

Livrable no 2 – Modèle de conception et Architecture logique

On vous demande de produire une première version de votre **modèle de conception** et de votre **architecture logique**. Au minimum, votre rapport devra inclure :

1 Le diagramme de classe de conception

de même qu'un texte explicatif décrivant les classes (mettre les noms des classes en gras dans le texte) et leurs relations. Voici quelques contraintes à respecter au niveau de votre design :

- Il est obligatoire d'appliquer le principe du contrôleur de Larman (**attention, ce n'est PAS la même chose qu'un MVC**).
- Prévoir dans votre design une classe ou un package « Afficheur » chargé de dessiner la vue 2D du projet en utilisant les classes de Swing (Graphics, notamment). C'est ce qui permettra de dessiner votre plan dans votre fenêtre principale. Voyez l'exemple à ce sujet dans le Wiki du cours.

2 L'architecture logique.

Produisez un diagramme de packages décrivant votre architecture logique. Accompagnez-le d'un texte explicatif. Sur le diagramme de classe de conception, tracez également vos package afin de montrer quelle classe appartient à quel package.

3 Les diagrammes de séquence de conception

et un texte explicatif (expliquant le schéma) pour chacun des éléments suivants:

3.1 **Déterminer l'élément sélectionné lors d'un clic de souris dans la vue du plan de la carrière. Tracez deux diagrammes distincts :**

- Le premier diagramme débute par un appel externe fait par Java lors d'un clic de souris dans le JPanel où est affichée votre vue en plan. Montrez le travail réalisé pour déterminer les coordonnées x,y (en unités de mesure, et non pas en pixel) correspondant au clic de souris. Indices : Pour tracer ce diagramme, vous devrez avoir compris comment fonctionne le système d'événement de Java et décidé comment se fait la gestion du zoom dans votre application. Vous ne pouvez pas réussir ce diagramme si vous n'avez pas fait un minimum de prototypage. Accompagnez d'un texte explicatif.
- Le second diagramme débute par un appel au contrôleur auquel on passe les coordonnées (en unités de mesure) du point sur lequel l'utilisateur a cliqué. On doit voir les traitements permettant de déterminer de manière effective quel élément on vient de cliquer. Accompagnez d'un texte explicatif.

3.2 **Montrez comment l'affichage du plan sera réalisé.**

Spécifiez quel événement (Swing) déclenche la procédure d'affichage, la méthode appelée du côté de l'interface utilisateur, l'appel fait à votre « afficheur de plan » et tous les appels internes à la couche du domaine, l'utilisation de l'objet Graphics, etc. Notez bien qu'un exemple semblable (mais plus simple) est présent sur le Wiki du cours. Vous devez démontrer que vous avez compris comment fonctionne la gestion de l'affichage. Accompagnez d'un texte explicatif.

3.3 Ajout d'un nouvel équipement.

Le point d'entrée du diagramme est l'appel au contrôleur (on suppose que les valeurs de tous les paramètres nécessaires à sa création ont été au préalable définies via des interactions au niveau de la couche interface-utilisateur). Accompagnez d'un texte explicatif.

3.4 Ajout d'un nœud par l'utilisateur (en tenant compte de la gestion de la grille et du zoom).

Prenez bien soin de faire apparaître le contrôleur.

3.5 Ajout d'un camion.

Lors de la correction, pour chacun de ces diagrammes, nous nous poserons la question suivante : « si l'équipe programme sur la base des diagrammes proposés, est-ce que l'application va fonctionner et faire ce qui est demandé? »

4 Algorithme

Spécifiez en pseudo-code un algorithme qui permet de choisir un itinéraire entre la position actuelle du véhicule et une destination donnée (pour ce faire, vous pouvez utiliser, au choix, du pseudo-code, un diagramme d'activité, un diagramme de séquence ou autre). Indiquez aussi quel est le nom de cette méthode et la classe à laquelle elle appartient. On doit d'ailleurs la voir apparaître sur votre diagramme de classes de conception et normalement elle sera appelée par au moins un de vos diagrammes de séquence demandés précédemment.

