

Cahier des Charges

Gestion de l'espace aérien

Groupe 6 :

CROZET Gabin

FINOT Guillaume

LAGARDE Noah

Table des matières

I / Contexte	3
QQOQCP	3
II / Objectifs	4
SMART	4
III / Périmètre	7
a) Enjeux	7
b) Contrainte	7
c) Limite du projet	8
IV / Fonctionnalités	8
MoSCoW	8
V / Planification	9
a) Découpage projet	9
b) Affectation des tâches	11
c) Planning	13
d) Maquettes	Erreur ! Signet non défini.

I / Contexte

QQOQCP

Quoi

Gérer l'espace aérien français

Qui

Clients : personnel de l' aérospatial

Membres du groupe : Guillaume, Gabin, Noah

Où

IUT Lyon1

Quand

Cahier des charges : 13 mai

Projet final : Semaine du 24 juin

Comment

Pas de moyen financier et à l'aide de nos compétences acquises lors des différents cours et de nos ordinateurs.

Pourquoi

Optimiser les trajets des avions afin d'éviter les accidents/conflits de vols dans le cadre d'un projet à visée pédagogique.

II / Objectifs

SMART

Spécifique

Nous devons gérer l'espace aérien français pour des vols entre plusieurs aéroports. On devra repérer si deux vols seront susceptible de rentrer en collision. Ils seront considéré ainsi si il se croisent en un point en moins de 15 minutes. Avec ces données nous ferons des graphes entre les différents vols dans lequel chaque sommet représentera des vols et les arêtes si le risque de collision est présent. Nous devons colorier ces mêmes graphes. Tout ceci avec une interface graphique ergonomique, intuitive et facile d'utilisation.

Mesurable

Nous devons créer une interface graphique qui va permettre de visualiser l'espace aérien français pour plusieurs vols entre plusieurs aéroports. Pour ce faire nous allons donc construire des graphes, où un sommet représente un vol, et dans lequel 2 sommets sont adjacents si les deux vols sont susceptibles de rentrer en collision.

Pour ce faire nous allons devoir créer une interface graphique, qui va permettre de visualiser les différents graphes qui seront construits et coloriés. Chaque sommet sera colorié en fonction de son niveau de vol, permettant de minimiser les conflits, une marge 5% sur les conflits sera considéré comme acceptable.

L'interface graphique devra nous permettre de visualiser le graphe en cours ainsi qu'une coloration de ce dernier, de charger directement un graphe au format .csv, de choisir parmi plusieurs algorithmes de coloration, d'afficher les différentes statistiques du graphe, de charger une liste de vols et d'aéroports au format .csv, d'afficher et d'éditer le kmax, et de zoomer sur le graphe. L'interface graphique devra être ergonomique et intuitive

Enfin il pourra également comprendre les fonctionnalités comme, la possibilité de changer la marge de sécurité pour les vols, de visualiser les vols et les aéroports sur une carte de la France, afficher les vols en cours à une heure donnée, sélectionner un aéroport pour visualiser les niveaux de chaque vol, et de sélectionner un niveau de vol pour afficher tous les vols de ce niveau. Il faudra également que notre produit final soit capable de charger une liste de 20 graphes d'évaluation et de les colorier en limitant à 5% les conflits en moins de 30 minutes.

Atteignable

Pour faire cette application permettant de visualiser et de gérer l'espace aérien français. On aura besoin dans un premier temps d'un langage de programmation adéquat. Nous utiliserons donc le Java qui permet d'implémenter les algorithmes dont nous aurons besoin. Ce langage permet d'implémenter une interface graphique. Nous utiliserons différentes librairies telles que GraphStream que nous avons vus en TP de Java qui permettra de visualiser les différents graphes.

Une interface graphique sera implémentée à notre projet final. Nous pourrons faire ceci grâce à nos cours d'IHM. Nous nous appuierons également sur les différentes documentations de Java

pour nous aider à mener à bien le projet, sur des formules mathématiques permettant de convertir les coordonnées (latitude et longitude).

