quel joueur il appartient.

4. Placement et contrôle des meeples

- Avant de proposer au joueur de placer le meeple sur un morceau de sa tuile qu'il vient de poser, il faut vérifier qu'aucun meeple ne contrôle déjà cette zone. Par exemple, pour une route, il faut vérifier qu'elle n'est pas reliée à d'autres routes et, si c'est le cas, il faut vérifier qu'aucun meeple n'est déjà présent sur cette route. Si et seulement si ces conditions sont respectées, alors proposer au joueur de placer son meeple sur le morceau route de la tuile qu'il vient de poser.

5. Comptage des points en cours de partie

- Si les deux extrémités d'une route sont adjacentes à une ville, un village, une ville avec un blason, une abbaye, ou que la route se referme sur elle-même, alors il faut retirer tous les meeples sur cette dernière, puis le ou les joueurs ayant le plus de meeples sur cette dernière récupèrent 1 point par tuile route. On retire ensuite les meeples sur la route en question.
- Quand une tuile ville est posée sur la grille, toutes les cases ville adjacentes les unes aux autres doivent être vérifiées de sorte à déterminer si la ville est fermée. Si la ville est fermée, il faut ajouter 2 points pour chaque case ville et 4 points pour chaque blason au joueur ou groupe de joueurs qui possède le plus de meeple sur la ville. Retirer ensuite tout les meeple de la ville.
- Chaque fois qu'une case est posée à côté d'une abbaye, il faut vérifier si elle est complètement entourée. Si oui, retirer le meeple et ajouter 9 points au joueur qui possède le meeple, sinon ne rien faire.

6. Fin de partie et comptage final des points

- Lorsque le nombre de tuiles dans la pile atteint 0 et que le joueur a terminé son tour, compter les points et arrêter la partie. Pour ce faire il faut parcourir toute la grille pour vérifier si un meeple est sur une case. Si oui, compter les points de cette case en fonction de son type.
- Adapter la méthode de comptage des points des routes pour la fin de partie. Rien ne change, sauf le fait que la route n'est pas forcément fermée.
- Adapter la méthode permettant de compter les points d'une ville fermée et réduire de moitié le nombre de points offerts par les cases ville.
- Adapter la méthode permettant de compter les points d'une abbaye, compter le nombre de tuiles adjacentes et ajouter ces points (1 par tuile adjacente) au score du joueur possédant le meeple.

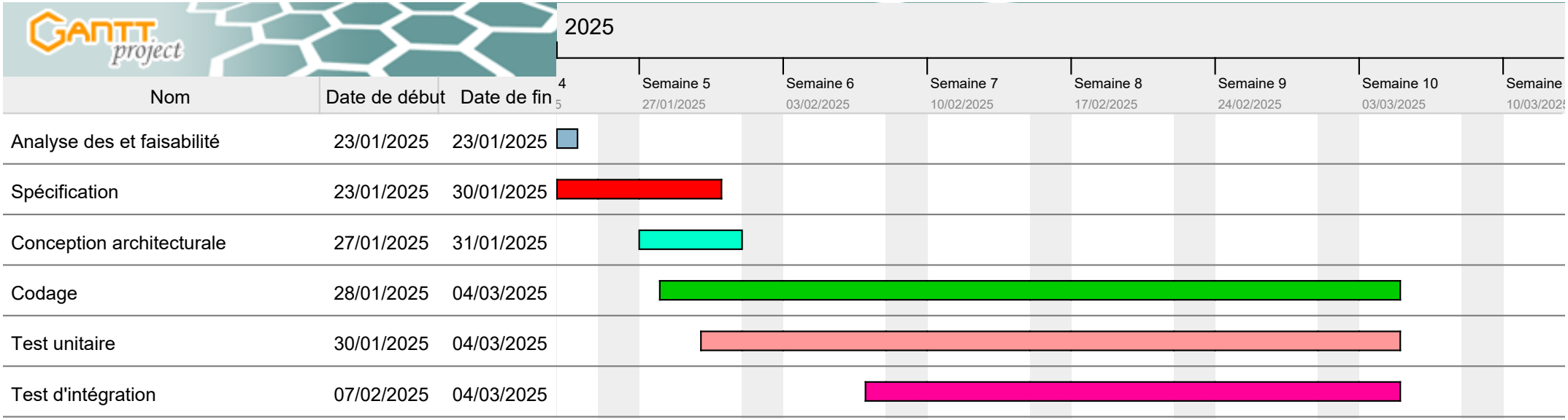
7. Règles spécifiques aux intelligences artificielles

- Les intelligences artificielles poseront une tuile aléatoirement parmi celles qu'elles ont le droit de poser.

Tâches

Nom	Date de début	Date de fin
Analyse des et faisabilité	23/01/2025	23/01/2025
Spécification	23/01/2025	30/01/2025
Conception architecturale	27/01/2025	31/01/2025
Codage	28/01/2025	04/03/2025
Test unitaire	30/01/2025	04/03/2025
Test d'intégration	07/02/2025	04/03/2025

Diagramme de Gantt



Nom	Date de début	Date de fin
Analyse des et faisabilité	23/01/2025	23/01/2025
Spécification	23/01/2025	30/01/2025
Conception architecturale	27/01/2025	31/01/2025
Codage	28/01/2025	04/03/2025
Test unitaire	30/01/2025	04/03/2025
Test d'intégration	07/02/2025	04/03/2025