

Sujet du projet de Génie Logiciel

LAB3416



ESIEA 2016-2017

La chasse aux œufs

Sommaire

Objectifs	3
Règles du jeu	4
Jardin	4
Enfants.....	6
Travail à faire	8
Consignes	9
GitHub	10
Lectures conseillées	12

Objectifs

Le sujet qui vous est proposé cette année n'est pas une fin en soi. C'est un prétexte pour mettre en application les compétences que vous avez acquises à l'école, notamment dans les cours de Programmation Orientée Objet, de Java et de Génie Logiciel.

Par équipe de six à huit membres, vous devez programmer un logiciel et rédiger un rapport. Le logiciel et le rapport compteront chacun pour la moitié des points.

Il n'y a aucune fonctionnalité d'imposée dans le logiciel. Il est toutefois conseillé de mettre en application, au minimum, ce que vous avez déjà réalisé durant les TD. Vous êtes libres de programmer tout ce qui vous passe par la tête. Essayez d'inventer des choses originales.

Le rapport, quant à lui, doit faire apparaître votre démarche de travail, les éléments importants du logiciel, des captures d'écran, les outils utilisés, votre méthodologie, etc. Idéalement il devrait suffire pour que le correcteur n'ait pas besoin de lancer le logiciel pour comprendre son fonctionnement¹.

Pour rappel, voici les éléments essentiels qu'on a vus en cours et en TD :

- UML ;
- Java ;
- Tests (TDD, 3T) ;
- Design pattern ;
- DAO (JDBC / CSV / ...) ;
- Swing (table model) ;
- Story board ;
- Maven ;
- Subversion ;
- Scrum ;

¹ Dans tous les cas, le code source du logiciel sera analysé. À vous de mettre en avant ce qui vous rapportera des bons points.

Règles du jeu

Le sujet de cette année parle d'œufs en chocolat cachés dans le jardin. Les informations sur les œufs, les enfants qui cherchent les œufs et le jardin sont indiquées dans un fichier texte.



Jardin

Les jardins sont rectangulaires et découpés en case rectangulaires de même taille, formant ainsi un quadrillage. Par convention, les lignes sont numérotées de haut en bas et les colonnes de gauche à droite. Il peut y avoir des grands comme des petits jardins.

Voici un exemple de représentation pour un jardin de six colonnes et cinq lignes. À titre d'illustration, j'ai colorié en vert les cases couvertes de pelouse.

	1	2	3	4	5	6
1						
2						
3						
4						
5						

Dans le fichier texte, ce jardin 6x5 sera indiqué comme suit :

```
# jardin(J) largeur hauteur
```

```
J 6 5
```

Les cases peuvent contenir des œufs en chocolat. Certaines cases sont vides. Certaines cases contiennent plusieurs œufs.

Voici un exemple de représentation pour un jardin qui aurait un œuf dans la case 4-2 et trois œufs dans la case 1-4. À titre d'illustration, j'ai colorié les cases possédant des œufs en orange et j'ai indiqué le nombre d'œufs en gras dans la case.

	1	2	3	4	5	6
1						
2				1		
3						
4	3					
5						

Dans le fichier texte, les œufs seront indiqués comme suit :

```
# chocolat(C) position(colonne-ligne) nombre  
C 4-2 1  
C 1-4 3
```

Le jardin est chaotique, il contient aussi des rochers.

Voici un exemple de représentation pour un jardin qui aurait un rocher dans la case 5-3. À titre d'illustration, j'ai colorié les cases possédant des rochers en noir.

	1	2	3	4	5	6
1						
2						
3					X	
4						
5						

Dans le fichier texte, les rochers seront indiqués comme suit :

```
# rocher(R) position(colonne-ligne)
```

```
R 5-3
```

Voici une représentation avec l'ensemble des éléments : le jardin, les œufs en chocolat et les rochers.

	1	2	3	4	5	6
1						
2				1		
3					X	
4	3					
5						

Dans le fichier texte, l'ensemble du jardin sera donc indiqué comme suit :

```
J 6 5
```

```
C 4-2 1
```

```
C 1-4 3
```

```
R 5-3
```

Enfants

Les enfants cherchent les œufs en chocolat dans le jardin. Il peut y avoir plusieurs enfants en même temps dans le jardin.

