**Mini Projet Programmation II**

**Hiver 2023**

**Prof. Ilham Benyahia**

**Ébauche**

**Apports de la programmation par événements dans un environnement objet**

**Objectifs du projet : Renforcer les compétences et les expériences du développement objet**

L’objectif principal du projet est de vous permettre d’approfondir la compréhension et l’application des concepts de la programmation par objet afin de développer des systèmes (plusieurs classes qui interagissent) et aussi valider la qualité de votre développement. Un autre objectif est de vous permettre de pratiquer et mieux assimiler les notions de la programmation par événements.

C**ontexte**

Ce projet se situe dans le contexte de villes intelligentes relativement à une mobilité minimisant la congestion. Plusieurs études ont montré que lorsque des conducteurs cherchent un stationnement, ils accentuent la congestion dans une ville car ils doivent rouler doucement et font ralentir la circulation le temps de rechercher et regarder s’il y a des places disponibles. Une solution a cette situation consiste à informer les utilisateurs concernés de la disponibilité des places du stationnement le plus proche dès qu’ils font une demande dans une ville donnée.

Le projet consiste à simuler cette application via un modèle objet restreint qui considère une portion d’une ville (un secteur avec quelques stationnements et quelques intersections, ..).

# Objectif du fonctionnement du système à base d’objets

Une fois qu’une zone d’une ville est modélisée (routes, stationnements, intersections, ..), on veut que l’application qui sera basée sur le modèle d’une ville aide les conducteurs qui font une requête de stationnement pour trouver le mieux (du point de vue localisation).

Ceci consiste alors à surveiller les états des stationnements en continue, connaitre leur localisation et être capable d’identifier le meilleur stationnement qui contient des places libres aux conducteurs qui ont l’intention de stationner.

Un calcul de chemins sera aussi requis pour choisir le stationnement le plus proche du conducteur qui a fait une demande de stationnement via l’application. On suppose qu’il y a souvent des places disponibles compte tenu des mouvements des stationnements. L’application fonctionne correctement si elle est capable d’aider les conducteurs en ayant les informations nécessaires.

****

**Figure 1 : Stationnement intelligent géré par une application**

Diagram

Description automatically generated

**Figure 2 : Communications du stationnement intelligent**

**Travail à faire dans un environnement objet**

# Phase 1 : Analyse : 10 pts

Assurez-vous d’avoir bien saisi l’énoncé et rédigez un paragraphe représentant les besoins et la manière dont fonctionnera votre application. Un premier schéma global peut donner une idée sur le futur fonctionnement.

# Phase 2 : Conception de la solution : 25 pts

## a) Les interfaces graphiques : 10 pts

Les interfaces à concevoir concernent les stationnements, les routes de la zone d’une ville, etc. On demande de concevoir des interfaces qui seraient adéquates pour chaque composante du système global selon le fonctionnement décrit précédemment. Commencer par faire la conception sur papier d’abord et ensuite présenter votre solution sous forme d’un graphe (Avec ou sans un outil comme Visio) pour les différents types d’interfaces que vous avez identifiées. Utiliser un graphe avec les localisations de divers stationnements, les routes associés, la distance entre stationnements et intersections de la zone d’une ville, etc.

Par la suite, il faut identifier dans la librairie jdk de java les composantes graphiques qui correspondent au mieux aux besoins de vos interfaces. Définissez un tableau qui présente chaque composante et sa documentation essentiellement sa fonction et quelques méthodes dont celles qui seront utilisées selon votre conception (selon le travail fait en classe).

Présentez ce tableau dans vos livrables.

## b) Conception de la solution : 15 pts

Pour définir la solution globale de ce système, on a besoin d’identifier les classes (incluant leurs attributs, méthodes) et leurs relations. Vous pouvez faire un graphe simple ou utiliser un outil comme UML ou Visio pour présenter les classes et leurs relations.

Pour les traitements, considérer particulièrement le calcul du plus court chemin entre la localisation du conducteur et chaque stationnement candidat. Il faut représenter un graphe et associer un algorithme qui permet de faire le calcul. Il est important d'expliquer clairement le fonctionnement de votre algorithme. Vous pouvez utiliser des hypothèses pour représenter différents paramètres de votre calcul selon le modèle que vous avez choisi pour cette application.

# Phase 3 : Implémentation de votre solution : 25 pts

Codez la solution conçue incluant les classes des deux phases (application et interface) et bien documentez votre code par des commentaires significatifs selon le guide des conventions en classe.

# Phase 4 : Tests de qualité de votre système : 15 pts

En utilisant les documents du cours (révision, qualité de la documentation), élaborer des guides sous forme de tableau à remplir afin de s’assurer de la qualité de votre solution (depuis l’analyse jusqu’à son codage). Choisir au moins deux méthodes de révision vues en cours et impliquant une équipe et présenter les détails de l'applications.

Faire plusieurs jeux de tests pour faire varier les requêtes des conducteurs vs les mouvements des stationnements de la ville (zone), leurs états et les disponibilités possibles. Montrer pour chaque jeux les interfaces associées. Rédiger les réponses des tests de fonctionnement et aussi de la validation de la qualité que vous rajouter au même document avec l’analyse (phase1) et la conception (phase 2).

# Travail d'équipe : 10 pts

Il s’agit d’un travail **d’équipe de quatre étudiants.** Quinze équipes doivent être formés.

Cette section présente la composition de l'équipe, les détails des tâches, des interactions, toute autres informations soulignant la qualité de l’équipe. Chaque phase du développement doit être considérée relativement aux activités des membres d'une même équipe.

# Références : 5 pts

Toute information (code, algorithme ou autre), issue de la littérature ou d’un site **doit être référencée clairement** et présentée dans la liste des références du document à joindre et qui peut inclure des références de la liste fournie si celles-ci sont utilisées.

# Présentation des résultats : 10 pts

Les présentations des résultats de chaque équipe auront lieu le 5 avril et le 12 avril. Des détails additionnels seront communiqués. (Groupe 1 et groupe 2 à déterminer)

# Livrables et modalités du travail

* L’énoncé est disponible **le mercredi 22 février 2023.**
* Date de remise du travail : **mercredi 5 avril.** (Aucune extension ne sera accordée)

* Remettre le code source, l’exécutable et le document Word (compte rendu). Mettre tout dans un dossier que vous compressez en **.zip** (pas rar). Le dossier doit porter le ***nom* d’un membre de l’équipe.**

- Les modalités de transmission du travail seront rajoutées.

**Quelques références utiles pour les composantes graphiques :**

[http://java.sun.com/products/jfc/tsc/articles/painting/index. htmlhttp://java.sun.com/products/jfc/tsc/articles/painting/index.html#background](http://java.sun.com/products/jfc/tsc/articles/painting/index.%20html) <http://java.sun.com/docs/books/tutorial/uiswing/index.html>

<http://fr.wikipedia.org/wiki/Algorithme_de_Dijkstra>

<http://interstices.info/jcms/c_15578/le-plus-court-chemin>