UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique et de génie informatique

Rapport d'APP

Modélisation et programmation orientée objet
GEN241

Présenté à
Domingo Palao Muñoz

Présenté par
Benjamin Chausse – chab1704

Sherbrooke – 16 janvier 2023

Table des matières

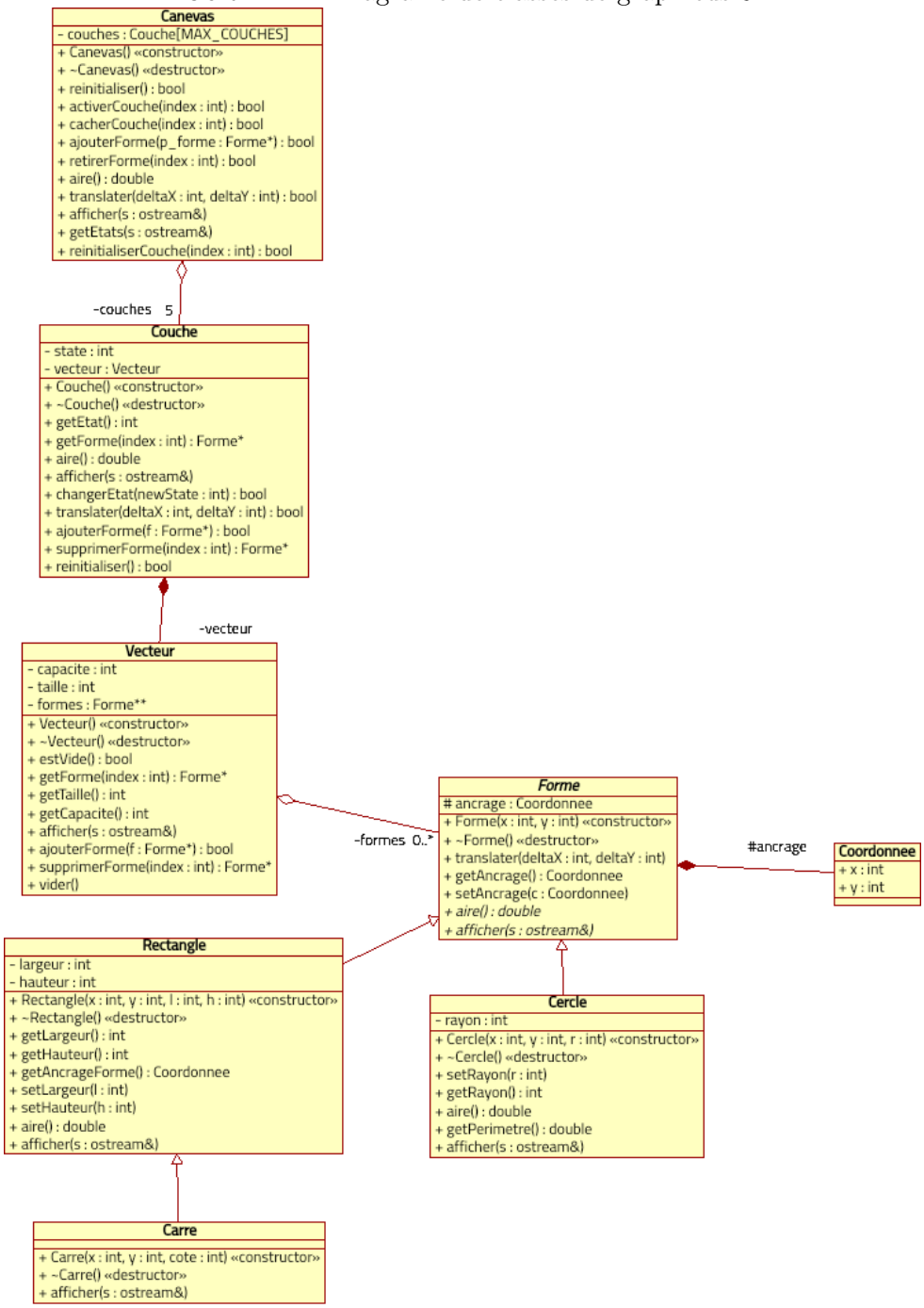
Table des figures

Liste des tableaux

1 Diagrammes UML

La figure 1-1 montre les diverses relation entre les classes de graphicus-02. Elle permet d'observer la hiérarchie des classes ainsi que les facteurs de dépendance entre les classes.

