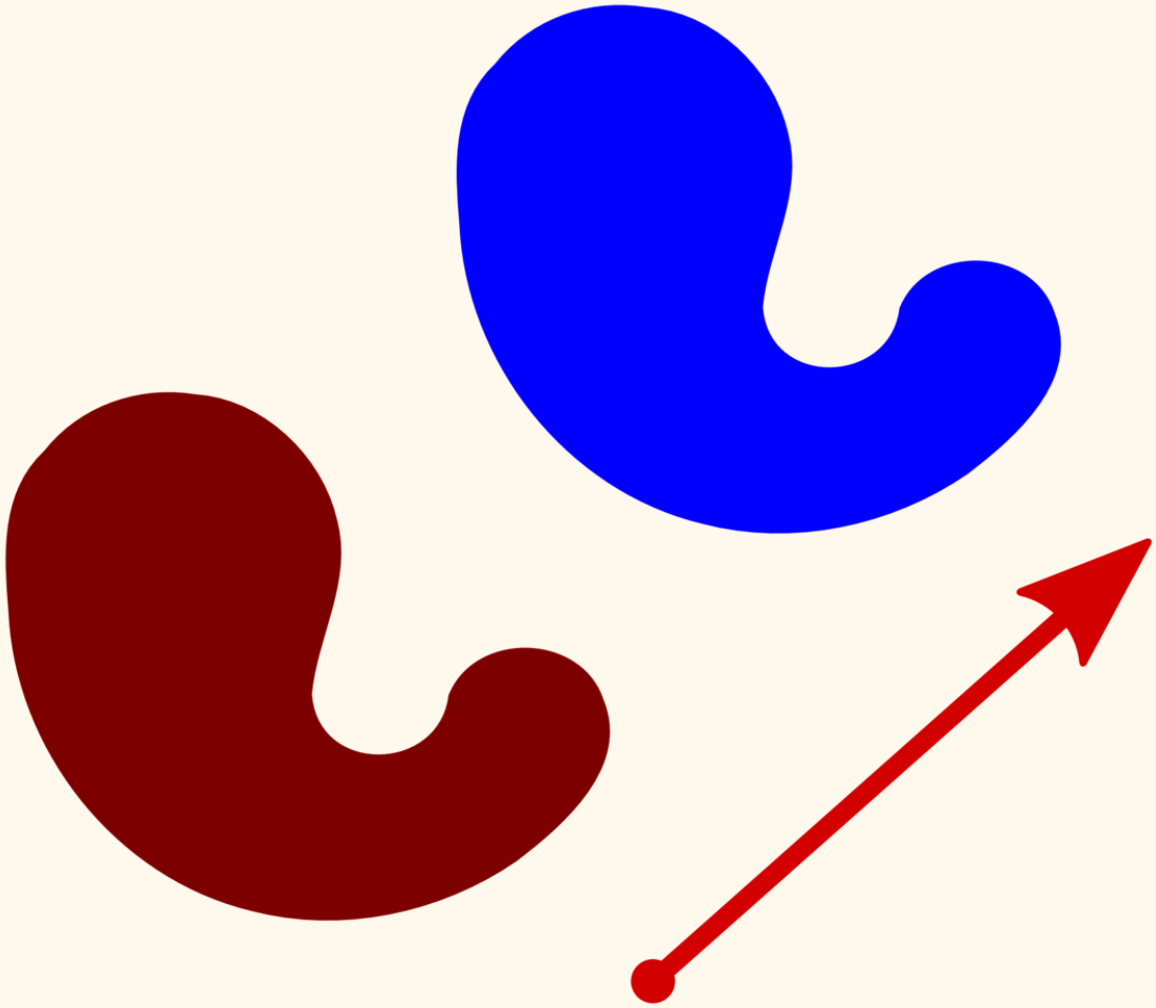
EPITECH.
L'ECOLE DE L'EXPERTISE INFORMATIQUE



COMMENT IMPLÉMENTER DE NOUVELLES TRANSFORMATIONS

Projet RayTracer : Marius Pain, Landry Gigant, Thomas Boué & Aubane Nourry

Introduction

L'intégration de nouvelles transformations pour le projet RayTracer d'Epitech requiert une approche méthodique et structurée, étroitement alignée avec une architecture spécifique afin de garantir sa compatibilité avec le corps du projet. Dans ce document, nous explorerons les étapes essentielles pour implémenter de nouvelles transformations dans le cadre du projet RayTracer. Nous mettrons en lumière l'importance de respecter les conventions établies dans le but de garantir la fonctionnalité et la compatibilité de ces objets avec notre projet, tout en assurant une expérience de développement harmonieuse.

