

Partie 3 - Diagramme de classes et diagramme d'interaction

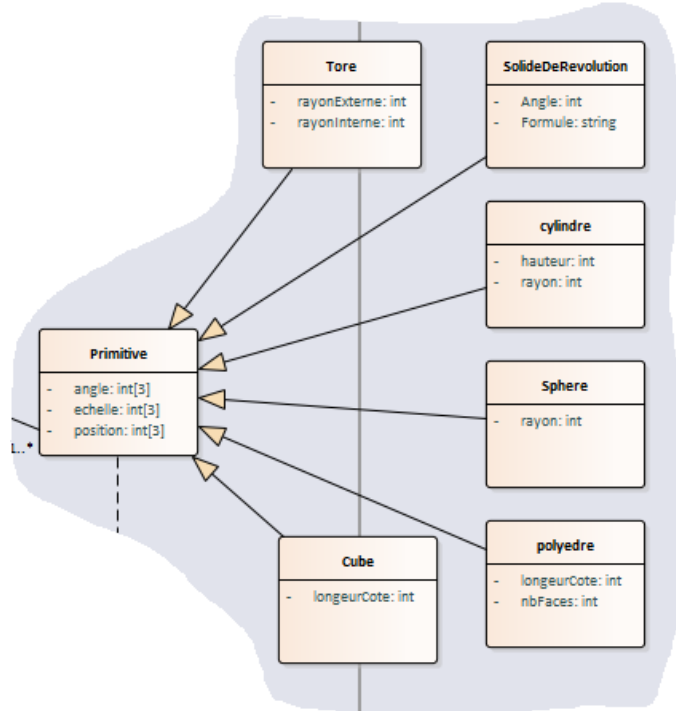
Cas d'utilisation:	CU-8.0-Télécharger Icône "Téléversement d'icônes stockées localement vers le serveur"
Système:	PolyIcône3D
Précondition(s):	Le designer doit avoir un compte. Une connexion à distance doit être disponible.
Postconditions(s):	Le fichier est sauvegardé sur le serveur distant. Dans le cas contraire, une erreur est affichée.
Acteur(s):	Designer, Serveur d'hébergement
Parties prenantes:	Designer: Le designer désire téléverser son nouvel icônes vers le serveur distant.
Type:	Primaire
Scénario principal:	Le designer se connecte sur son compte, il ouvre un projet existant, il sélectionne certaines icônes puis il les téléverse vers le serveur d'hébergement.
Scénario secondaire:	Le designer se connecte sur son compte, il ouvre un projet existant, il sélectionne certaines icônes puis il les téléverse vers le serveur d'hébergement. Une erreur survient et le téléversement ne peut être complété. Un message informe le designer de l'erreur.

Question : Expliquez en vos propres mots la/les différence(s) entre le diagramme de concepts et le diagramme de classes.

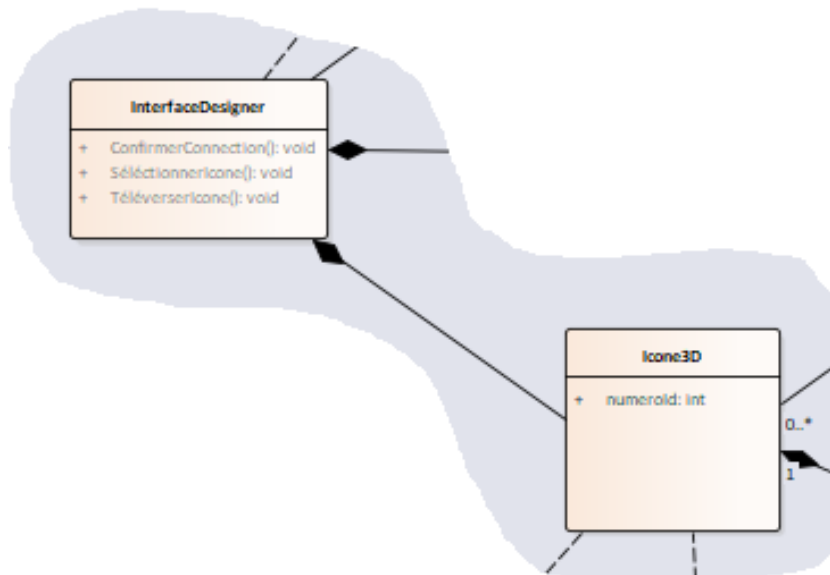
La différence entre un diagramme de concept et un diagramme de classe réside dans le fait qu'un concept est une représentation d'un élément ou d'un regroupement d'éléments concrets du monde réel, alors qu'une classe est une représentation purement logicielle de ces concepts. Étant donné qu'une représentation logicielle peut être abstraite, les relations entre les concepts et leur représentation sous forme de classe peuvent être différentes.

