

TP UML

« Jeu de société : jeu de l'oie »

Plan

- I. Sujet
- II. Méthodologie
 - Les spécifications (séance 1)
 - Le prototypage (séance 1)
 - La réalisation du logiciel (séance 2 et 3)
 - Génération du code (optionnel)

I. Sujet

Le jeu de l'oie. Source : Wikipédia http://fr.wikipedia.org/wiki/Jeu_de_l'oie.

« Le jeu de l'oie est un jeu de société de parcours où l'on déplace des pions (représentant les oies) en fonction des résultats de deux dés. Traditionnellement, le jeu de l'oie comprend 63 cases disposées en spirale enroulée vers l'intérieur et comportant un certain nombre de pièges. Le but est d'amener son oie en premier à la dernière case. (...) »

La règle de base est intangible. Le jeu se joue avec 2 dés. Un premier coup décide de celui qui va commencer. Une image d'oie signale les cases fastes disposées de 9 en 9. Nul ne peut s'arrêter sur ces cases bénéfiques et on relance alors les dés.

Qui tombe à 6, où il y a un pont, ira à 12. *d 6 → 12*

Qui tombe à 19, où il y a un hôtel, se repose quand chacun joue 2 fois. *piège*

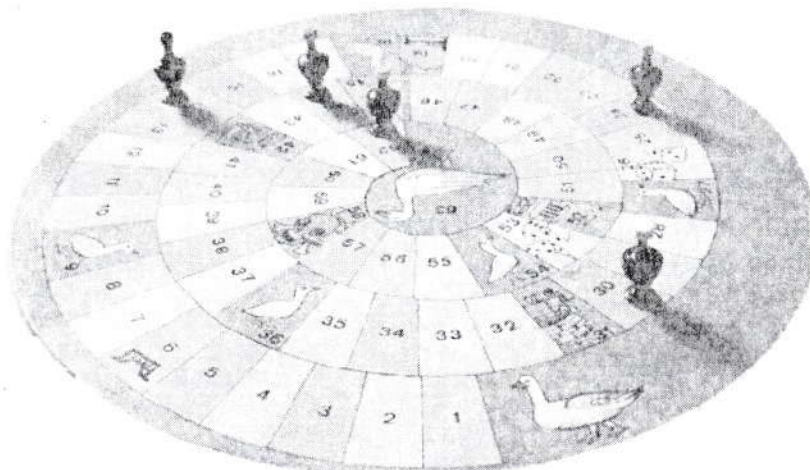
Qui tombe à 31, où il y a un puits attend qu'on le relève. *piège*

Qui tombe à 42, où il y a un labyrinthe retourne à 30. *d 42 → 30*

Qui tombe à 52, où il y a une prison attend qu'on le relève. *piège*

Qui tombe à 58, où il y a la mort recommence. *d 58 → 0*

Le premier arrivé à 63, dans le jardin de l'oie, gagne la partie, à condition de tomber juste sinon il retourne en arrière. » *fin partie*



II. Méthodologie

Nous travaillerons sur 3 aspects : les spécifications, le prototypage et la réalisation du logiciel.

Les spécifications (séance 1)

Nous allons tout d'abord travailler sur les spécifications du problème en abordant les aspects métiers.

Travail à effectuer :

- Dans modelio créer le paquetage : « Business view » et y créer un diagramme de classes (par exemple « matériel »).
- Elaborer le diagramme de classes en vous basant sur les différentes pièces nécessaires à ce jeu de plateau : « Le jeu de l'oie contient un plateau de 63 cases, 6 pions de différentes couleurs et deux dés. ».
- Sous le paquetage « Business view » créer un nouveau diagramme de classe « plateau ».
- Elaborer le diagramme de classes pour couvrir les différentes cases du plateau. **Attention** si vous avez besoin de classe du diagramme précédent faites les glisser depuis l'arbre vers votre diagramme.
- Vous pouvez apporter des détails en utilisant un diagramme d'objet et en créant les objets correspondants aux différentes cases du plateau.

Ces spécifications devront être validées avant de pouvoir poursuivre.