Créer un nouvelle transformation

Une fois la transformation sélectionnée, il est temps de procéder à son intégration dans l'architecture existante du projet RayTracer.

1. Les Prérequis

Vous devrez dans un premier temps :

- Créer une interface pour votre transformation.
- Mettre à jour l’analyse syntaxique pour pouvoir appliquer votre transformation.
- Optionnel: Implémenter cette transformation à des objets primitifs.

2. Votre interface principale

Afin d'implémenter votre interface principale, vous devrez simplement définir deux méthodes minimum.

a. Destructeur de classe

En tant qu'interface, il est primordial de définir un destructeur de classe qui deviendra obligatoire pour implémenter des classes enfants de cette interface. Le rôle du destructeur est de libérer l'espace mémoire des potentiels pointeurs contenus dans les enfants de cette interface. Vous devrez donc faire en sorte que tous les enfants de cette interface implémentent ce destructeur.

b. Méthodes signature

Les méthodes signatures seront les méthodes qui vont effectuer la translation choisie. Vous devrez également faire en sorte que les enfants de cette interface implémentent ces méthodes car elles représentent l'essence même de l'interface.

Par exemple, la transformation nommée "scale" qui permet d'agrandir ou de diminuer la taille d'un objet, est défini de cette manière:

"
```
class ICanScale {

      public:

            ~ICanScale() = default;


            virtual void scale(double multiplier) = 0;

};
```
"

Ici, l'interface "`ICanScale`" a deux méthodes:

- Le destructeur qui permet de libérer les éventuels pointeurs créés par les enfants de cette interface.
- La méthode "`scale(double multiplier)`"

Ici, l'interface "`ICanScale`" n'a qu'une méthode signature car elle n'a pas besoin de plus. Or, si plusieurs paramètres sont possibles, vous pouvez ajouter des méthodes avec des paramètres différents, comme la translation par exemple.

3. Mettre à jour l'analyse syntaxique

Une fois que votre interface de transformation est créée, vous arrivez à la dernière étape. Il ne reste plus qu'à modifier la partie analyse syntaxique du programme pour pouvoir renseigner votre nouvelle transformations dans les fichiers de configuration. *(Comment créer un fichier de configuration)*. Pour ce faire, rendez-vous dans les fichiers Scene.cpp et Scene.hpp.

Tout d'abord, vous devez ajouter une ligne dans le constructeur de la classe. Cette ligne permet de définir que ce type de transformation existe. Elle permet également de spécifier la fonction que le programme va devoir utiliser. Par exemple, si vous voulez ajouter la transformation qui tord un objet, vous devrez ajouter cette ligne:

```
“{“shear”, [this](libconfig::Setting &transformation,
std::shared_ptr<IPrimitive> primitive),
{applyShear(transformation,primitive);}”
```

Ensuite, vous devrez implémenter une méthode qui permet d'appliquer votre transformation. Par exemple, pour créer la transformation qui tord un objet, vous devrez réaliser une méthode qui avec ce prototype:

“`void RayTracer::Scene::applyShear(libconfig::Setting &transformation, std::shared_ptr<IPrimitive> primitive)`“

Cette méthode va permettre d'appliquer la transformation qui tord un objet à la liste des objets primitifs (“`_primitives`”) cités dans la liste des transformations. Pour ce faire, vous devez:

- Vérifier que l'objet en question hérite de l'interface de votre transformation
- Appliquer votre transformation sur l'objet en question en appelant la méthode adaptée.

4. Optionnel: Implémenter votre transformation à des objets

Une fois que vous avez terminé de créer votre transformation, l'idéal serait

de pouvoir implémenter votre transformation à tous les objets pour

lesquels la transformation est possible.

Pour cela, consultez *la documentation pour réaliser un objet primitif*.

5. Conclusion

Vous avez désormais toutes les clés en main pour implémenter une nouvelle transformation pour notre RayTracer, n'hésitez pas à faire une pull-request pour ajouter votre transformation si vous souhaitez contribuer au projet. Nous vous remercions par avance de votre contribution.