

Introduction

Une grande firme d'ingénierie fait appel à votre équipe pour développer une application (RecyclApp) qui permet de faire le design d'un centre de tri.

Un centre de tri est une usine qui reçoit en entrée le contenu (pêle-mêle) des bacs à recyclage. Différentes machines et/ou stations manuelles, permettent de séparer le contenu en différentes catégories/produits. Dépendamment des équipements (stations) choisis et de la façon dont elles seront inter-connectées, on obtiendra en sorties des ballots de matière plus ou moins purs.

La conception d'un centre de tri en fonction des exigences du client (i.e. du taux de pureté des extrants) est une tâche extrêmement complexe : une multitude d'équipement doivent être sélectionnés de manière à ce que le design final permette la gestion du flux de matière sans voir apparaître des goulots d'étranglement. Actuellement, la conception de ces centres est réalisée de manière traditionnelle (par essai et erreur) par des ingénieurs d'expérience. Les plans sont réalisés sous Visio et les calculs avec Excel. Ce travail est lourd et fastidieux.

L'application que vous développerez permettra aux concepteurs de faire le design du centre de tri de manière interactive. Le concepteur (utilisateur) pourra définir le panier de produit consommé par l'usine (kg/h de chaque produit entrant), placer et déplacer les équipements sur le plan (à l'aide de la souris), spécifier le comportement de chaque équipement/machine/station, connecter les sorties et entrées des machines entre elles à l'aide de convoyeurs. Il pourra aussi voir les quantités en kg/h de chaque produit passant en tout point du réseau.

Description détaillée

L'application utilisera la souris pour placer les stations (équipements de tri) dans le plan et les connecter entre eux à l'aide d'arcs (représentation d'un convoyeur). Il est également possible de placer des points d'entrée de matière, des points de sortie de matière (sorties de l'usine) et des points de jonction (représentant la fusion de plusieurs convoyeurs).

Des fenêtres permettront de définir les paramètres des différents éléments. Par exemple, pour une entrée, on peut définir les quantités (Kg/h) de chaque produit qui entre par ce point.

Pour une station on peut notamment définir un *nom*, une *description*, le *nombre de sorties* de la station. On peut ensuite définir une matrice de récupération spécifiant, pour chaque produit en entrée, le « pourcentage » de ce produit qui se retrouve à chacune des sorties. Il y a une sortie par défaut pour chaque station. Les produits pour lesquels on n'a pas spécifié de pourcentage se retrouvent par défaut à 100% dans cette sortie.

Chaque arc (ou convoyeur), relie une sortie d'un nœud (station, entrée, sortie, jonction) à l'entrée d'un nœud. Une jonction est le seul type de nœuds qui peut avoir plusieurs arcs entrants.

Un arc est visuellement représenté par une ligne avec une flèche pleine à son extrémité. Chaque arc peut être

d'une couleur différente, choisie par l'utilisateur.

Une station est visuellement représentée par une boîte rectangulaire (d'une couleur au choix de l'utilisateur). On y affiche certaines informations, par exemple son nom. On peut également décider d'afficher une image représentant la station. L'utilisateur décide si l'ensemble du schéma doit afficher l'information textuelle ou les images. Dans tous les cas le cadre du rectangle est toujours présent. Chaque sortie de la station est identifiée par un petit no.

En cliquant sur (ou en passant la souris au-dessus?) un élément (nœud ou arc), il est possible de voir toute l'information se rapportant à cet élément (quantité des produits en transit, etc.)

Pour chaque arc on peut notamment connaître les quantités des différents produits qui transitent par celui-ci. On peut également obtenir cette information pour un nœud donné (il s'agit de la somme des arcs entrant).