Sur le plan algorithmique, plusieurs algorithmes de coloration de graphes nous sont mis à disposition pour effectuer la coloration. Ils nous permettront de visualiser le niveau de vol des différents avions tout en limitant le plus possible les conflits. Ce sont les cours de maths qui nous permettront de faire ceci.

Réaliste

La gestion de l'espace aérien est quelque chose de très important, en effet, il permet la bonne circulation des avions, c'est donc quelque chose de concret. Il est important de pouvoir réguler cette circulation pour éviter les collisions entre les appareils.

C'est ce que nous souhaitons faire en créant une application qui permet de gérer l'espace aérien, et dans notre cas nous aurons des données que nous devrons traiter pour limiter au maximum les conflits.

Temporel

La date butoir du projet sera le 24 juin, toutes les parties du projet devront être achevées pour cette date

III / Périmètre

a) Enjeux

Le projet revêt une importance cruciale puisque la vie des passagers à bord de l'avion sont en jeu. Avec ce logiciel, les avions pourront circuler à des niveaux de vols différents. Favorisant la sûreté de ce mode de transport essentiel au regard du contexte actuel caractérisé par une croissance permanente du trafic aérien. Il pourra également améliorer la rationalisation du système interne et en optimisant l'effectif salarial ; leur organisation sera de meilleure qualité aura un impact abonnant la satisfaction client en limitant les retards, et les annulations des vols.

b) Contrainte

Ce projet est d'une ampleur conséquente, il y a donc une nécessité de le découper en petites étapes mettant l'accent sur les différentes fonctions à implémenter. S'ajoutent à cela des contraintes techniques pour répondre aux attentes du client telle qu'une gestion rigoureuse des erreurs. Dans notre cadre il n'y a pas de contraintes financières particulières étant donné que c'est un projet à visée pédagogique. Nous serons 3 étudiants sur ce projet, une certaine marge sur la date butoir a été prise pour anticiper d'éventuels imprévus

c) Limite du projet

Les fonctionnalités principales seront notre priorité, les fonctionnalités secondaires seront implémentées par la suite afin que notre projet final soit fonctionnel et efficace. Faire attention à ne pas aller trop loin si le temps ne nous permet pas pour faire attention aux détails. Certains graphes ne pourront pas avoir une coloration optimale du fait qu'il n'existe actuellement pas d'algorithmes pouvant faire une coloration parfaite, il existera des graphes test qui ont un k_{max} inférieur au nombre de couleurs possible, on limitera alors les conflits dans ces cas-là. On négligera également la forme sphérique de la Terre du fait de la complexité des calculs.

IV / Fonctionnalités

MoSCoW

Vitaux :

- Extraction des données
- Construction d'un graphe en fonction des données d'un fichier
- Affichage d'un graphe
- Coloration d'un graphe en fonction des contraintes données
- Implémenter différents algorithmes de coloration

Essentiels :

- Choisir les algorithmes de coloration
- Afficher/éditer k_{max}
- Bien organiser le code en package

- Gérer les erreurs
- Afficher les stats du graph en cours

Conforts :

- Zoom avant et zoom arrière
- Changer la marge de sécurité

Luxe :

- Visualisation de la carte de France en interface graphique avec les aéroports.
- Afficher les vols en cours à une durée donnée.
- Visualiser le niveau de chaque vol sur la carte de France
- Afficher les vols pour un niveau donné
- Afficher les différents niveaux de chaque vol pour un aéroport donné