Un enfant ne peut se déplacer qu'en avançant. Il peut néanmoins tourner sur lui-même, à gauche ou à droite, de 90°.

Les déplacements sont codifiés. Par exemple, la séquence « AADADAGA » signifie que l'enfant avance, puis avance, puis fait une rotation à droite, puis avance, puis fait une rotation à droite, puis avance, puis fait une rotation à gauche, puis avance. Chaque mouvement prend une seconde, y compris les rotations. Ainsi, un tour complet (360°) prend quatre secondes.

Au début du jeu, les enfants ont également une position et une orientation initiales. L'orientation correspond aux points cardinaux.

Voici un exemple de représentation, pour un enfant initialement dans la case en haut à gauche, regardant vers l'est et exécutant la séquence « AADADAGA ». À titre d'illustration, j'ai colorié en bleu le chemin de l'enfant, en indiquant par une flèche la dernière orientation connue.

	1	2	3	4	5	6
1	→	→	↓			
2		↓	←			
3		↓				
4						
5						

Dans le fichier texte, les enfants seront indiqués comme suit :

```
# enfant(E) position initiale(colonne-ligne) orientation initiale(N|E|S|W)
deplacements(A|G|D) nom
E 1-1 E AADADAGA Julien
```

Note : Pour des raisons de flexibilité, les enfants et leurs déplacements sont indiqués dans un fichier texte différent de celui du jardin. Il est ainsi possible de mixer un lot d'enfants avec un jardin spécifique.

Lorsqu'un enfant arrive sur une case contenant des œufs en chocolat, il en prélève un et le range dans son panier. Cela lui prend une seconde. Il passe donc une seconde pour aller sur la case et une autre seconde pour prendre et ranger l'œuf en chocolat, soit deux secondes.

Un enfant ne peut pas aller sur une case où se trouve un rocher. Les instructions d'avancer (A) vers une case contenant un rocher sont ignorées. En toute logique, la séquence reprend donc à l'instruction de rotation (G|D) suivante.

Une case ne peut contenir qu'un seul enfant à la fois. Un enfant désirant aller vers une case déjà occupée par un autre enfant doit se mettre en pause tant que la case n'est pas vide. Chaque seconde, un enfant est donc en mouvement vers l'avant, en rotation ou en pause.

Tous les enfants commencent leurs séquences respectives en même temps. Lorsque la séquence d'un enfant est terminée, celui-ci se met en pause et reste sur la case où il est définitivement.

Travail à faire

Au minimum, vous devez donc lire un fichier de jardin et un fichier d'enfant. Vous pouvez commencer avec les exemples proposés dans le sujet. Vous devrez ensuite créer vos propres jeux de données, avec des jardins de tailles et d'organisations différentes. Vous pouvez aussi faire varier le nombre d'enfants et leurs séquences de déplacement.

À la fin du jeu, vous devez indiquer les nombres d'œufs en chocolat rangés dans les paniers de chaque enfant, sous forme de tableau. Vous pouvez également indiquer les informations lues dans les fichiers.

Et puis vous devez imaginer d'autres fonctionnalités sympas. Voici quelques propositions :

- faites des graphiques ;
- développez un éditeur de jardins et d'enfant ;
- faites des animations montrant les enfants en déplacement, en temps réel, dans le jardin. Vous avez le droit de reprendre les mêmes codes couleur que dans l'énoncé. Pour cela, vous pouvez par exemple utiliser une JTable que vous mettrez à jour toutes les secondes. Vous pouvez aussi dessiner un graphique AWT ;
- utilisez des durées d'action différentes. Par exemple une seconde pour une rotation, deux secondes pour un déplacement et cinq secondes pour ranger un œuf dans le panier ;
- les enfants peuvent se voler les œufs dans leurs paniers ;
- Ajoutez des œufs magiques aux effets spéciaux. Il peut y en avoir plusieurs couleurs ;
- Ajoutez des enfants autonomes (sans séquence prédéfinie). La classe Java « Thread » devrait vous aider ;
- Accolez plusieurs jardins, avec des trous dans les barrières de séparations, pour permettre aux enfants de passer d'un jardin à l'autre ;
- Associez-vous avec un autre groupe pour faire des concours de ramassage ;
- Etc.