Toute l'édition doit se faire en mètres (et non pas en pixels). Il est possible de zoomer/dézoomer lors de la visualisation. Une grille peut être activée par l'utilisateur pour faciliter l'édition (l'utilisateur spécifie la distance en mètres entre chaque point). On peut rendre la grille « magnétique » (dans ce cas on peut placer un élément seulement sur un point de la grille, pas entre deux points). Lors de l'insertion d'une image, on spécifie sa taille en mètres. Lors du déplacement de la souris dans le plan, on peut avoir en tout temps les coordonnées cartésiennes (en mètres) associée à la position de la souris ou de l'élément sous la souris.

Concepts avancés

Certaines stations particulières ne réalisent pas un tri à proprement parler; elles transforment plutôt certain des produits en entrée en un certain nombre d'autres produits (exemple naïf : transformer des « gros » morceaux de plastique en un certain nombre morceaux de taille moyenne et petite). Encore une fois, une matrice permettra de définir quels pourcentages d'un produit donné est transformé en d'autres produits.

Le schéma peut être exporté sous la forme d'une image.

Il est possible d'enregistrer / charger son travail dans un fichier. On demande également un mécanisme permettant d'annuler/répéter des opérations lors de l'édition du réseau (undo/redo).

Chaque station et arc a une capacité maximale en kg/h. Le système doit clairement détecter si le système modélisé est correct (tout est connecté, les taux d'utilisation des équipements est inférieur à 100%, etc.) et offrir une rétroaction à l'utilisateur.

Consignes à propos du projet

- Le travail doit être réalisé en équipe de 4.
- Le forum peut être utilisé pour trouver des coéquipiers. Chaque équipe doit se trouver un **nom d'équipe constitué d'un seul mot**. Une fois l'équipe constituée, vous devez en aviser Martin

Savoie (martin.savoie.2@ulaval.ca) comportant le nom de l'équipe, et la liste des Noms, Prénoms, IDUL.

- Le projet doit être réalisé en Java
- L'environnement de développement est NetBeans
- Il est possible d'utiliser l'environnement Eclipse si tous les membres de l'équipe sont d'accord et que vous obtenez l'autorisation écrite du professeur.
- L'utilisation de toute autre librairie autres que les librairies standard Java 6 / 7 / 8 est interdite à moins d'obtenir l'autorisation du professeur par courriel.
- Les diagrammes UML doivent être produits avec Visual Paradigm.

Remarques

Certains éléments du descriptif de projet **sont volontairement flous à ce stade** (si nous vous transmettions des spécifications parfaites accompagnées de diagrammes UML... vous n'auriez pas à faire l'analyse et ce ne serait plus un projet complet).

Il vous appartient de faire la lumière là-dessus et de développer une bonne compréhension du projet. Vous serez appelé à poser des questions, en classe de même qu'à vos « conseillers » (Martin Savoie et Guillaume Lorquet) qui vous accompagneront tout au long du projet. Si vous doutez de quelque chose n'hésitez pas à poser des questions.

Livrable no 3

Votre client est impatient de voir votre avancement dans le projet et aimerait que vous lui fassiez une démonstration afin de s'assurer que vous êtes dans la bonne voie. Il exige que les fonctionnalités suivantes soient opérationnelles lors de la démonstration :

