

Eléments d’analyse

Corinth est un jeu qui se joue à trois ou quatre joueurs. Chacun des joueurs évolue de façon quasi individuelle. Un plateau commun permet aux joueurs de récupérer des biens ou des pas. Des cartes individuelles attribuées à chacun permettent de résumer les biens récupérés au cours de la partie. La partie prend fin après un certain nombre de tours défini en fonction du nombre de joueurs.

- Le programme a besoin :
 - D'un tableau de 6 cases de type structuré afin de représenter le plateau commun. Chaque case de ce tableau prendra à chaque manche comme informations le nombre et les valeurs des dés lancés. Le type structuré est déclaré dans le Headers "déclaration".
 - La carte personnelle regroupant les informations de chacun des joueurs dont la quantité de marchandises acquises, le nombre de pièces et de chèvres et de dé jaunes récupérées, les bâtiments construits et l'avancée du joueur dans son intendant. Cela a été exprimé sous forme d'un type structuré "joueur" déclaré dans le Headers "déclarations".
 - Chaque manche commence par un lancé de dé effectué par un joueur. L'ordre de passage des joueurs pour le lancé de dé est décalé à chaque manche de même pour l'ordre de jeu.
- Lors d'une manche, le programme affiche:
 - La carte personnelle des joueurs
 - Le plateau après le lancé de dé, et au fur et à mesure des acquisitions.
 - L'intendant quand celui ci est choisi par le joueurA la fin du jeu, le gagnant sera affiché.
- L'application est faisable et le traitement informatique et l'acquisition des données est essentiellement modélisé par des entiers et des chaînes de caractères.
- Le programme a quelques limites:
 - Lors d'une manche, s' il ne reste plus de biens à acquérir sur le plateau (toutes les possibilités ont été prises par les joueurs précédents) le reste des joueurs ne joueront pas, contrairement aux règles du jeu qui attribuent un pas dans l'intendant a tout joueur n'ayant pas pu jouer.
 - La deuxième limite est que lorsque le joueur choisit une case du plateau n'ayant aucun dé (par erreur par exemple) celui-ci n'acquiert rien.
 - Enfin la dernière limite serait l'interface graphique pas forcément esthétique.
- Un scénario possible:
 - 1ere manche le joueur 1 n'achète pas de dé jaune. Il lance les dés.
 - Résultat:
 - 2 Dé de valeurs 2 sur les chèvres
 - 3 Dé de valeurs 5 sur les pièces d'or
 - 4 Dé de valeur 3 sur le marchand vert
 - Le joueur 1 choisit de prendre 4 marchandises du marchand vert qu'il place dans l'échoppe 1. Il n'achète pas de bâtiments par faute de moyen.
 - Il reste:
 - 2 Dé de valeurs 2 sur les chèvres
 - 3 Dé de valeurs 5 sur les pièces d'or
 - Le joueur 2 choisit les dé sur la case pièce d'or et décide de faire bouger l'intendant. Il s'arrête sur une marchandise orange qu'il place dans l'échoppe 2. Il n'achète pas de bâtiments par faute de moyen.
 - Le joueur 3 choisit d'acquérir les 2 chèvres. Il n'achète pas de bâtiments par faute de moyen.
 - La manche 1 est terminée. A la manche 2, le joueur 2 lancera les dés et choisira sa case en premier suivi du joueur 3 puis 1.
- Remarque: Si l'échoppe d'un marchand est entamée, celle -ci devra être complétée avant qu'une échoppe du même marchand ne soit commencée.

Schéma des Structures de données utilisées

- Intendant :

 - position (a et b)
 - nombre de points gagnés par l'intendant
 - nombre de fois que s'est arrêté l'intendant
 - plateau 7*7 (structure)
- Joueur :

 - nom
 - compteur de points
 - compteur d'or
 - compteur de chèvres
 - compteur de dés jaunes
 - 4 quartiers (structure)
 - 4 bâtiments (structure)
 - Intendant (structure)
- Plateau :

 - état de la case : 0 pour pas possible d'y aller, 1 pour possible
 - bonus associé à la case
- Plateau (pour placer les dés) :

