Université Nice Sophia Antipolis

Traffic Control : Sujet 6

Auteur :
Balde Thierno
Diop Mamadou
Fezai Ahmed
Chennouf Mohamed

Responsable :
M. Guilhemi Molines
M. Philippe Collet

9 octobre 2017

1 Scope

Notre projet est une simulation de trafic routier . Nous aurons la possibilité de gérer le nombre de voiture mais aussi d'ajouter un événement dans la ville qui, suivant son importance va influer sur le trafic . Nous allons importer notre carte via une API déjà existante.

2 Exigences

2.1 User stories

Afin de pouvoir repérer le maximum d'exigences nous avons établie des cas utilisations :

- Le maire de Nice (christian estrosi) souhaite organiser le prochain Roland-Garros. Afin d'éviter des embouteillages , il souhaite simuler le trafic autoroutier lors d'un événement de tennis telle que Roland-Garros.Il estime qu'il y aura environ 300 véhicules en direction de l'evenement.
- Le maire de Marseille (jean-claude gaudin) souhaite simuler le trafic autoroutier lors de deux événements simultanés (un match de foot et un tournois de hockey sur glace).Il sait qu'il existe une intersection entre les chemins emprunter par les supportère des deux événements et souhaite gérer de la meilleure façon le feu de cette intersection
- Le prochain match Nice-Paris se déroulera à Alliance Riviera à Nice, pour cela le maire de Nice veut absolument éviter que les personnes ayant payé leur place 150 euros arrivent en retard au match. De ce fait il souhaite simuler l'arrivée de 150 voitures venant de l'auroute A8 ainsi que 300 venants du centre-ville de Nice. Il aimerai donc élaborer un system de gestion des feux capable d'éviter tous embouteillages entre la sortie autoroute et le stade ainsi qu'entre Nice Centre et le stade.

2.2 Exigences fonctionnelles

De ces cas utilisations ressortent des éxigences fonctionnelles (module du système à développer : ce que le système doit faire) :

- Service de simulation : pouvoir générer un nombre de véhicule et l'insérer dans la simulation
- Service d'événement : pouvoir ajouter un événement à la simulation
- Service de classification des chemins : pouvoir évaluer les chemins les plus empruntés par les automobilistes lors d'un événement
- Service de gestions des temps des feux : gérer le temps des feux en fonction du service de classification des chemins

2.3 Exigences non-fonctionnelles

- Fiabilité : Lorsque l'on parle de trafic routier, il faut à tout pris éviter l'accident , de ce fait notre système doit à tout instant être capable de remplir sa mission qui est de gérer le temps des feux sans que les feux ne soit en contradiction.
- Efficacité : Il faut que notre système soit suffisamment performant pour que le trafic soit fluide : (qu'il n'y ai pas de bouchon en direction de l'événement sans perturber le trafic environnant)
- Adaptabilité : Il faut que notre système soit adaptable à d'autres environnements (plusieurs villes ou événements d'importances différentes)
- Passage à l'échelle : Il faut que notre système soit capable de gérer de grandes villes avec un nombre important de véhicule

Nous avons mis aussi des attributs orientés-processus eux aussi rangés par ordre importance :

