CAHIER DES CHARGES STRUCTURE DE DONNÉES

NB : quand je mets des choses entre parenthèse quand je parle d’une fonction ou d’une méthode c’est ce que ça renvoie, pas ses paramètres !!!

1. Ce qui faut faire
2. Structure
3. Création de la base du projet :

* Les Heather (les différentes classes)
* Les fonctions de bases pour chacune des classes : celles pour avoir accès facilement aux membres privés, constructeur, constructeur par recopie, destructeur si besoin (seulement dans ligne)

1. Méthodes propres à chaque classe : arrêt

* Définition ses méthodes de base sont nécessaires pour accéder aux membres privés (les get…) :

Nom (string)

Durée min (int)

Position (structure position qui prend deux int)

* Les méthodes propres à cette classe (ne doit pas s’appeler get… car est calculé) :

Estunterminus (bool) : test si la position de l’arrêt est égale à la position de d\_queue ou d\_tete de la ligne sur laquelle se trouve le tramway.

* Un chainon d’une liste doublement chainée donc c’est important d’avoir un pointeur sur le chainon précèdent et le suivant.
* Mettre les deux autres classes en ami pour pouvoir l’utilisé partout.
* Écriture des méthodes du premier tiret.

1. Méthodes propres à chaque classe : tramway

* Définition ses méthodes de base sont nécessaires pour accéder aux membres privés (les get…) :

Vitesse actuelle (bool),

Vitesse maximal (int)

Numéro ligne (int)

Position (structure position qui prend deux int)

Identifiant (int)

Sens de déplacement (bool)

Position par rapport aux arrêts (bool : à un arrêt ou non)

Tempsarret (int) : à ajouter dans le pv et faire une méthode pour le donner, il sera incrémenté si position par rapport aux arrêts est vrai et remis à 0 dès que position par rapport à un arrêt est faux

* Les méthodes propres à cette classe (ne doit pas s’appeler get… car est calculé) :

Distance avec le prochain tram (double) : calcule un double en fonction de la position du tram courant et du tram suivant

Distance avec le prochain arrêt (double) : calcule un double en fonction de la position du tram courant et de l’arrêt vers lequel il se dirige

Peut avancer (bool) : vérifie si le tram peut avancer :

°si la position est la même et (facultatif c’est évident : que le sens de circulation est différant) et que les deux sont à un terminus 🡪 renvoie faux

°si la position est la même et que ce n’est pas un terminus 🡪 revoie vrai

°si le sens de circulation est le même, calculer la distance entre les tramways et si elle est > a la distance minimum (qui se trouve dans le privé) 🡪 renvoie vrai sinon 🡪 renvoie faux

* Un chainon d’une liste simplement chainée donc c’est important d’avoir un pointeur sur le chainon suivant.
* Mettre la classe ligne en ami pour pouvoir l’utilisé.
* Écriture des méthodes du premier et deuxième tiret.

1. Méthodes propres à chaque classe : ligne

* Définition ses méthodes de base sont nécessaires pour accéder aux membres privés (les get…) :

Nom liste tram (vector de int) : changer le void

Nom liste arrêt (vector de string) : changer le void

Distance entre arrêt (vector de double)

Insérer arrêt (arret&) : en fonction de la position de l’arrêt à insérer

Supprimer arrêt (void) : en fonction de sa position ou de son nom à voir à définir

Rechercher arrêt (arret&) : en fonction de son nom ou de sa position à définir nullptr si par trouvé

Insérer tram (tramway &) : en fonction de son identifiant

Supprimer tram (void) : on recherche l’identifiant à supprimer

Rechercher tram (tramway &) : en fonction de son identifiant nullprt si pas trouvé

Tete arret (\*arret) : nécessaire pour la class arrêt savoir si c’est un terminus

Queue arret (\*arret)

* La liste simplement et doublement chainée donc c’est important d’avoir un pointeur sur la tête et la queue des arrêts et la tête des trams.
* Écriture des méthodes du premier et deuxième tiret.

1. Choses générales à gérer

Lié les trams et l’arrêt qui le suit qui est gardé dans sa classe. Il faudrait déjà le mettre dans le constructeur car il faut que sa position au début soit celle d’un arrêt donc facile de savoir sur quel arrêt on pointe. Dès qu’on quitte cet arrêt il faudrait que ce qui est dans les paramètres passe au suivant (dans le cas ou le sens est 1 et le précèdent quand le sens est 0) c’est-à-dire au tram où il est sensé bientôt arrivé

Il faudrait initialiser le « tempsarreter » à tempsminarreter - 2 dans le constructeur pour qu’au départ ils n’attendent pas trop longtemps les trams aux arrêts.

1. Contenu du main

Voir b et c

1. Interface

Ce qu’on doit respecter :

* Lignes sous forme de segments
* Tram sous forme de carrés
* Coordonnées 2D
* La durée de la simulation est définie par l’utilisateur
* La fenêtre se ferme quand c’est fini
* Les trams doivent être placés aux arrêts au début
* Toutes les secondes ou ms ont doit tout re-tester pour faire avancer chaque tram, actualiser son sens, son arrêt suivant, sa distance avec tram devant lui,…

1. Documents à lire

* Faire des fonctions dans le main permettant pour pouvoir lire des documents textes types contenant les informations d’un

arrêt 🡪nom ; duréemin ; position x et y.

tramway 🡪 id ; numligne ; vMax ; position x et y(attention doit correspondre à celle d’un arrêt sinon pas possible) ; sens(0 ou 1) ; distance min tram suiv ;