- **EntréeUsine.** À l'ouverture, on a déjà un **point d'entrée de matière** (EntréeUsine) qui reçoit le contenu (pêle-mêle) des bacs à recyclage. Aux fins de la démo, on suppose deux types de matière P1 et P2 qui ont chacun un taux d'arrivée de 1000 kg/h (on n'exige donc pas que ce soit éditable pour la démo). On exige cependant pouvoir déplacer cet équipement à volonté à l'aide de la souris. **(5%)**
- **SortieUsine.** On doit pouvoir, à l'aide de la souris, **ajouter des points de sortie de matière** (SortieUsine) à l'endroit désiré. **(5%)**
- **Jonction.** On doit pouvoir, à l'aide de la souris, **ajouter des points de jonction** (représentant la fusion de plusieurs convoyeurs) à l'endroit désiré. **(5%)**
- **Station.** On doit pouvoir, à l'aide de la souris, **ajouter des stations à l'endroit désiré**. Lors de l'ajout, on nous demande de spécifier combien il y a de sortie sur cette station. **(5%)**
- **Édition station.** On doit pouvoir, à l'aide d'un clic (ou double-clic) de la souris, avoir accès à une interface permettant de **spécifier les caractéristiques et comportements de chaque station (15%)**
 - Nom/description
 - La capacité de traitement maximale (en Kg/h) de la station
 - Matrice de récupération :
 - Les quantités en kg/h de chaque produit entrant (en lecture seule)
 - Le pourcentage, pour chaque produit en entrée, qui se retrouvera à chacune des sorties (pour un produit donné, la somme des sorties doit donner 100%)
 - Une cellule vide correspond à une valeur de 0 %
- **Convoyeurs.** On doit pouvoir, à l'aide de la souris, **connecter des équipements** (Station, EntréeUsine, SortieUsine, Jonction) entre elles à l'aide de convoyeurs, conformément aux spécifications **(15%)**
 - Une jonction est le seul type de nœuds qui peut avoir plusieurs arcs entrants
 - Puisqu'une station peut posséder plusieurs sorties, l'utilisateur rattache chaque convoyeur sortant d'une station à l'une de ses sorties en particulier, et il ne peut y avoir plus d'un convoyeur rattaché à une sortie d'une station.
- **Affichage (25%).**
 - On doit pouvoir zoomer/dézoomer lors de la visualisation
 - On doit pouvoir activer une grille pour faciliter l'édition
 - L'édition doit se faire en mètres (et non pas en pixels)

- On peut rendre la grille « magnétique »
- Les différents types d'équipements sont affichés/représentés de manière distincte et le nom de celle-ci est affichée sur le plan.
- Un convoyeur est représenté par une ligne avec une flèche pleine à son extrémité.
- On doit pouvoir visualiser, **en tout temps**, les coordonnées cartésiennes (en mètres) du déplacement de la souris dans le plan
- **Calculs.** On peut voir les quantités en kg/h de chaque produit/matière passant en tout point du réseau... (le tout correctement calculé par l'application!) **(25 %)**

Éléments minimums à inclure dans le rapport :

- Diagramme(s) de classe de conception et textes explicatifs mis à jour. Les principales méthodes doivent apparaître de manière à ce que votre design puisse être compris.
- Diagramme(s) de séquence demandés au livrable #2 et textes explicatifs, mis à jour.

Consignes supplémentaires :

- Pour avoir les points associés à une fonctionnalité donnée, celle-ci doit être opérationnelle (lors de la démo) et être correctement documentée dans le rapport (lorsqu'un ou des diagramme(s) correspondant à cette fonctionnalité était demandé).
- Une correspondance/cohérence raisonnable entre les différents diagrammes est nécessaire.
- Les diagrammes doivent être produits avec Visual Paradigm.
- Vous devez fournir votre code Java qui sera compilable et exécutable sous NetBeans ou Eclipse (pour les équipes autorisées à utiliser Eclipse).
- Une correspondance/cohérence raisonnable entre les différents diagrammes et le code est attendue (et sera vérifiée).
- Assurez-vous que si on change la résolution de l'écran votre application fonctionne quand-même
- On vous rappelle que l'utilisation du GRASP Contrôleur est obligatoire.
- Vous devez soumettre un seul fichier .zip appelé **NomEquipe-tp3.zip** contenant ceci :
 - **NomEquipe-tp3.pdf**
 - **NomEquipe-tp3.vpp**
 - **NomEquipe-tp3.jar** constituant une version exécutable de votre application qui sera utilisée pour la **démo qui aura lieu en classe ou sous Adobe Connect (un courriel d'information suivra).**
 - Un sous-dossier contenant le **projet NetBeans ou Eclipse.**

Amusez-vous bien!