UNIVERSITÉ DE SHERBROOKE

Faculté de génie

Département de génie électrique et génie informatique

**Rapport APP1**

Modélisation et programmation orientées objets

APP1

Présenté à

Eugène Morin

Domingo Palao Munoz

Étienne Lepage-Lepitre

Présenté par

Équipe numéro 2

Mathieu Désautels - DESM1210

Vincent Kilanowski – KILV1201

Sherbrooke – 16 janvier 2023

Tables des matières :

[1. Pseudocode de la fonction AddForme() 3](#_Toc124504963)

[2. Diagramme de cas d’utilisation 4](#_Toc124504964)

[3. Diagramme de classe 5](#_Toc124504965)

[4. Diagramme de séquence 6](#_Toc124504966)

[5. Diagramme d’état-transition 7](#_Toc124504967)

[6. Plan de test 8](#_Toc124504968)

[6.1 Test de validation du cas #1 8](#_Toc124504969)

[6.2 Test de validation du cas #2 8](#_Toc124504970)

Tables des illustrations :

[Tableau 1 : Tableau de tests du cas #1 8](#_Toc124504959)

[Tableau 2 : Tableau de tests du cas #2 8](#_Toc124504960)

# Pseudocode de la fonction AddForme()

**FONCTION** addForme( tforme\* ) : succès

// tforme ( Forme\* ) : C’est la forme à être ajouter dans le vecteur. Le t sert à avoir une variable différente que celle du nom de la classe.

// succès ( booléen ) : retour booléen lors de la fin de la fonction, *vrai* si aucun problème, *faux* si erreur.

// pvecteur ( tableau de Forme\* ) : taille capacité, pointeur vers le vecteur contenant toute les formes

**DÉBUT**

// taille (entier) : valeur privée de la classe Vecteur qui correspond au nombre d’élément dans le vecteur au moment du test

// capacite (entier) : valeur privée de la classe Vecteur qui correspond au nombre d’emplacement disponible dans le vecteur pour stocker des formes

// \*vecteurTemp ( tableau de Forme\* ) : taille capacite, **nouveau** vecteur dynamique temporaire qui sera utiliser dans l’opération de doublage

// i (entier) : compteur

**SI** tforme est égal à *NULL* **ALORS**

succès := *faux*

Retourner succès

**SINON SI** taille est égal à capacite **ALORS**

Capacite := capacite \* 2

**POUR** i := 0 **À** capacite / 2 **PAR PAS DE** 1

vecteurTemp[ i ] := pvecteur[ i ]