V / Planification

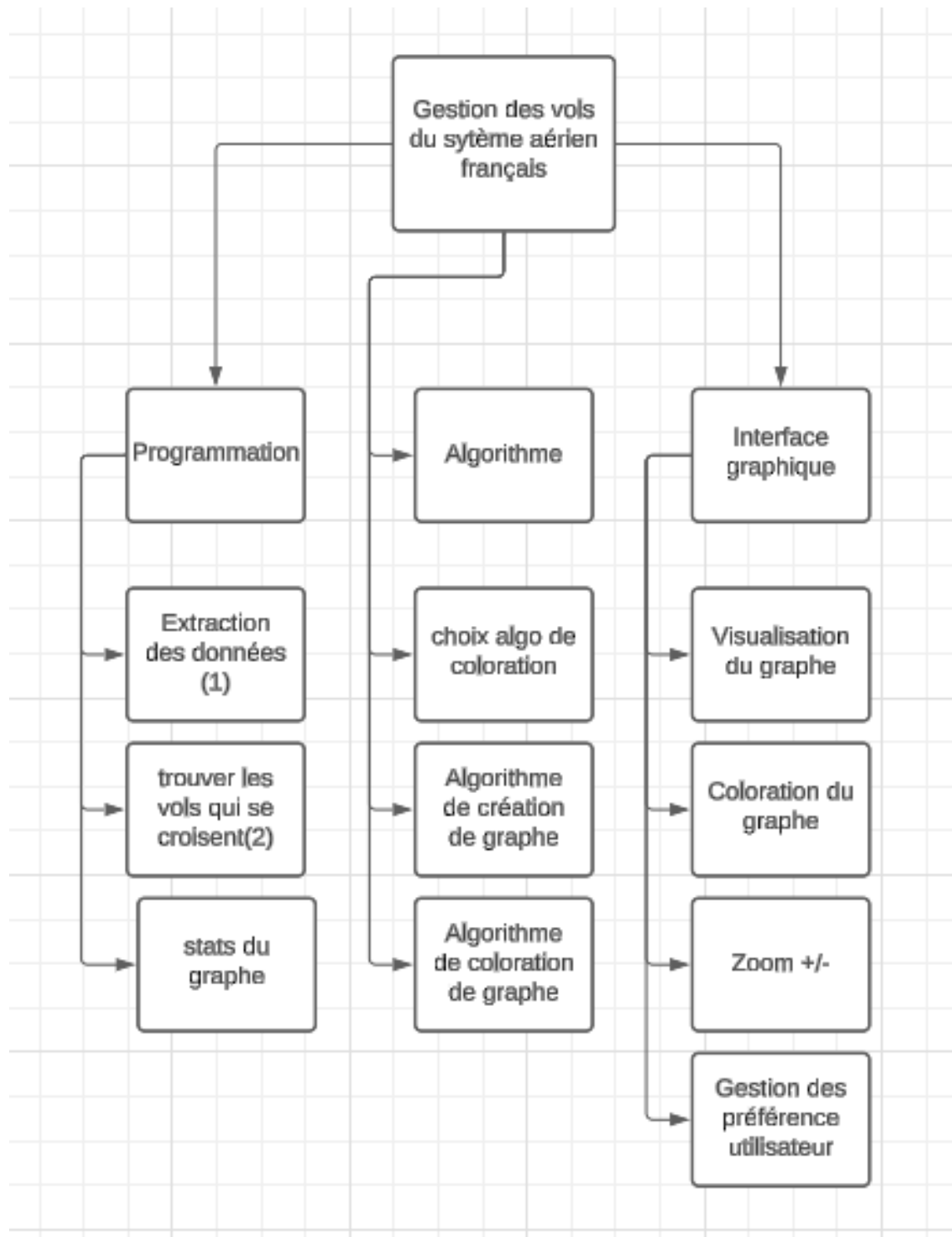
a) Découpage projet

(Temporel/Structurel/PBS/WBS/OBS)

Nous choisissons le découpage PBS (Product Breakdown Structure), car c'est le plus adapté à notre projet. En effet, il possède des avantages, comme la possibilité de travailler sur deux tâches en parallèle ou encore d'identifier toutes les fonctions qui composent le produit à réaliser.

