LOG2410 – TP4 Hiver 2017

Conception à base de patrons I

1 - Objectifs

Ce laboratoire permettra aux étudiants de se familiariser avec l'implémentation des patrons de conception « Composite », « Proxy » et « Template Method ». Cette implémentation est effectuée à l'aide du logiciel Visual Studio et le langage C++ sera utilisé tout au long du processus de développement. Un cadriciel est fourni, qui doit être complété afin d'obtenir le résultat final souhaité.

2 - Patron Composite et Proxy (50 points)

L'application commande du système de PolyScino est constitué d'un ou plusieurs commandes simples et composées. Les commandes permettent d'orienter, tourner et de prendre des photos. Les commandes sont toujours appliquées au robot même lorsqu'il s'agit de prendre des photos par la caméra. Lors de la prise de photo, le robot doit déléguer certaines commandes à la caméra.

Implémentation

On vous demande de compléter les fichiers suivants :

- `CmdBoucleFor.cpp`
- `Robot.cpp`

afin que les tests programmés dans la méthode `Test_TP4::executeCompositeTest()` s'exécutent avec succès.

`Test_TP4::executeProxyTest()` s'exécutent avec succès.

Questions à répondre

1) Identifiez les points suivants :
   a) L'intention du patron Composite et Proxy.
   b) La structure des classes réelles qui participent aux patrons ainsi que leurs rôles. Faite un diagramme de classes avec Enterprise Architect pour l'instance du patron composite. Ajouter des notes en UML pour indiquer les rôles, et exportez le tout en pdf).
   c) La structure des classes réelles qui participent au patron ainsi que leurs rôles. Faite un diagramme de classes avec Enterprise Architect pour l'instance du patron proxy. Ajouter des notes en UML pour indiquer les rôles, et exportez le tout en pdf.
2) Dans l'implémentation actuelle du système PolyScino, quel objet ou classe est responsable de la création de l'arbre des composantes.

3 - Patron Template Method (50 points)

Le système PolyScino offre des fonctionnalités de plusieurs diagnostiques automatiques qui peuvent être lancées à partir de la classe de base *`CameraAbs`*. Ces fonctionnalités doivent tenir compte de la mémoire de stockage des photos afin de permettre la sauvegarde des photos et des séquences de photos, le niveau de la batterie ainsi que l'activation et la désactivation du flash.

Implémentation

On vous demande de compléter le fichier suivant :

- `CameraAbs.cpp`

afin que les tests programmés dans la méthode `Test_TP4::executeTemplateMethodTest()` s'exécutent avec succès.

Questions à répondre

1) Identifiez les points suivants :
   a) L'intention du patron Template Method.

b) La structure des classes réelles qui participent au patron ainsi que leurs rôles (faite un diagramme de classes avec Enterprise Architect, ajouter des notes en UML pour indiquer les rôles, et exportez le tout en pdf).

2) Selon vous, la classe abstraite ***CmdAbs*** est-elle une instanciation du patron de conception Commande ? Justifiez votre réponse.

4 – À remettre

1) Une archive `LOG2410_TP4_matricule1_matricule2.zip` qui contient les éléments suivants :
   a) Le fichier `ReponsesAuxQuestions.pdf` avec la réponse aux questions 2.1a), 2.2), 3.1a) et 3.2)
   b) Le fichier `DiagrammeDeClasses_Composite.pdf` pour le diagramme de classes des deux patrons composite de la question 2.1b)
   c) Le fichier `DiagrammeDeClasses_Proxy.pdf` pour le diagramme de classes des deux patrons composite de la question 2.1c)
   d) Le fichier `DiagrammeDeClasses_TemplateMethode.pdf` pour le diagramme de classe de la question 3.1b).
   e) Les trois fichiers C++ que vous avez modifiés, c'est-à-dire,
      `CmdBoucleFor.cpp,`
      `Robot.cpp,`
      `Et CameraAbs.cpp`.
      Vous ne pouvez pas modifier les autres fichiers `.h` et `.cpp`. Le correcteur va insérer vos trois fichiers dans le code, et ça doit compiler et s'exécuter.