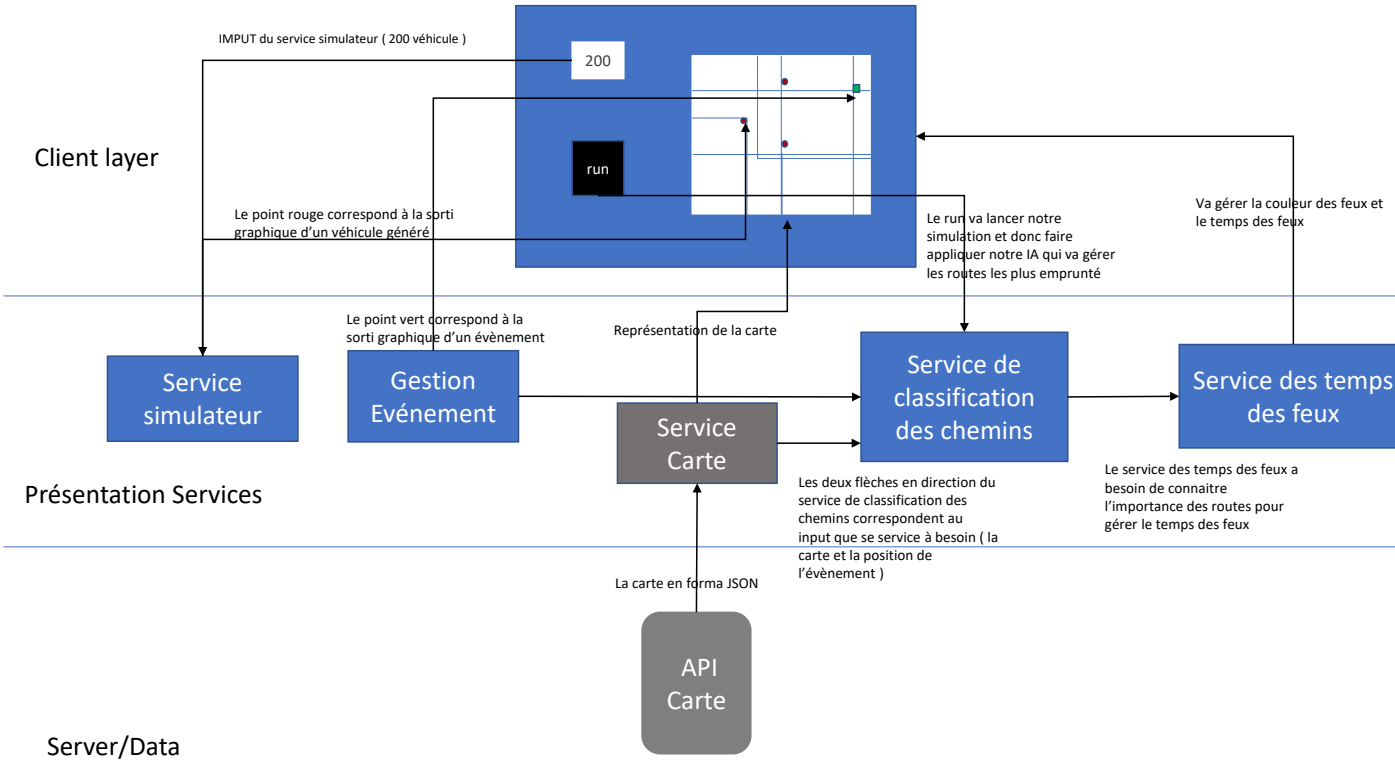
- Testabilité : facilité à tester

- Compréhensibilité : conception, architecture et code facile à comprendre/apprendre
- Intégrabilité : facilité à intégrer des composantes

2.4 Contrainte

- Pas de gestion sur l'heure de début de l'événement : étant donné que c'est une simulation on suppose qu'à partir du moment où l'événement est mis et que l'on lance la simulation , celui-ci est actif.
- Pas de bouchon en direction de l'événement (car les automobilistes ont surement payé leurs places pour assister à l'événement) cependant une fois l'événement fini , dans le cas d'un chemin retour , on peut se permettre une circulation un peu moins fluide.
- Respecter les rues à sens unique de circulation .

3 Walking Skeleton



Service simulation : c'est un service qui va nous permettre de mettre en input un certain nombre de véhicules et de les positionner sur la carte

