Compréhension personnelle du projet de programmation 2

Objectif : Développer un système de classe qui communiquent entre elles et mettre en exergue la programmation par évènements.

Problème posé : Problème de recherche de stationnement rapide

Solution propose par l’user : Lui informé des parkings dispo a l’instant T ou il va dans l’application développée pour gérer les stationnements.

Recadrage de l’objectif : développer une application qui simule se fonctionnement en ce basant sur un espace restreint. Comme le Rideau center ou nous avons souvent des problèmes de stationnement qui se posent sur l’espace de parking de ce Mall.

Exigences de l’application  et questions:

* L’application gère juste une zone de parking bien précis d’une zone.
* L’application doit surveiller l’état des différents parkings continuellement.

1. D’où ma première question Comment souhaitez-vous que les données sur la disponibilité des places de stationnement soient collectées et traitées ? Existe-t-il déjà des systèmes de capteurs installés dans les parkings ?

* L’application devra calculer la distance entre le conducteur et le parking vide une fois trouvé.

1. Ce qui me pousse a ma seconde question comment faire ce calcul? quel est le type d’environnement de la zone cad est-ce qu’il s’agit d’un environnement 3D,2D ou encore une utilisation du GPS au quel cas la question se pose sur comment implémenter ca car il s’agit d’un calcul complexes il y’a beaucoup de données a prendre en compte pour ce calcul aussi pour un environnement 3D ou l’on devra considère les immeubles et autres éléments de la zone , le 2D sera l’option la plus probable des 3 qui est moins complexes du point de vue calculs.

* Les interfaces devront communiquer entre elles

1. Renvoyant à ma troisième question qui est Comment souhaitez-vous que les différentes parties du système (interfaces graphiques, calcul de chemins, surveillance des stationnements, etc.) communiquent entre elles ?
2. Quel niveau de qualité attendez-vous pour le code source et l'interface utilisateur de l'application ?
3. L’utilisateur sera-t-il en mesure de réserver un parking?
4. Comment faire toutes ces conceptions graphiques et papier? Voulez vous des dessins de route et voitures et parkings?

Résumé du fonctionnement :

Notre application de stationnement fonctionne en permettant aux utilisateurs de trouver rapidement des places de parking disponibles à proximité de leur destination. L'utilisateur peut rechercher des parkings à l'aide de critères tels que la distance et la disponibilité. L'application utilise des données en temps réel sur l'occupation des parkings pour fournir des informations précises et fiables sur la disponibilité des places de stationnement.

L'application utilise une carte pour afficher les parkings à proximité de la destination de l'utilisateur. Lorsqu'un parking est sélectionné, l'application fournit des informations détaillées sur le parking, notamment son emplacement, sa capacité et sa disponibilité. Pour déterminer la distance entre le conducteur et le parking, l'application utilise une méthode de calcul basée sur la distance euclidienne en 2D. Si l'application devait prendre en compte la 3D, elle utiliserait une méthode de calcul basée sur la distance euclidienne en 3D.

Pour résumer, l'application de stationnement permet aux utilisateurs de trouver rapidement des places de parking disponibles à proximité de leur destination, en utilisant des données en temps réel sur l'occupation des parkings et en fournissant des informations précises sur la disponibilité des places de stationnement.

Pour ce qui est du schema il est conceptuel dans la tete mais besoin de plus d’info pour le dessiner

Phase 2 : Conception des interfaces

Dans ce projet on peut concevoir 3 types d’interfaces comme :

Une interface pour afficher la carte de la zone, avec les routes et les parkings. Également une interface pour afficher les informations sur chaque parking, comme le nombre de places disponibles. Enfin, il faudra une interface pour permettre aux conducteurs de faire une demande de stationnement.

Encore une fois la représentation du point de vue application est possible mais on sait pas s’il s’agit de faire un dessin représentant tout cela ou faire de graphes (qui sont bizarres eux-aussi)

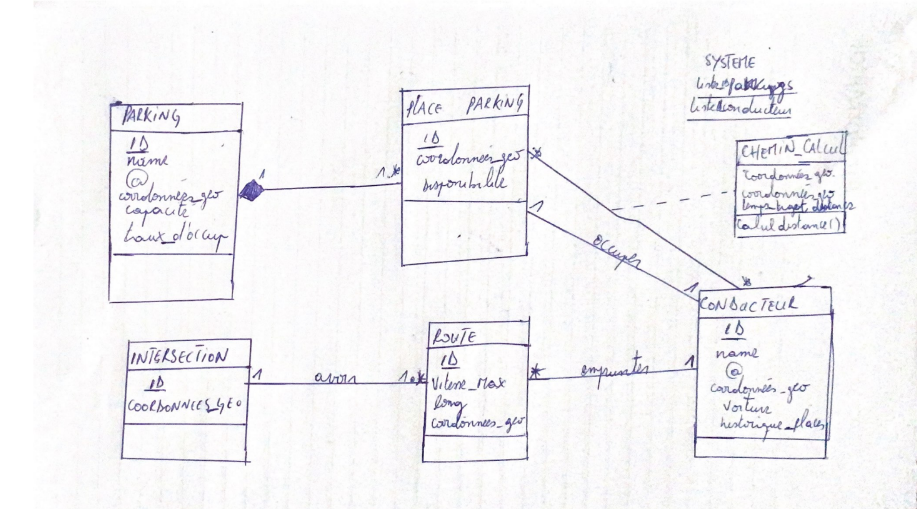
Les correspondances aux besoins de ces interfaces sont :

* Pour afficher la carte de la zone, on peut utiliser la classe Graphics2D.
* Pour afficher les informations sur les parkings, on peut utiliser la classe JLabel.
* Enfin, pour permettre aux conducteurs de faire une demande de stationnement, on peut utiliser la classe JButton.

B. Identification des classes possibles :

Pour ce projet on aura les classes pour représenter les parkings, les routes et les intersections, ainsi qu'une classe pour représenter les conducteurs. On peut également avoir besoin d'une classe pour calculer le chemin le plus court entre deux points. Enfin, on aura besoin d'une classe principale qui permettra de coordonner le fonctionnement de l'ensemble du système.

* Essayons une esquisse de diagramme de classes pour ce projet :



Explication du diagramme de classe:

* Classe Parking :
  + Attributs : nom, adresse, coordonnées géographiques, capacité, taux d'occupation.
  + Relations : peut avoir plusieurs places de parking.
* Classe Place de parking :
  + Attributs : identifiant, disponibilité, coordonnées géographiques.
  + Relations : peut appartenir à un parking, peut être occupée par un conducteur.
* Classe Route :
  + Attributs : identifiant, vitesse maximale, longueur, coordonnées géographiques de début et de fin.
  + Relations : peut avoir plusieurs intersections, peut être empruntée par un conducteur.
* Classe Intersection :
  + Attributs : identifiant, coordonnées géographiques.
  + Relations : peut avoir plusieurs routes qui y convergent.
* Classe Conducteur :
  + Attributs : nom, prénom, adresse, coordonnées géographiques, historique des places de parking occupées, voiture.
  + Relations : peut occuper une place de parking, peut emprunter des routes.
* Classe Chemin\_calcul:
  + Attributs : liste de routes empruntées pour aller d'un point A à un point B, distance totale, temps de trajet.
  + Relations : peut être calculé par l'algorithme de calcul du chemin le plus court.

P.S : Ceci est un essai de diagramme de classes amélioration a venir.