* Pas besoin d’écriture dans ces fichiers.
* Faire deux ou trois groupes fichers (sachant qu’il faut 2 fichiers différants par ligne) de ce type et on pourrait faire une petit interface utilisateur où l’on le demande avec quel fichier il veut effectuer la simulation (genre un fichier avec 2 lignes 5 trams et 10 arrêts ou un avec 2 lignes 3 trams et 6 arrêts par exemple). Jsp si c’est utile mais je trouve ça peut être un petit truc cool à faire pas trop compliqué pour nous démarquer des autres à voir

1. Documentation et aspect du programme

* Commenter chaque fonction du programme pour que ce soit plus facile à lire pour le correcteur.
* Mettre les méthodes toujours dans le même ordre comme vu en cours (constructeur et constructeur par recopie en premier dans le .h et le .cpp pour faciliter la lecture)

1. Rapport texte

* Expliquer le fonctionnement de l’interface graphique
* Expliquer l’organisation du programme (organigramme schéma)
* Expliquer les choix de l’organisation
* Définir la contribution de chacun

1. Présentation power point

* 5 min en tout
* Jsp ce qu’il faut dire ? notre organisation ? nos difficultés ?

1. Ce qui reste à faire
2. Structure
3. Création de la base du projet : fonctions de base : constructeur tramway,

Constructeur et constructeur par recopie tramway : ajt position arret initialiser à vrai, et voir tout le reste en fonction du privé il faudrait que tout soit renseigné (ex vactuelle aussi a 0)

Constructeur par recopie de ligne (s’il est nécessaire jsp à vous de voir)

1. Méthodes propres à chaque classe : arrêt : Estunterminus (bool) 🡪 .h et .cpp
2. Méthodes propres à chaque classe : tramway : Vitesse maximal (int) 🡪 dans le .h et .cpp

Sens de déplacement (bool) 🡪 .h et .cpp

Position par rapport aux arrêts (bool : à un arrêt ou non) 🡪 .h et .cpp

Tempsarret (int) 🡪 .h (pv, get… existe déjà mais n’est pas écrit) et .cpp

Distance avec le prochain tram (double) 🡪 changer nom .h + vérif. si ça marche

Distance avec le prochain arrêt (double) 🡪 changer nom .h + vérif. si ça marche

Peut avancer (bool) 🡪 modifier dans le .cpp avec les critères vus plus haut et regarder comment faire la gestion du temps dans cette fonction (pour savoir si le tram est arrêté depuis assez lgt à l’arrêt pour pouvoir repartir)

1. Méthodes propres à chaque classe : ligne

* Définition ses méthodes de base sont nécessaires pour accéder aux membres privés (les get…) :

Nom liste tram (vector de int) 🡪 changer le void

Nom liste arrêt (vector de string) 🡪 changer le void et attention .h et .cpp nom de la fonction est différente et mettre un return si il n’y a pas de tête

Distance entre arrêt (vector de double) 🡪 changer le void et le type qui est dans <> (pv)

Insérer arrêt (arret&) 🡪 changer le void et retourner l’adresse du nouvel arrêt

Supprimer arrêt (void) 🡪 a faire dans .cpp

Rechercher arrêt (arret&)  🡪 changer le bool en une adresse d’arrêt

Insérer tram (tramway &)  🡪 changer le void et retourner l’adresse du nouveau tram

Supprimer tram (void) 🡪 FINI

Rechercher tram (tramway &) 🡪 a faire dans .cpp

Tete arret (\*arret) 🡪 .h et .cpp

Queue arret (\*arret) 🡪 .h et .cpp

1. Pas encore fait tout doit être fait
2. Interface

Rien n’est encore fait

1. Documents à lire

Rien n’est encore fait

1. Documentation et aspect du programme

Rien n’est encore fait

1. Rapport texte

Presque tout doit encore être fait

1. Présentation power point

Rien n’est encore fait

1. Qui fais quoi maintenant

Je propose, à débattre sur discord si vous voulez échanger les tâches

1. Structure

Je propose que je fasse le 1) et le 3)

Alban le 2) le 4) et le 5)

1. Interface

Nour tu es celle qui à l’air le moins perdu face à ça et ton pote avais l’air de bien savoir le faire donc tu es celle (je crois) qui y arrivera le plus facilement de nous 4 donc go focus toi là-dessus

1. Documents à lire

Heuss t’avais l’air de pouvoir le faire rapidement et bien te souvenir de comment on le fais donc go vasy

1. Documentation et aspect du programme

Ça je kifferais bien le faire genre une petite touche finale pour que ce soit tout beau tout bien fais genre une espèce de charte graphique sur l’écriture du code pour que ce soit plus agréable à lire et compréhensible

1. Rapport texte

Expliquer le fonctionnement de l’interface graphique : je pense nour comme elle l’aura fait elle pourra plus facilement l’expliqué au moins les idées générales et ce qui est de la forme je peux réécrire moi par la suite.

Expliquer l’organisation du programme (organigramme schéma) : je pense que Alban et heuss vous pouvez vous le partager a mon avis vous kiffez bien faire des beaux schémas et tt donc faites-vous plaiz

Expliquer les choix de l’organisation : je peux le faire moi

Définir la contribution de chacun : ça aussi je peux le faire moi

1. Présentation power point

Ça se serait cool qu’on le fasse tous ensemble mais pour l’instant c’est notre dernier problème donc on verra plus tard

1. Remarque

Est-ce que les tramways ne devraient pas être dans une liste chainée circulaire ?