Le prototypage (séance 1)

Nous allons définir les entrées sorties du logiciel. Nous nous intéresserons au cas « jouerUnTour » qui débute quand c'est le tour de l'acteur de jouer et se termine quand il a fini. Le joueur peut ne pas jouer (si son oie est en prison, dans le puit ou à l'hôtel), jouer et tomber sur une case simple ou sur une case spécifique l'obligeant à recharger de case (labyrinthe, pont ou mort), il peut aussi rejouer s'il est sur une case fast ou enfin avoir gagné s'il est au jardin.

Travail à effectuer :

- Dans modelio, créer le paquetage : « Logical view » et y créer un diagramme de séquence (« diagramme de séquence système »).
- Créer directement à partir de l'arbre (onglet Model), en faisant un clic droit sur le paquetage « Logical view », un acteur « Joueur » et une classe « System ».
- Faire glisser l'acteur et l'objet dans le diagramme de séquence système.

- Renommer l'acteur avec votre prénom et rendre l'objet de type System anonyme. Pour cela sélectionner l'objet puis cliquer 1 fois dessus (pas de double clic).
- Le diagramme de séquence commence avec l'opération « jouerUnTour() » qui durera jusqu'à la fin du cas.
- Utiliser les différents fragments étudiés en cours et TD afin de spécifier les différents affichages du logiciel (par exemples : « Vous devez passer votre tour car vous êtes en prison », « Rejouer, vous êtes sur une case fast », ou « L'oie rouge avance jusqu'à la case 12 » ...).

Scénario : Le diagramme de séquence demandé commence lorsque c'est à votre tour de jouer. La première chose à déterminer c'est si votre oie est ou non bloquée sur une case. Par exemple si au tour précédent vous étiez tombé sur le puit, sur l'hôtel ou en prison vous devrez passer votre tour et le système devra afficher une phrase du type : « vous passez votre tour car vous êtes en prison ».

Par contre si votre oie n'est pas bloquée vous pouvez jeter les dés. J'ajoute la contrainte suivante : pour lancer les dés vous pouvez cliquer d'abord sur le premier dé puis le deuxième, ou inversement.

Vous allez pouvoir avancer votre oie du nombre correspondant à la valeur des deux dés. L'oie sera sur une case pouvant être de deux types différents : soit une case normale et vous afficherez un message du type : « vous êtes sur la case 2 » soit si c'est une case spéciale « vous êtes sur le pont ». Dans le deuxième cas il vous faudra exécuter la spécificité de la case. A la fin de l'exécution une nouvelle case sera retournée, soit il s'agit de la même case sur laquelle était votre oie avant d'exécuter la spécificité (ex : si vous étiez tombé en prison l'oie n'aura pas bougé), soit il s'agit d'une autre case (ex : si vous étiez tombé sur le pont votre oie se sera déplacée la case n°12). Dans ce dernier cas vous afficherez une phrase du type « et vous êtes maintenant sur la case n°12 ».

Il vous faut aussi vérifier si vous vous étiez déplacé (à la suite du lancer de dés) sur une case comportant le dessin d'une oie (correspondant à une case fast). Si c'est le cas vous devez relancer les dés.

Enfin si l'oie c'est déplacée sur le jardin alors il faudra afficher « Vous avez gagné ! ».

La réalisation du logiciel (séance 2 et 3)

Nous allons, à partir du diagramme de séquence système, spécifier le comportement interne du logiciel. Pour cela nous réfléchirons à partir du diagramme de séquence détaillé quels objets devront être créés, quelles opérations et quels attributs devront être implémentés.

Travail à effectuer :

- Sous le paquetage « Logical view », créer un diagramme de séquence « diagramme de séquence détaillé » et un diagramme de classes « diagramme de classes participantes ».
- Faire glisser l'acteur sur le diagramme de séquence.
- Au fur et à mesure que vous réfléchissez sur le diagramme de séquence au besoin d'un nouvel objet, d'une nouvelle opération, il vous faut la créer dans le diagramme de classes et l'utiliser dans le diagramme de séquence. Pour chacune des opérations que vous créez dans le diagramme de classe vous devez vous demander s'il la classe a besoin d'attributs supplémentaires ou de recevoir d'autres données (en envoyant un message à une autre classe) pour réaliser correctement l'opération.
- Attention de bien compléter le diagramme de classe participantes au fur et à mesure que le diagramme de séquence détaillé avance (association, agrégation, composition, généralisation ...).