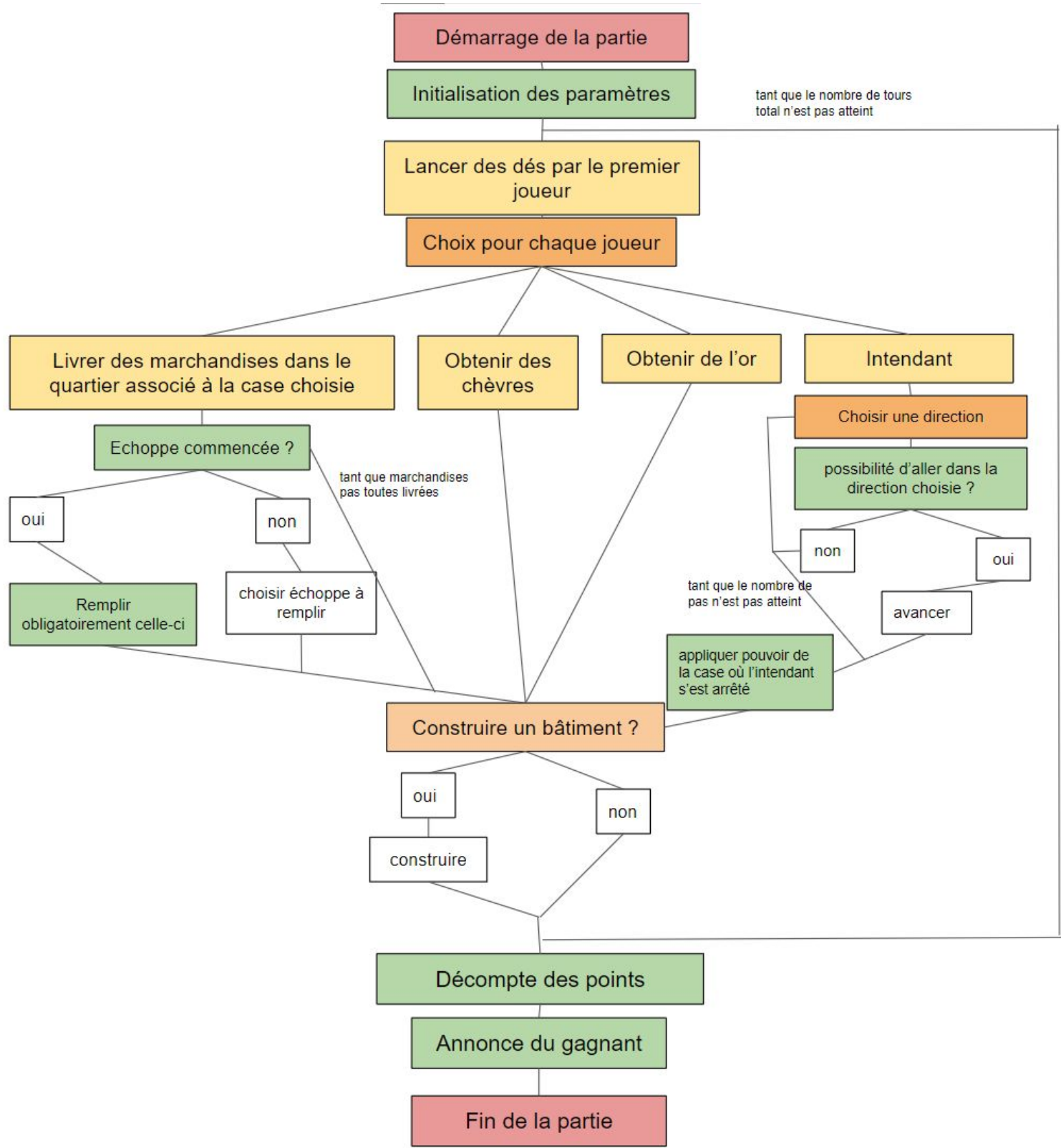
 - valeur des dés placés
 - nombre de dés placés (blancs, jaunes et au total)
- Quartier :

 - statut : 0 pour pas terminé, 1 pour terminé
 - nombre de points bonus si quartier terminé
 - 4 échoppes (structure)
- Échoppe :

 - statut : 0 pour pas terminé, 1 pour terminée
 - nombre de marchandises actuellement
 - nombre de marchandises au maximum
 - nombre de points bonus si échoppe terminée
- Bâtiment :

 - nom
 - statut : 0 pour pas construit, 1 pour pas construit
 - prix en or
 - prix en chèvres