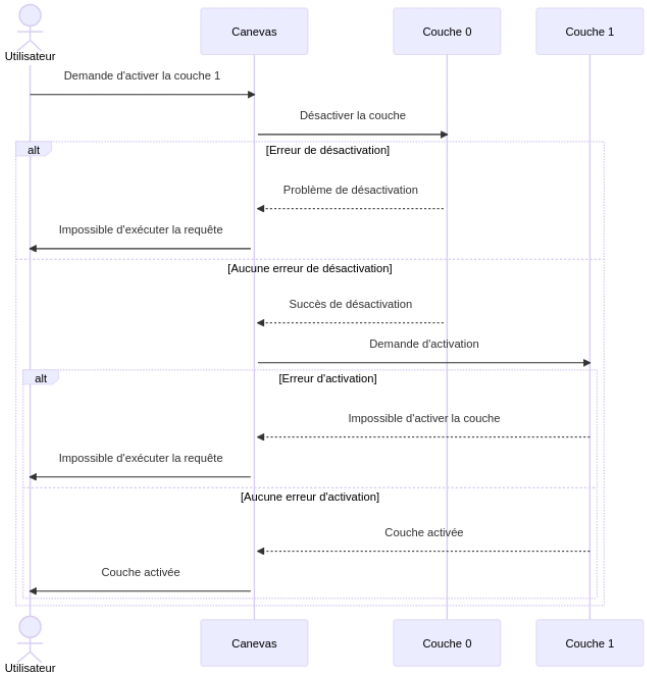
FIGURE 1-1 – Diagrame de classes de graphicus-02



La figure 1-2 montre comment l'application procéderait lorsqu'un utilisateur voudrait activer

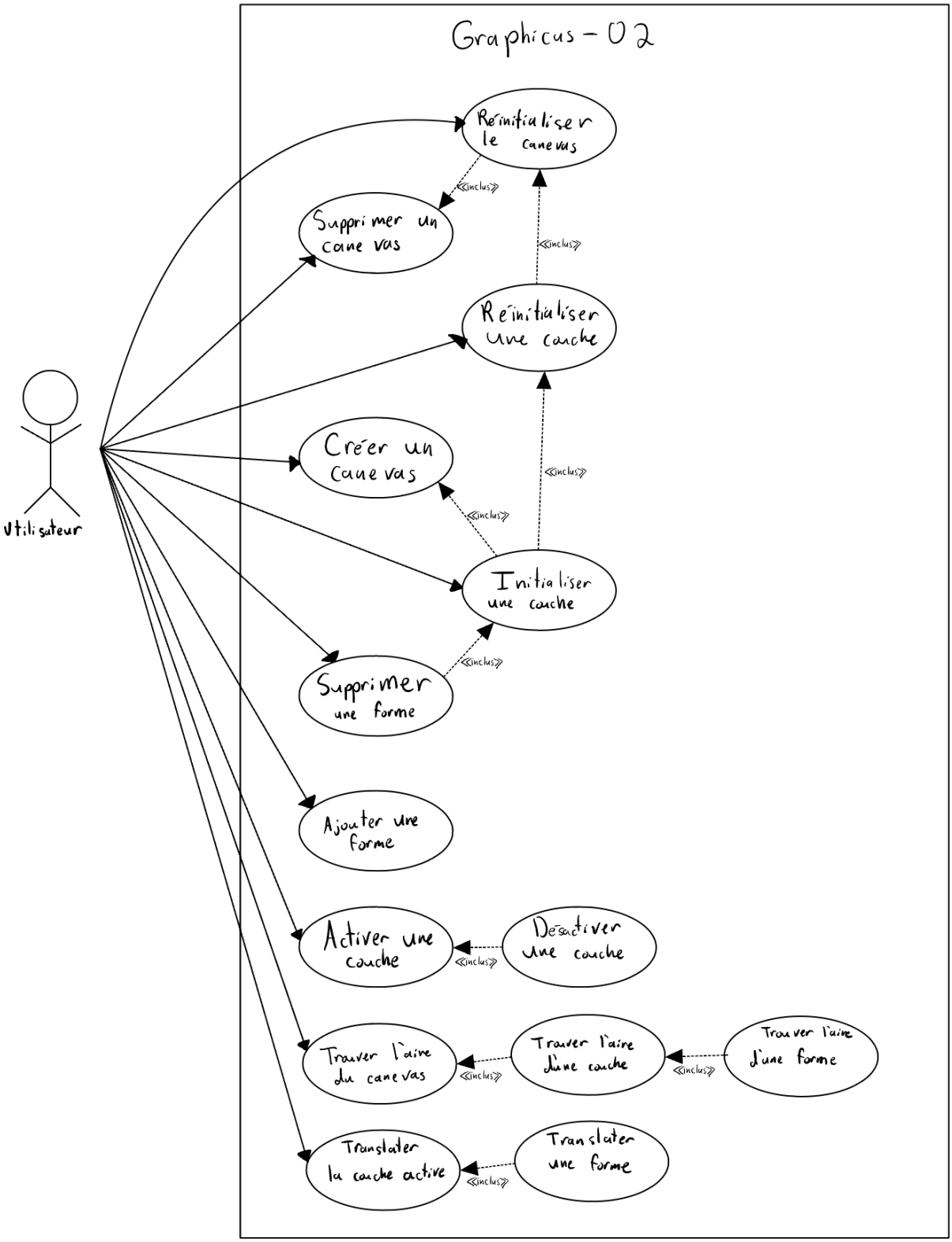
la couche 1 du canevas alors que la couche 0 est présentement active. Les autres couches ont été omises pour alléger le diagramme.