Génération du code (optionnel)

- Une fois les deux diagrammes terminés vous pouvez cocher sous l'onglet JAVA (pour toutes les classes) la case Élément Java. Vous pouvez aussi déplacer toutes vos classes sous un paquetage (également un élément Java) nommé par exemple (moteurDeJeu). Enregistrer votre projet. Le code Java peut alors être généré (clic droit sur le projet, Java designer, Générer).
- Pour utiliser Eclipse pour coder les opérations :
 - o Télécharger depuis Moodle le fichier « javadesigner.jar » et le placer sous votre projet.
 - o Ouvrir Eclipse en donnant pour chemin le même workspace que celui de votre modélisation sous modélio.
 - o Sous Eclipse :
 - File/new.../JavaProject. ATTENTION : le nommer de manière identique au projet créé sous modélio.
 - Clic droit sur le projet « MonProjet », puis Properties, menu « Java Build Path », onglet « Libraries ». Si « javadesigner.jar » n'apparaît pas dans la liste alors l'ajouter : bouton « Add JARs... ». Sélectionner le fichier « javadesigner.jar » sous « MonProjet » et Valider.

Modelio

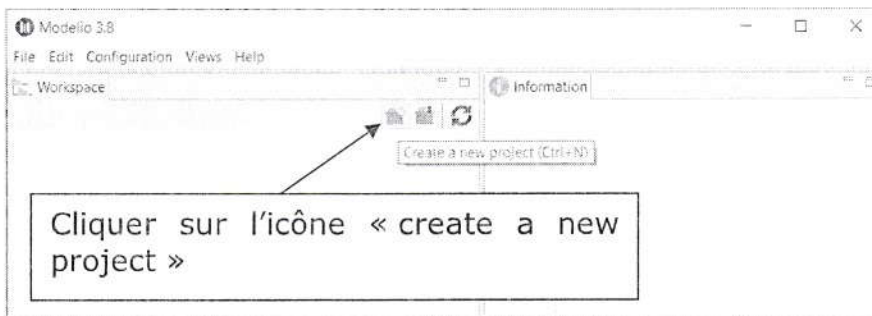
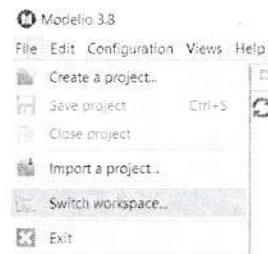
Plan

- I. Créer un projet
- II. Diagramme de classes
- III. Diagramme de séquence

III. Créer un projet

Lancer Modelio puis faire : « File / Switch workspace »

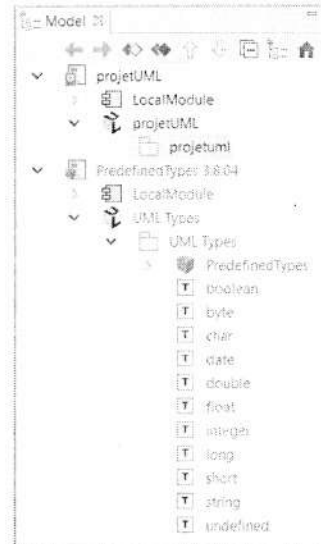
Choisir le chemin jusqu'à un dossier que vous nommerez OBLIGATOIREMENT workspace (au besoin utiliser le bouton « créer un nouveau dossier » :



Nommer le projet (exemple : projetUML mais trouver un nom plus explicite illustrant votre projet !) et cocher « Java project »



Vous obtenez l'arbre ci-contre et vous êtes positionnés sur un diagramme de classe.



Vous pouvez à présent créer vos paquetages pour organiser votre modélisation. Clic droit sur votre projet puis:



IV. Diagramme de classes

Pour créer le diagramme, clic droit sur le paquetage puis : create diagram...

class attribut opération

Onglet Properties

Pour créer les paramètres d'une opération la sélectionner dans l'arbre puis clic droit.

