

Projet: documents d'analyse

F. Barthélemy

1 Étape d'analyse du projet

Nous vous proposons de réaliser une réflexion préalable à la réalisation du projet. Cette analyse décrira la première version du projet que vous réaliserez. Cette analyse ne vous engage pas : si vous voulez revenir ultérieurement sur certains choix ou faire évoluer votre projet avec de nouvelles caractéristiques, vous pourrez le faire.

Cette étape a pour but de vous aider à clarifier vos idées et de vous assurer que vous ne partez pas dans une mauvaise direction. Votre analyse sera validée par un enseignant, ce qui vous permettra de continuer sereinement votre travail.

Il est fondamental de mettre en oeuvre dans le projet l'approche objet présentée en cours. Vos données doivent être encapsulées dans des objets. Vous aurez donc plusieurs classes à écrire. Au début du projet, pour l'analyse, vous pouvez partir sur une représentation simple. À la fin du projet, on appréciera l'utilisation d'héritage et/ou d'interfaces.

2 Scènes et plans

On appellera *scène* une animation comportant un fond et différentes images, les unes immobiles et les autres en mouvement. Votre programme, dans sa méthode `main`, construira et affichera une scène donnée. Mais votre programme devra être conçu de façon à permettre la réalisation facile d'autres scènes et celle que vous montrerez ne sera qu'une illustration de cette aptitude de votre programme.

La méthode `main` commencera par construire tous les objets nécessaires pour représenter une scène, puis elle comportera une boucle avec à chaque tour de cette boucle le calcul d'une image de la scène, l'affichage de cette image et une temporisation utilisant `Thread.sleep`.

La notion de `plan` fait référence à une notion de profondeur. On utilise ce terme comme dans la langue courante, on parle de premier plan pour désigner ce qui est proche et arrière-plan pour désigner ce qui est lointain. Cette notion de plan est utile pour définir ce qui est visible quand plusieurs images sont superposées. En cas de superposition, l'image la plus proche cache celle qui est plus lointaine. Cette notion de plan sera utile pour construire vos scènes. Nous vous suggérons donc de l'utiliser dans le projet.

3 Quelques questions pour guider votre analyse

1. Quel scénario allez-vous choisir pour la scène construite par la méthode `main` ? Allez-vous avoir des changements de direction des objets animés ? Allez-vous mettre plusieurs images dans le même plan ? Si oui, que se passera-t-il en cas de collision de deux images ?
2. Quel rôle va avoir la notion de plan dans le processus de calcul de l'image à afficher ?
3. Une image en mouvement et une image fixe seront-elles représentées de la même façon dans votre programme ?
4. Le fond de la scène est-il une image fixe comme une autre ? Sinon, quelle est la différence ?
5. Les classes fournies imposent deux représentations différentes des images : une avec le type `int[][][]` (classe `ImageReader`) l'autre avec le type `Pixel[][]` (classe `Afficheur`). Il y aura donc de toute façon des conversions à faire entre représentations d'images. Votre programme va-t-il utiliser une de ces deux représentations ou va-t-il en utiliser une troisième ?
6. Comment sera représentée une scène dans votre programme ?
7. Comment sera représentée la notion de plan dans la scène ?
8. Votre `main` comportera une boucle avec à chaque tour de boucle le calcul d'une nouvelle image et une temporisation utilisant `Thread.sleep`. A un tour de boucle donné, comment sera calculée la position d'une image sur le fond ?

4 Les classes et attributs

Créez un squelette du projet (version 1) avec ses différentes classes, éventuellement, les différents packages. Pour chaque classe, vous donnerez les attributs ou variables d'instances permettant de représenter les données du programme.

Vous donnerez également les entêtes des constructeurs et des méthodes les plus importantes. (Rappel : l'entête est constitué du type du résultat, du nom de la méthode et des types et noms des paramètres).

Ce squelette du projet est à rendre via un formulaire de l'ENF.

5 Méthode main

Donnez l'esquisse de la méthode `main` qui construit une scène puis l'affiche. Par esquisse, nous entendons un code approximatif qui fixe les grandes structures (boucles, if) de ce code, les principaux constructeurs utilisés et les principales méthodes appelées.

6 Documents à remettre

Vous devez remettre via l'ENF deux fichiers :

1. La réponse aux questions via un fichier libre office (fichier fourni à compléter).
2. L'archive eclipse comportant les classes, attributs et esquisse de la méthode main.