FIGURE 1-2 – Diagrame de séquence d'activation de couche



La figure 1-3 montre les différentes actions que l'utilisateur peut exécuter dans l'application. Nous voyos par la suite comment certaines de ces actions agissent en arrière plan et dépendent d'autres actions (parfois même d'autres actions accessibles à l'utilisateur).

FIGURE 1-3 – Diagramme de cas d'utilisation



2 Pseudo-code

Algorithme 1 Ajout d'un élément au vecteur

```
1: FONCTION VECTEUR : :AJOUTERFORME(f) : Booléen
2:     // f : Forme à ajouter au vecteur
3:     // Vecteur :: taille : nombre d'éléments actuellement dans le vecteur (entier)
4:     // Vecteur :: capacite : capacité maximale du vecteur actuellement (entier)
5:     // Vecteur :: formes : Liste de pointeurs vers les formes du vecteur (*Forme)
6: DÉBUT
7:     // newCapacite : nouvelle capacité du vecteur (entier)
8:     // newFormes : nouvelle liste de pointeurs vers les formes du vecteur (*Forme)
9:     SI f est de valeur nulle ALORS                    ▷ Vérifier si la forme est valide
10:        Retourner Faux
11:    SI Vecteur :: taille == Vecteur :: capacite ALORS    ▷ Si le vecteur est plein
12:        newCapacite := Vecteur :: capacite × 2
13:        SI L'espace disponible en mémoire < newCapacite × taille(Forme) ALORS
14:            Retourner Faux
15:        newFormes := allouer(newCapacite × taille(Forme))
16:        POUR CHAQUE Forme i dans Vecteur :: formes, FAIRE :
17:            newFormes[i] := Vecteur :: formes[i]
18:        FIN POUR
19:        Vecteur :: capacite := newCapacite
20:        libérer(Vecteur :: formes)
21:        Vecteur :: formes := newFormes
22:    Vecteur :: taille := Vecteur :: taille + 1
23:    Vecteur :: formes[Vecteur :: taille] := f
24:    Retourner Vrai
25: FIN
```

3 Plan de test

3.1 Identification

TABLE 3-1 – Informations générales du plan de test

Nom :	Scénario de test du rapport d'APP
But :	Vérifier les fonctionnalitées globales de graphicus-02
Acteur principal :	classe `Test`
Date de création :	2023-01-14
Auteur :	Benjamin Chausse – chab1704
Version :	1.0

3.2 Scénario

3.2.1 Préconditions

— Un canevas vierge vient d'être créé
— Ce cas d'utilisation emploie la méthode `tests_application_cas_02` de la classe `Tests`. La première ayant été utilisée lors de la validation en classe.

3.2.2 Postconditions

— Le déroulement du test se fait sans intervention de l'utilisateur.
— L'affichage doit-être accessible à l'évaluateur :
  — soit par écriture dans un fichier
  — soit par affichage dans un terminal
— Le numéro de chaque étape est identifié avant son exécution.
— La description de chaque étape est affichée avant son exécution.
— Les informations passées à chaque étape sont affichées avant son exécution.

3.2.3 Limitations

— Le test ne vérifie pas la totalité des cas limites (ex : capacité insuffisante en mémoire pour agrandir le vecteur).

3.3 Enchaînement nominal

3.3.1 Étapes 1 à 4

1. Activer la couche d'index 4
2. Ajouter les formes suivantes au canevas :
   — un cercle de centre (1, 2) et de rayon $\frac{1}{\sqrt{\pi}}$
   — un rectangle anchré en (3, 4), de dimensions 3×4
   — un carré anchré en (-1, -1), de côté 2
3. Afficher le canevas
4. Imprimer l'aire du canevas (doit être égale à $1 + 12 + 4$ soit 17)

3.3.2 Étapes 5 à 13

5. Activer la couche d'index 3
6. Ajouter les formes par défaut au canevas. Soit :
   — un cercle de centre (0, 0) et de rayon 1
   — un rectangle anchré en (0, 0), de largeur 1 et de hauteur 1
   — un carré anchré en (0, 0), de côté 1

7. Afficher le canevas
8. Translater la couche active de (1, 1)
9. Afficher le canevas
10. Supprimer la forme de l'index 0 (la première)
11. Activer la couche d'index 4
12. Supprimer la forme de l'index 2 (la dernière)
13. Afficher le canevas

3.3.3 Étapes 14 à 18

14. Initialiser la couche d'index 4
15. Afficher le canevas
16. Imprimer l'aire du canevas (doit être égale à $1 + 1$ soit 2)
17. Réinitialiser le canevas
18. Afficher le canevas