5 Diagramme d'états

- Pour une transaction
- Pour le déroulement d'une simulation (indice: adoptez le point de vue de l'utilisateur qui contrôle/regarde la simulation à l'écran)

6 Plan de travail mis à jour

De plus, vous devez fournir un plan de travail (**Gantt**) mis à jour (voir consignes du livrable #1). Vous pourriez être amenés à revoir la séquence d'implémentation des cas d'utilisation que vous aviez prévus dans votre plan de travail fourni avec le livrable #1, considérant que le client souhaite qu'au minimum les fonctionnalités suivantes soient supportées par la version de votre application qui sera rendue disponible au livrable #3 (d'autres pourront s'ajouter d'ici la publication du troisième énoncé):

- Ajouter, déplacer et supprimer des équipements avec la souris;
- Ajouter, déplacer et supprimer des nœuds et des arcs avec la souris;
- Affichage de la carrière par le système ;
- Ajouter des tas, matériaux;
- Ajouter un camion;
- Ajouter un chargeur;
- Éditer les caractéristiques d'un élément sélectionné dans le plan;
- Exécuter, arrêter et changer la vitesse de la simulation;
- Grille magnétique et Zoom;
- Enregistrer, charger le projet dans un fichier;
- ...

1. Votre rapport doit aussi inclure une section **détaillant la contribution de chacun des membres de l'équipe** dans le cadre des travaux du livrable #2.
2. Finalement, votre remise #2 doit comprendre **une version fonctionnelle (compilable et exécutable) de votre projet de même que l'exécutable en .JAR**, implémentant AU MINIMUM le squelette de votre interface utilisateur, i.e. une fenêtre avec les différentes zones prévues pour l'affichage (même si ces zones ne contiennent rien) de même qu'une barre de boutons (et optionnellement un menu).

Votre rapport doit prendre la forme d'un **seul document continu (format PDF) contenant schémas (lisibles) et le texte**. Incluez également (**en annexe, c'est-à-dire à la fin du document**) les documents suivants issus du premier livrable (que vous aurez mis à jour, s'il y a lieu): vision, modèle du domaine, modèle des cas d'utilisation, glossaire.

Vous devrez également fournir le **fichier Visual Paradigm** (un seul fichier) qui doit obligatoirement être utilisé pour produire les schémas. La remise doit contenir un fichier nommé par exemple **equipe42.pdf** et **equipe42.vpp**.

Conseils :

- Les éléments demandés doivent être cohérents entre eux (ex : les méthodes présentes dans les diagrammes de séquence doivent se trouver dans les diagrammes de classe).
- Il est impossible de faire d'abord les diagrammes de classes, puis de faire les diagrammes de séquences sans revenir modifier les diagrammes de classes. En créant les diagrammes de séquences, vous découvrirez des erreurs de conception dans vos diagrammes de classes de conception que vous devrez corriger. Vous devez prévoir plusieurs « itérations » au sein de l'équipe avant d'arriver à des diagrammes satisfaisants qui pourront être remis dans votre rapport.
- Assurez-vous de ne pas confondre « Diagramme de séquence système (DSS) » et « Diagramme de séquence de conception ».

Consignes supplémentaires :

- Les diagrammes UML doivent être produits avec Visual Paradigm Community Edition.
- Votre remise doit être effectuée en utilisant le **dépôt GIT**
 - La procédure de remise vous sera envoyée en temps et lieu
- Une pénalité de 1 point (sur 15) est appliquée si les fichiers ne sont pas nommés de la bonne façon
- Une pénalité de 1 point (sur 15) est appliquée si les diagrammes ne sont pas lisibles dans le rapport et que le correcteur doit se rapporter au fichier Visual Paradigm pour faire la correction.
- Pour tout retard non motivé (voir plan de cours; motifs acceptables pour s'absenter à un examen), la note 0 sera attribuée.

Amusez-vous bien!

P.S. Votre travail n'est pas de faire un rapport. Votre travail est de faire le design de votre application. Le rapport sert à témoigner de la qualité de votre processus de conception.