Bien entendu, le projet doit être développé en Java. N'oubliez pas d'utiliser Maven, de faire des tests, de vérifier la qualité de votre code, etc. Dans le rapport, n'oubliez pas de mettre des captures d'écran, des extraits de code ou de fichier, des maquettes, des « story boards », des diagrammes UML, des CR de réunion, etc. Vous avez le droit d'écrire un gros rapport.

Consignes

Le non-respect des consignes suivantes entrainera des pertes de points :

- le code de votre logiciel doit être déposé sur GitHub, dans un projet GitHub créé pour l'occasion ;
- le projet doit être envoyé par email à l'adresse « thierryler@gmail.com », avant la date indiquée en cours, en mettant en copie tous les membres de votre groupe ;
- le titre de l'email doit être « Projet GLPOO ESIEA 2016-2017 groupe Dupont », où vous devez remplacer « Dupont » par le nom du responsable de votre groupe ;
- l'email doit contenir un fichier ZIP nommé « `projet_GLPOO_ESIEA_2016-2017_groupe-Dupont.zip` ». Cette archive doit contenir :
 - votre rapport au format PDF, nommé « `rapport.pdf` »,
 - un fichier texte, nommé « `notice.txt` », contenant la liste des membres de votre groupe (format NOM Prénom classe) et l'adresse de votre projet sur GitHub,
- si vous avez des questions, elles doivent être posées par email à l'adresse « thierryler@gmail.com » ;
- merci de passer le correcteur orthographique sur votre rapport ;
- relisez ou faites relire votre rapport, pour éviter les phrases qui ne veulent rien dire ;
- rédigez votre rapport dans un style simple. On ne vous demande pas de faire du Hugo ou du Shakespeare ;
- éviter de me faire un copié-collé du cours, pour m'expliquer ce qu'est UML ou Scrum par exemple. C'est la façon dont vous vous en êtes servi qui est intéressante. Dites-vous que le professeur sait déjà ce qu'est UML ;
- n'attendez pas le dernier moment pour envoyer votre projet... La date de remise du projet vous a été donnée en cours.