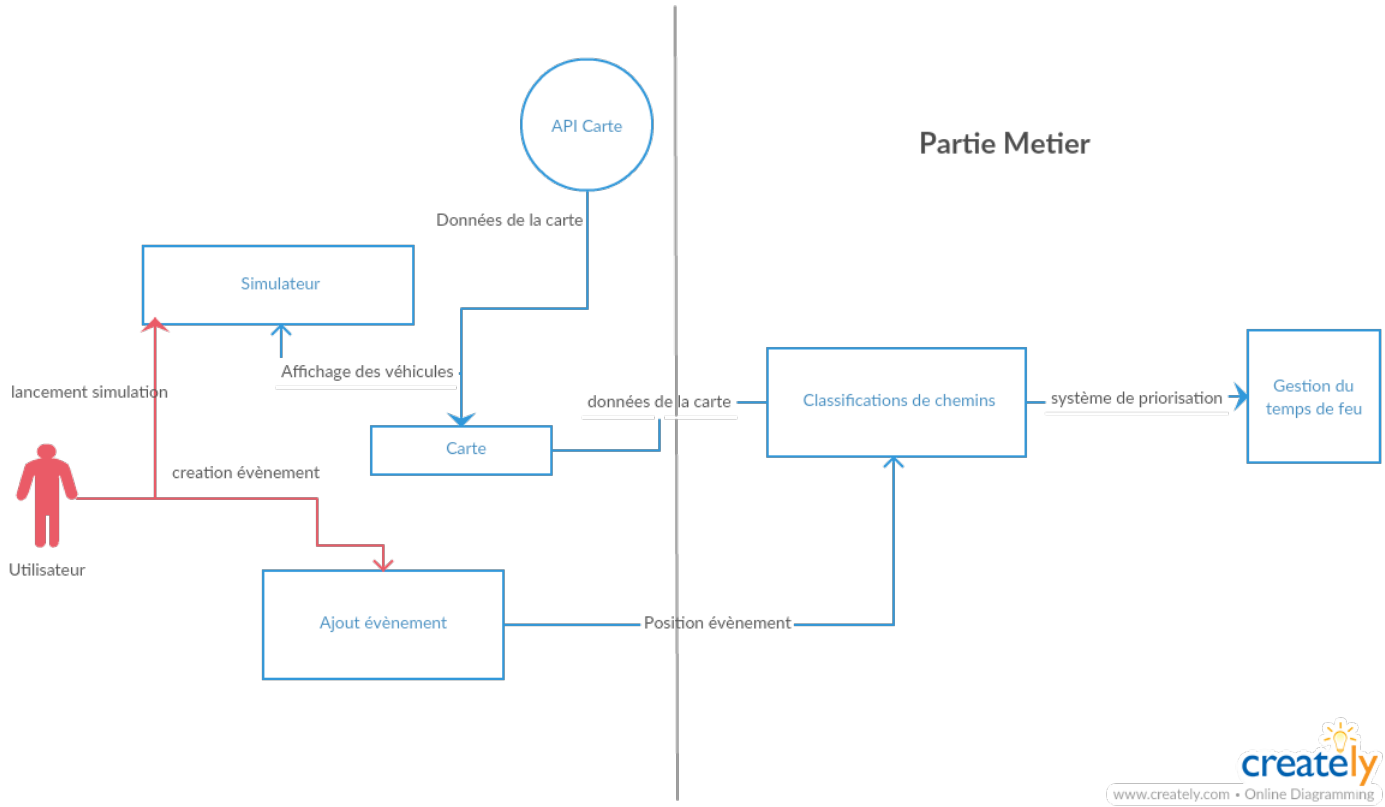
Gestion Événement : c'est un module qui va positionner un événement sur la carte

Service de classification de chemin : c'est un service qui utilise des algorithmes de graphe afin de mettre un poids fort aux routes les plus empruntées et un poids faible à celle qui le sont moins