**SUPPRIMER** élément pointer parpvecteur

pvecteur := vecteurTemp

pvecteur[ taille ] := tforme

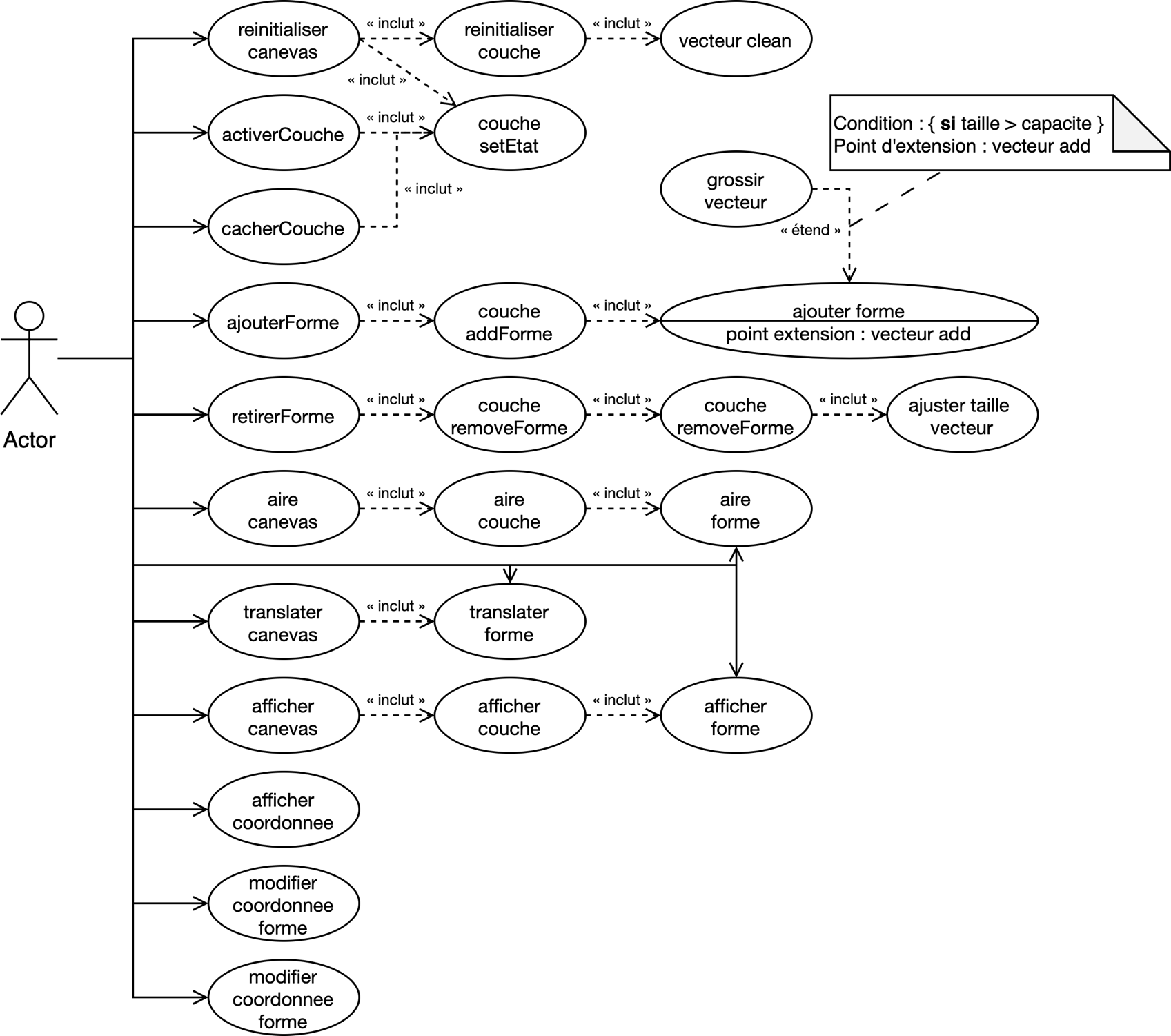
taille := taille + 1

succès := *vrai*

Retourner succès

**FIN**

# Diagramme de cas d’utilisation



# Diagramme de classe

# Diagramme de séquence

# Diagramme d’état-transition

# Plan de test

## Test de validation du cas #1

Tableau : Tableau de tests du cas #1

|  |  |  |  |
| --- | --- | --- | --- |
| **Numéro de test (Étapes)** | **Description** | **Résultat attendu** | **Réussite** |
| Étape #1 | Essayer d’activer la couche 5. | Couche invalide. | Réussi |
| Étape #2 | Essayer d’activer la couche 6. | Couche invalide. | Réussi |
| Étape #3 | Ajout des formes :  Rectangle (x=0, y=0, l=1, h=1, aire=1)  Rectangle (x=0, y=0, l=1, h=1, aire=1) | Afficher les formes ajoutées. | Réussi |
| Étape #4 | Afficher le canevas et l’aire du canevas. | Afficher le canevas et l’aire du canevas. | Réussi |
| Étape #5 | Activer la couche 0 et ajouter les formes :  Carre (x=0, y=0, c=1, aire=1)  Carre (x=1, y=1, c=1, aire=1) | Couche 0 activer. Afficher les formes ajoutées. | Réussi |
| Étape #6 | Faire une translation de x=-10 y=-10. | Translation réussie. | Réussi |
| Étape #7 | Afficher le canevas et l’aire du canevas. | Afficher le canevas et l’aire du canevas. | Réussi |
| Étape #8 | Retirer les 2 premiers éléments. | Retirer la forme de la position 0 deux fois. | Réussi |
| Étape #9 | Afficher le canevas et l’aire du canevas. | Afficher le canevas et l’aire du canevas. | Réussi |
| Étape #10 | Réinitialiser le canevas. | Canevas réinitialiser. | Réussi |
| Étape #11 | Afficher le canevas et l’aire du canevas. | Afficher le canevas et l’aire du canevas. | Réussi |

## Test de validation du cas #2

Tableau : Tableau de tests du cas #2

|  |  |  |  |
| --- | --- | --- | --- |
| **Numéro de test (Étapes)** | **Description** | **Résultat attendu** | **Réussite** |
| Étape #1 | Activer la couche 3. | Couche activer. | Réussi |
| Étape #2 | Ajout des formes :  Rectangle (x=0, y=0, l=1, h=1, aire=1)  … Rectangle (x=0, y=0, l=15, h=15, aire=225) | Afficher les formes ajoutées. | Réussi |
| Étape #3 | Activer la couche 2. | Couche activer. | Réussi |
| Étape #4 | Ajout des formes :  Cercle (x=0, y=0, r=7, aire=43.9823)  Carre (x=4, y=4, c=5, aire=25) | Afficher les formes ajoutées. | Réussi |
| Étape #5 | Afficher le canevas et l’aire du canevas. | Afficher le canevas et l’aire du canevas. | Réussi |
| Étape #6 | Activer la couche 3 et retirer les formes à la position 10 et 11. | Retirer la forme de la position 10 deux fois. | Réussi |
| Étape #7 | Afficher le canevas et l’aire du canevas. | Afficher le canevas et l’aire du canevas. | Réussi |
| Étape #8 | Activer la couche 4. | Couche activer. | Réussi |
| Étape #9 | Test si impossible de retirer avec un mauvais index. Essayer de retirer la forme à l’index 10. | Impossible de retirer. | Réussi |
| Étape #10 | Réinitialiser le canevas. | Canevas réinitialiser. | Réussi |
| Étape #11 | Afficher le canevas et l’aire du canevas. | Afficher le canevas et l’aire du canevas. | Réussi |
| Étape #12 | Teste si la couche de base active est bien la couche 0. Ajouter les formes :  Cercle (x=0, y=0, r=7, aire=43.9823)  Carre (x=4, y=4, c=5, aire=25) | Afficher les formes ajoutées. | Réussi |
| Étape #13 | Afficher le canevas et l’aire du canevas. | Afficher le canevas et l’aire du canevas. | Réussi |