Partie 4 - Assignment de responsabilité et patrons GRASP

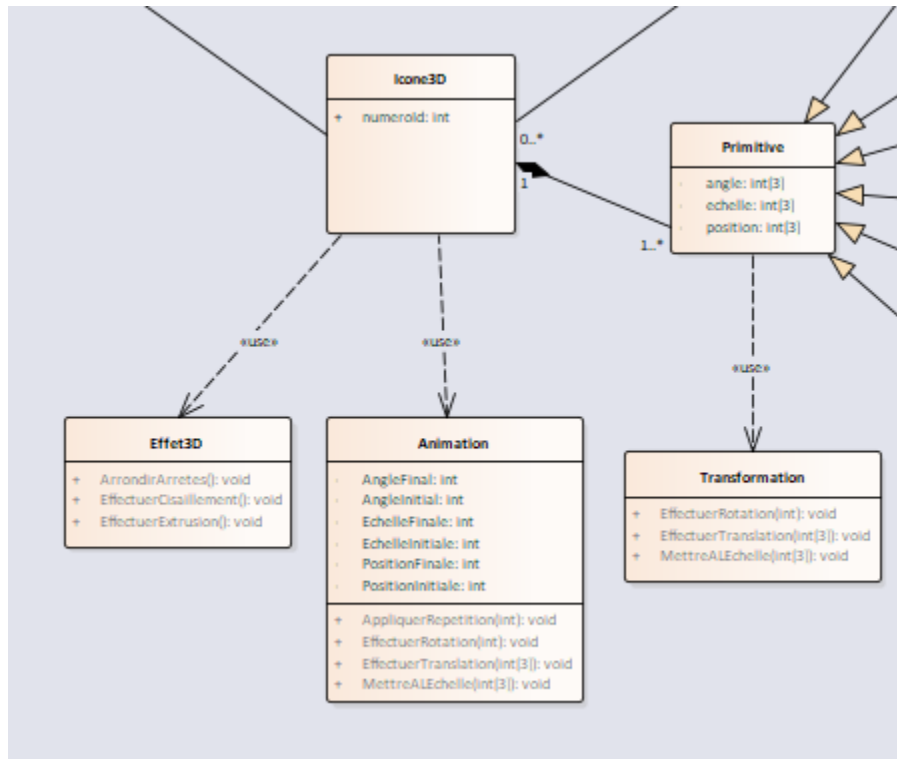
Primitive : Gérer les différentes alternatives quant à la création des primitives. Selon le patron de polymorphisme, la classe Primitive sera une classe abstraite qui implémentera une série de variations disponibles.



InterfaceDesigner : Gérer l’affichage de l’interface utilisateur. Selon le patron contrôleur, c’est cette classe qui devrait gérer l’affichage puisqu’elle centralise les éléments à afficher en un seul endroit et s’occupe de la gestion des événements qu’il déclenche.



Icône3D : Gérer l'état du modèle. Selon le patron expert, c'est la classe Icône3D qui contiendra toutes les informations de notre icône, c'est-à-dire le modèle.



Icône3D : Créer les primitives qui le composent. Selon le patron créateur, c'est la classe Icône3D qui devrait avoir la responsabilité de créer les primitives qui composera ce dernier.