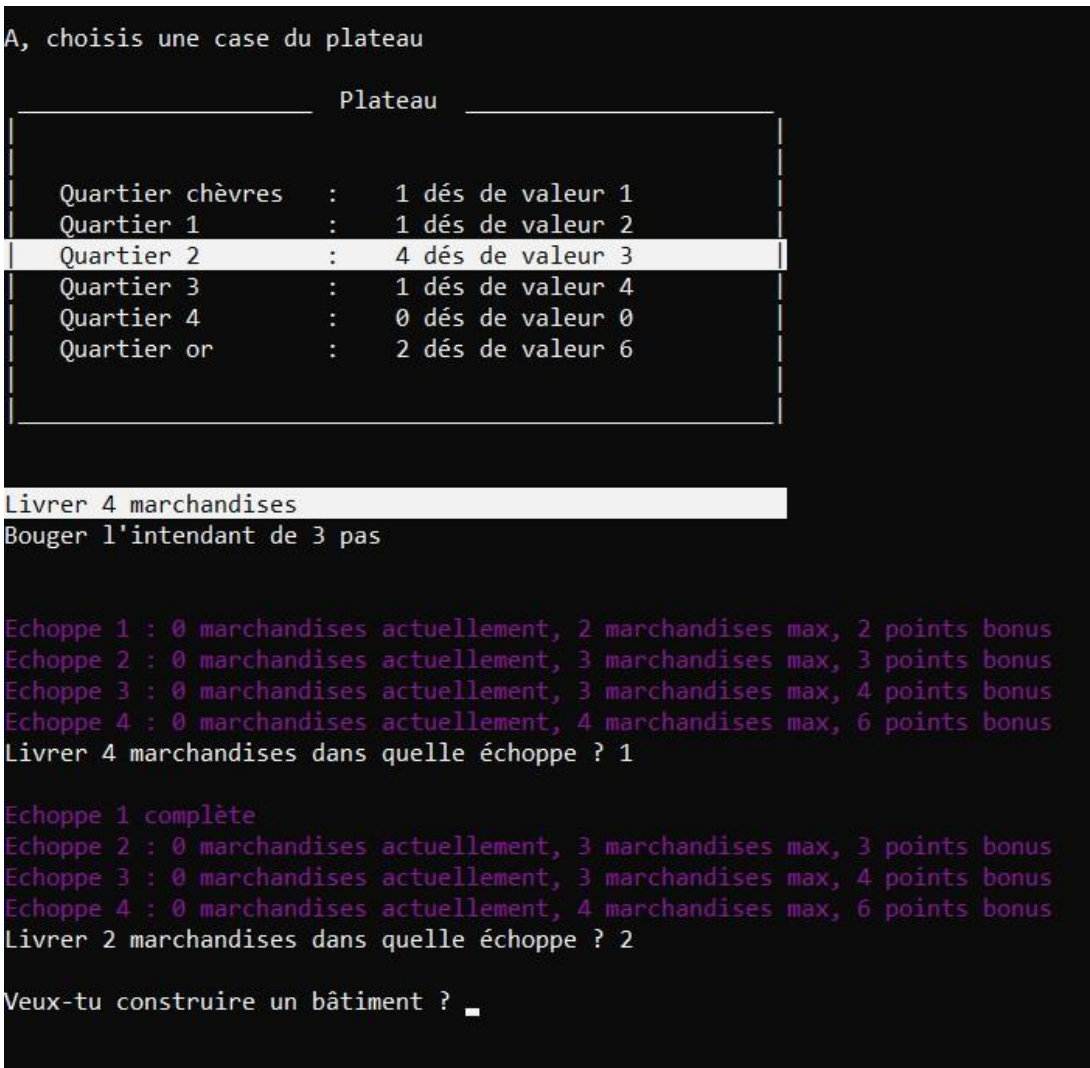
Eléments de conception



Capture d’écran



Carte personnelle du joueur



Plateau de jeu et échoppes violettes

Lots prévus

Semaine 1	Semaine 2	Semaine 3
Compréhension détaillée des règles du jeu. Simplification classification priorisation des étapes principale du jeu. Premier schéma algorithmique. Partage du travail.	Amélioration et optimisation des idées issues de la première semaine. écriture en langage informel des sous programmes principaux Retranscription en langage C et debogage.	Combinaison des sous programmes dans un seul même programme. Debogage. Amélioration de l’interface graphique.

Conclusion

(que changeriez-vous dans votre démarche projet pour être plus efficient une prochaine fois ?)

Un point qui serait à améliorer dans un futur projet serait de plus communiquer concernant nos sous programme et leurs logique de fonctionnement afin d'éviter des problèmes d'incompatibilité lors de la mise en commun. Cela permettrait une meilleure gestion du temps.