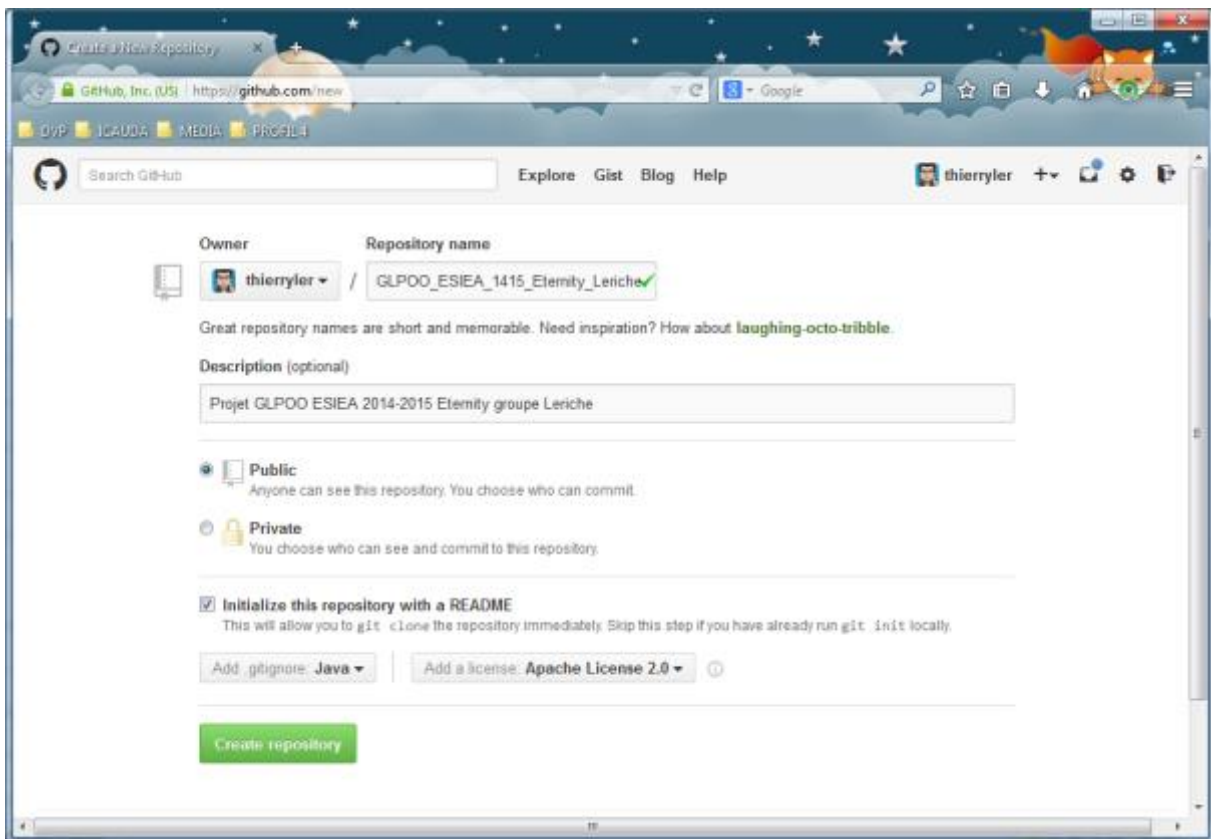
GitHub

GitHub (<https://github.com/>) est un repo Git devenu incontournable depuis quelques années. Je recommande à tous les membres de votre groupe de créer un compte (c'est gratuit).

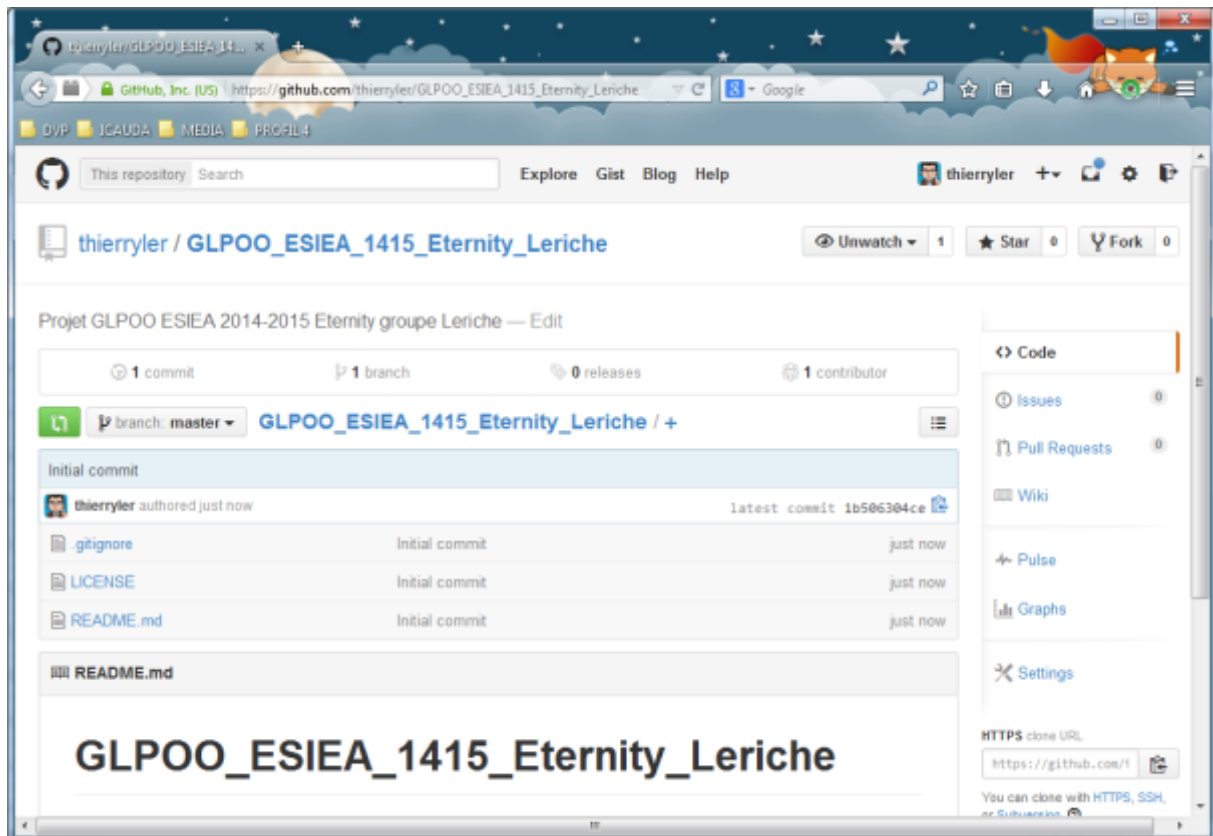
Vous devez travailler en groupe en utilisant GitHub comme gestionnaire de source. C'est l'URL de votre projet GitHub que vous devez indiquer dans la notice que vous enverrez par email.