On aurait pu choisir également le découpage WBS (Work Breakdown Structure) mais pour des raisons de contraintes de temps, nous préférons choisir le découpage PBS. En effet, le découpage WBS découpe chaque fonctionnalité en trois étapes, l'étape de conception,

de développement, et de test. Dans le cadre d'un projet comportant un grand nombre de fonctionnalités et avec un délai relativement court, cette méthode n'est donc pas la plus judicieuse.



b) Affectation des tâches

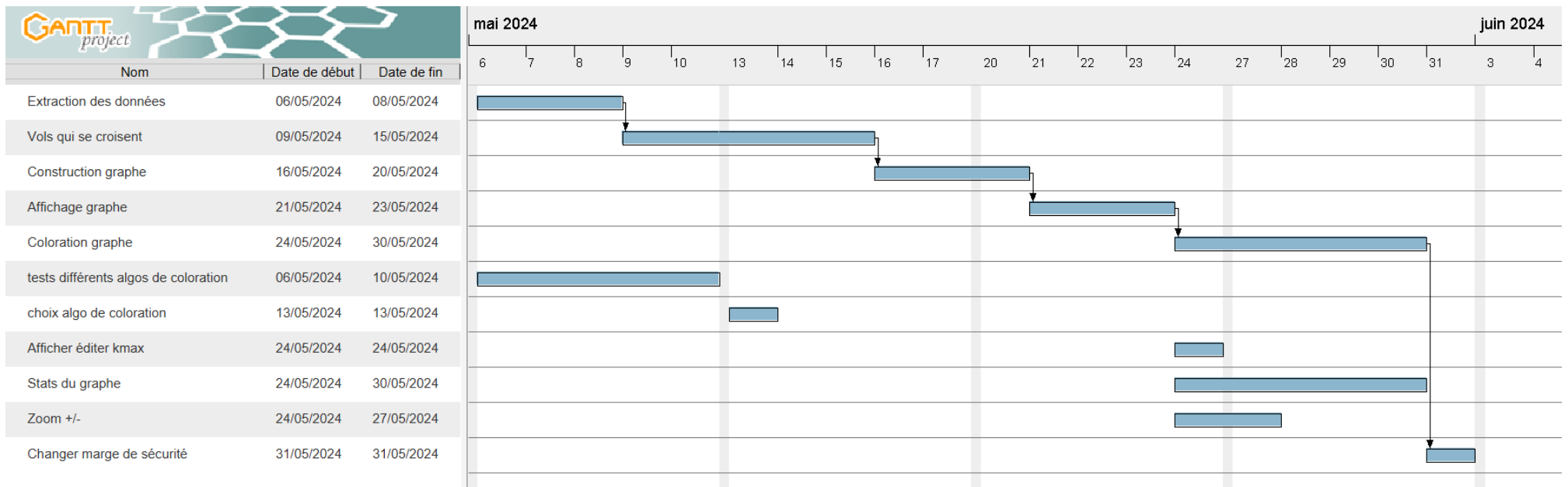
(RACI)

	Noah	Gabin	Guillaume	Professeur
Extraction des données	A	I	R	C
trouver les différents vols qui se croisent	A	I	R	C
Construction d'un graphe en fonction des données d'un fichier	R	A	I	C
Affichage d'un graphe	R	A	I	C
Coloration d'un graphe en fonction des contraintes données	I	R	A	C
Implémenter différents algorithmes de coloration	I	R	A	C

Choisir les algorithmes de coloration	I	R	A	C
Afficher/éditer kmax	A	I	R	C
Bien organiser le code en package	I	R	A	C
Gérer les erreurs	R	R	R	C
Afficher les stats du graph en cours	I	A	R	C
Zoom avant et zoom arrière	R	I	A	C
Changer la marge de sécurité	R	I	A	C

La gestion des erreurs et l'organisation se fait tout au long du projet par les différents membres

c) Planning



d)Maquette

Lorsque qu'un bouton est sélectionné, les différentes sous-fenêtres apparaissent