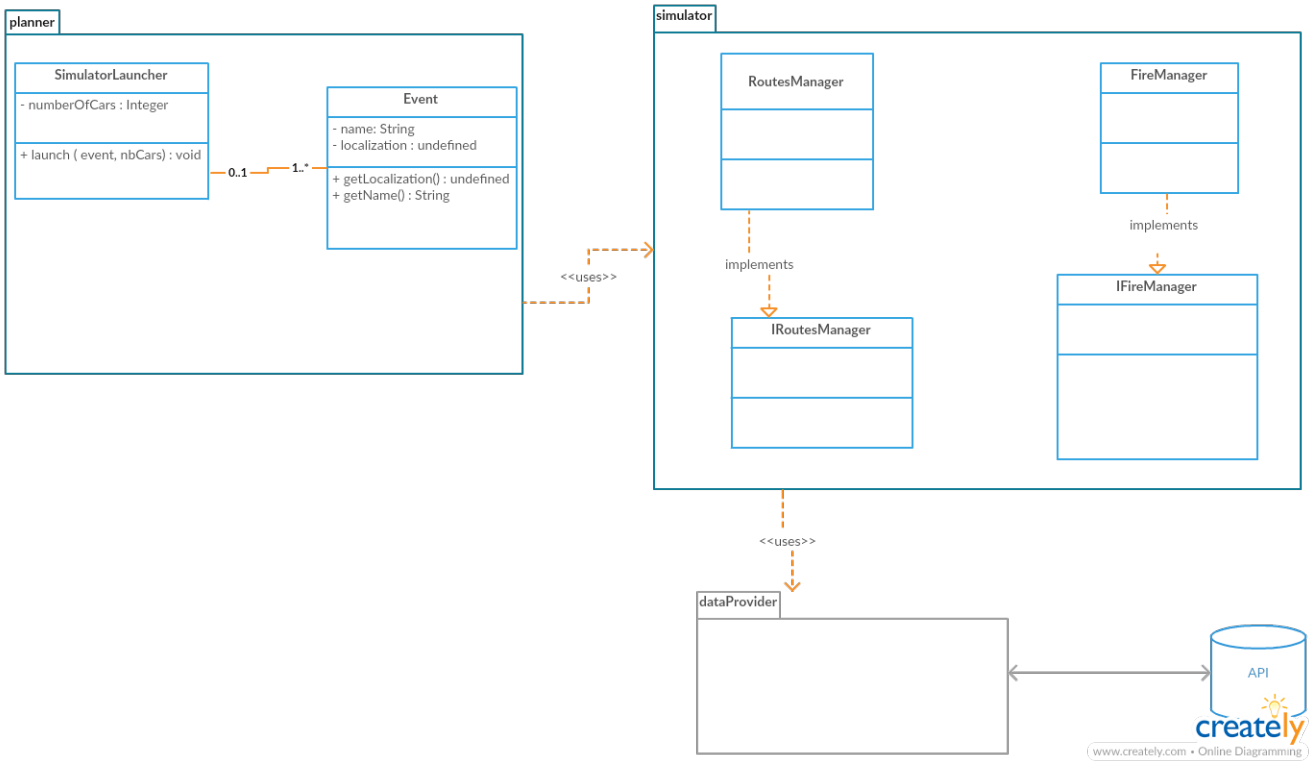
Service temps des feux : service qui va prendre en entrée le retour du Service de classification de chemin et va gérer les feux en conséquence

Service carte : utilise un API qui génère une carte au format X et la représente graphiquement

4 Diagramme de composants



5 Diagramme de classe (à compléter au fur et à mesure de l'implémentation)



Nous avons également mis en place un diagramme de classes (évolutives au fil des semaines qui suivront) séparées dans des packages que sont :

Planer : On y retrouve les classes nécessaires pour la configuration d'une simulation (ajout de l'évènement, véhicules ...)

Simulator : Cela regroupe les classes qui interviendront dans l'exécution de la simulation du trafic (de la gestion des routes à celle des feux).

Data Provider : Contenant les classes nous permettant de récupérer toutes les informations nécessaires pour le trafic, il interagit avec un jeu de données (que nous créerons) ou avec une API.

6 Technologie et Language

6.1 Intérêt de la technologie

Nous avons choisi d'utiliser Java pour différentes raisons qui sont les suivantes :

Java nous permet grace à javax.swing.jfram / javax.swing.panel / javax.swing.jbutton .... de creer une interface de simulation et des actionneurs de facon très rapide .

Java a une très bonne documentation.

Java a de très bon Outils pour notre simulation qui sont :

- frameworks de tests comme jUnit ou Mockito
- Spring est un framework libre pour construire et définir l'infrastructure d'une application java, dont il facilite le développement et les tests.
- Maven qui est un outil de gestion et d'automatisation de production des projets logiciels

Java gère bien les exceptions, l'heritage, les interfaces.

Java a une rapidité d'itérations élevé et le développement est moins lourd

Java a une bonne gestion des dépendances (API exportation des cartes)

6.2 préférence et gain de temps

Nous sommes un groupe qui a déjà l'habitude de coder avec Java et utiliser des outils sité ci-dessus. Nous avons donc majoritairement une préférence pour ce language. Cela nous permettra aussi d'économiser un gain de temps énorme sur l'apprentissage du langage et de ces technologies.

7 Environement de travaille

Nous utilisons la plateforme colaborative github :`https://github.com/mohamedchennouf/Al_Traffic_ControlV6`, nous nous attribuons des taches graces à Trello `https://trello.com/` qui nous permet de nous distribuer des issus/tache/defect.

8 Team roles

nom	role
Mohamed Chennouf	Scrum master
Mamadou Abdoulaye Diop	Full-stack developer
Ahmed Fezai	Front-end
Balde Thierno	Back-end

Ce sont des rôles provisoires mais de manière générale tous le monde sera amené à réaliser des modules qui peuvent toucher le front-end comme le Back-end ainsi que les tests associés.