Property	Value
Name	attribute
Type	string
Visibility	Public
Multiplicity min	1
Multiplicity max	1
Value	Valeur initiale
Access mode	Read/Write
Type constraint	
Abstract	Attribut static

Property	Value
Name	operation
Visibility	Public
Abstract	Opération static
Class	
Cannot be specialized	
Passing	Out
Redefined operation	<null>

Create diagram...
Create element > Parameter
Return Parameter

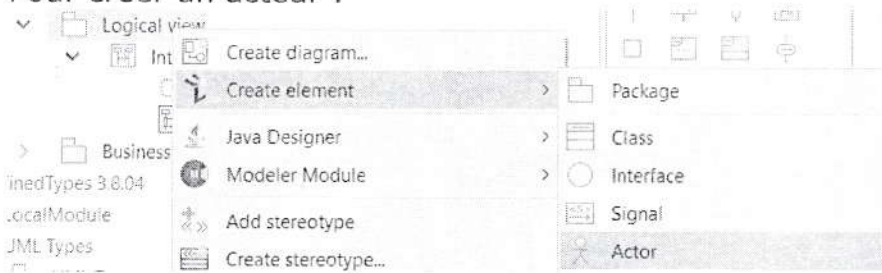
Pour créer une qualification, clic droit sur le lien puis :

Create element > Qualifier

V. Diagramme de séquence

Pour créer le diagramme, clic droit sur le paquetage puis : create diagram...

Pour créer un acteur :

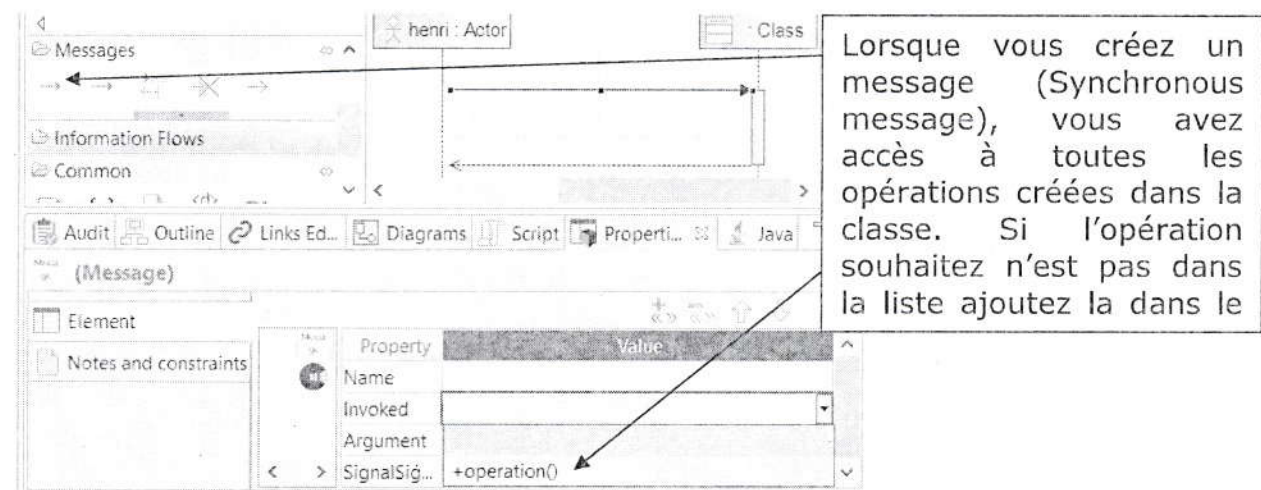


Puis faire glisser l'acteur de l'arbre (onglet Model) vers le diagramme de séquence :



Pour renommer l'acteur le sélectionner puis cliquer 1 fois dessus.

Pour créer un objet faire de même :



Utilisation Modelio

henri : Actor : Class

Messages
Information Flows
Common

Audit Outline Links Ed... Diagrams Script Properti...

(Message)

Element
Notes and constraints

Property	Value
Name	
Invoked	
Argument	
SignalSignature	+operation()

Pour créer un appel à une méthode de sa classe (Inner Synchronous Message).

Donner le nom du paramètre de sortie.

Sélectionner l'opération dans la liste déroulante (Invoked).

Nodes

InteractionUse

Opt
Break
Par
Strict
Loop

Ajoute un nouvel espace par exemple à un alt.

Sélectionner les crochets pour pouvoir écrire la condition de garde.

Property Value
Guard

Pour ajouter une partie dans un alt, un par ...

Nodes

Interaction Operand - Create an Interaction Operand