

Projet de second semestre de programmation orienté objet:

IsochroneTL

Description des bonus:

Ce document explique les différents bonus que nous avons rajouté à notre projet:
Les trois premiers bonus rajoutés sont similaires à ceux décrits dans la documentation du projet.

Animation:

Comme son nom l'indique cette fonction permet d'animer la carte en changeant différents paramètres via un menu déroulant (mis en place via un JComboBox). Celui-ci permet de choisir entre plusieurs types d'animation:

- minute par minute, qui incrémente minute par minute l'heure actuelle.
- heure par heure, qui incrémente heure par heure l'heure actuelle.
- et arrêt aléatoire, qui choisit un arrêt au hasard dans le JComboBox d'arrêt et le met comme arrêt de départ.

Les animations commencent en prenant l'arrêt et l'heure de départ donné par l'utilisateur (il peut donc la modifier si voulu avant l'animation), celle-ci se remettra au paramètre avant animation. On utilise un timer qui arrive à son terme toutes les millièmes de secondes (on a alors l'impression que l'image change en continu) pour effectuer ces modifications.

Affichage du trajet:

La seconde fonctionnalité ajoutée au projet permet à l'utilisateur de visualiser son trajet en arrêtant sa souris sur un point pendant plus d'une demi seconde. Le visuel du trajet affichera en bleu et blanc la partie du trajet où l'utilisateur peut "utiliser" les TL (malgré le fait que si celui-ci marche entre deux trajets cela ne peut pas être détecté). Ensuite lorsque l'utilisateur sort de la zone couverte par les TL, le trajet affiche une route directe (à pied donc) et de couleur noir.

On utilise pour cela une nouvelle classe, PathTileProvider, qui dessine, en fonction d'un FastestPathTree et d'une position (où la souris est placée), les différentes lignes.

On affiche également dans un JPanel, le temps de trajet et l'heure d'arrivée. Ce JPanel devient invisible quand le trajet n'est plus affiché.

Changement de l'arrêt de départ via la souris:

Nous avons ensuite permis à l'utilisateur de choisir son point de départ indépendamment des arrêts proposés par la liste fournie par les transports Lausannois. Celui-ci peut donc, via le clic droit de la souris, modifier son arrêt de départ, l'algorithme cherchera ensuite l'arrêt le plus proche et effectuera la recherche à partir de celui-ci.

Taille minimum:

De base, le modèle fourni ne limite pas la taille minimum de la fenêtre, celle-ci peut donc être réduite, cachant donc une partie voire la totalité des menus situés dans la partie supérieure de la fenêtre.

Navigateur au clavier:

Ayant programmé le projet sur des machines portables munies de Trackpad et non de souris, nous nous sommes vite rendus compte que selon la manière dont celui-ci avait été réglé par l'utilisateur l'affichage s'en trouvait fortement modifié, surtout au niveau du zoom où il est parfois difficile de zoomer "lentement" via le trackpad.

Nous avons donc choisi de permettre à l'utilisateur d'interagir avec le programme via les touches du clavier:

- les flèches et les touches WASD servant à se déplacer.
- les touches QE respectivement à dézoomer et à zoomer.

Néanmoins, il faut que le CenterPanel possède le focus et donc il faut presser trois fois Tab pour ensuite pouvoir se déplacer avec le clavier.

Anti-Aliasing :

Nous avons également ajouté de l'anti-aliasing pour les tuiles isochrones pour que le résultat soit plus esthétique.