Pour créer un projet, connectez-vous puis allez à l'adresse <https://github.com/new>

Indiquez « GLPOO_ESIEA_1617_Groupe_Dupont² » comme nom de projet. Je vous conseille d'initialiser un fichier « README » et d'ajouter le fichier « .gitignore » pour Java. Pour la license, prenez « Apache 2 »...



² Vous pouvez choisir un autre nom pour le projet sur Github, si vous le préférez.



Dans votre console Git Bash, vous pouvez faire un clone comme indiqué dans la capture d'écran suivante :

```
MINGW32/c/mesprojets/GLPOO_ESIEA_1415_Eternity_Leriché

Thierry@ONYX13 /c
$ cd mesprojets/

Thierry@ONYX13 /c/mesprojets
$ git clone https://github.com/thierryler/GLPOO_ESIEA_1415_Eternity_Leriché.git
Cloning into 'GLPOO_ESIEA_1415_Eternity_Leriché'...
remote: Counting objects: 5, done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (5/5), done.
Checking connectivity... done.

Thierry@ONYX13 /c/mesprojets
$ ls
GLPOO_ESIEA_1415_Eternity_Leriché

Thierry@ONYX13 /c/mesprojets
$ cd GLPOO_ESIEA_1415_Eternity_Leriché/

Thierry@ONYX13 /c/mesprojets/GLPOO_ESIEA_1415_Eternity_Leriché (master)
$ ls
LICENSE  README.md

Thierry@ONYX13 /c/mesprojets/GLPOO_ESIEA_1415_Eternity_Leriché (master)
$
```

Lectures conseillées

- Tutoriel de Michel Douez, nommé « Programmez un Snake avec Java2D » :
<http://michel-douez.developpez.com/articles/java/snake/>
- Tutoriel de Thierry Leriche, nommé « Charger des données depuis un fichier CSV simple en 5 minutes » :
<http://thierry-leriche-dessirier.developpez.com/tutoriels/java/charger-donnees-fichier-csv-5-min/>
- Tutoriels de Baptiste Wicht, à propos de Swing :
<http://baptiste-wicht.developpez.com/tutoriels/swing/>
- Tutoriel de Nicolas Zozol, nommé « JTables - Un autre regard » :
<http://nicolas-zozol.developpez.com/tutoriel/java/jtable/>
- Tutoriel de Thierry Leriche, nommé « Afficher un tableau avec un Table Model Swing en 5 minutes » :
<http://thierry-leriche-dessirier.developpez.com/tutoriels/java/afficher-tableau-avec-tablemodel-5-min/>
- Mémento Scrum à destination de l'équipe v1.1 :
<http://thierry-leriche-dessirier.developpez.com/tutoriels/general/memento-scrum-destination-equipe/>
- Mémento des raccourcis d'Eclipse :
<http://thierry-leriche-dessirier.developpez.com/tutoriels/eclipse/raccourcis/>
- Mémento des mots-clés de Java :
<http://thierry-leriche-dessirier.developpez.com/tutoriels/java/mots-cles-java/>