

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

RAPPORT APP1

Modélisation et programmation orientées objets
APP1

Présenté à
M. Eugène Morin, M. Domingo Palao Munoz et M. Étienne Lepage-Lepitre

Présenté par
Équipe numéro 29
Youcef Labassi – laby1302
Alexandre Rousseau – roua0701

Sherbrooke – 16 janvier 2023

TABLE DES MATIÈRES

1.	Diagrammes UML pertinents	4
1.1	Diagramme de cas d'utilisation	4
1.2	Diagramme de classes	4
1.3	Diagramme de séquences	5
1.4	Diagramme d'états-transitions	5
2.	Pseudocode	6
3.	Plans de test sommaire	7
3.1	Tests unitaires Forme	7
3.2	Tests unitaires Vecteur	7
3.3	Tests unitaires Couche	8
3.4	Tests unitaires Canevas	8

LISTE DES FIGURES

Figure 1: Diagramme de cas d'utilisation	4
Figure 2: Diagramme de classes	4
Figure 3: Diagramme de séquences	5
Figure 4: Diagramme d'états-transitions	5

LISTE DES TABLEAUX

Tableau 1: Tests unitaires de la classe Forme	7
Tableau 2: Tests unitaires de la classe Vecteur	7
Tableau 3: Tests unitaires de la classe Couche	8
Tableau 4: Tests unitaires de la classe Canevas	8

1. DIAGRAMMES UML PERTINENTS

1.1 DIAGRAMME DE CAS D'UTILISATION

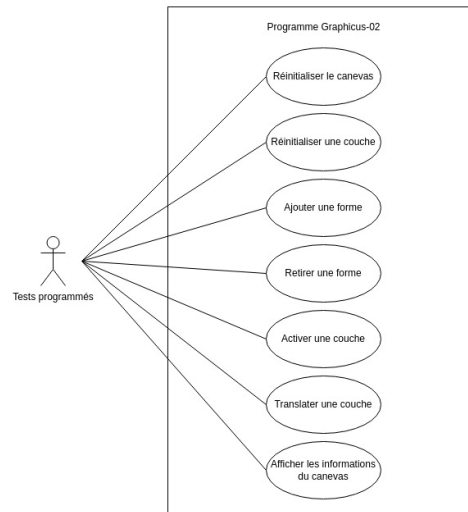


Figure 1: Diagramme de cas d'utilisation

1.2 DIAGRAMME DE CLASSES

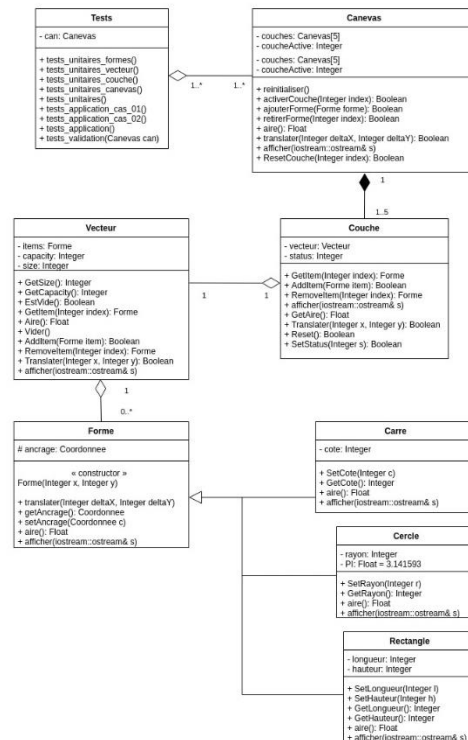


Figure 2: Diagramme de classes

1.3 DIAGRAMME DE SÉQUENCES

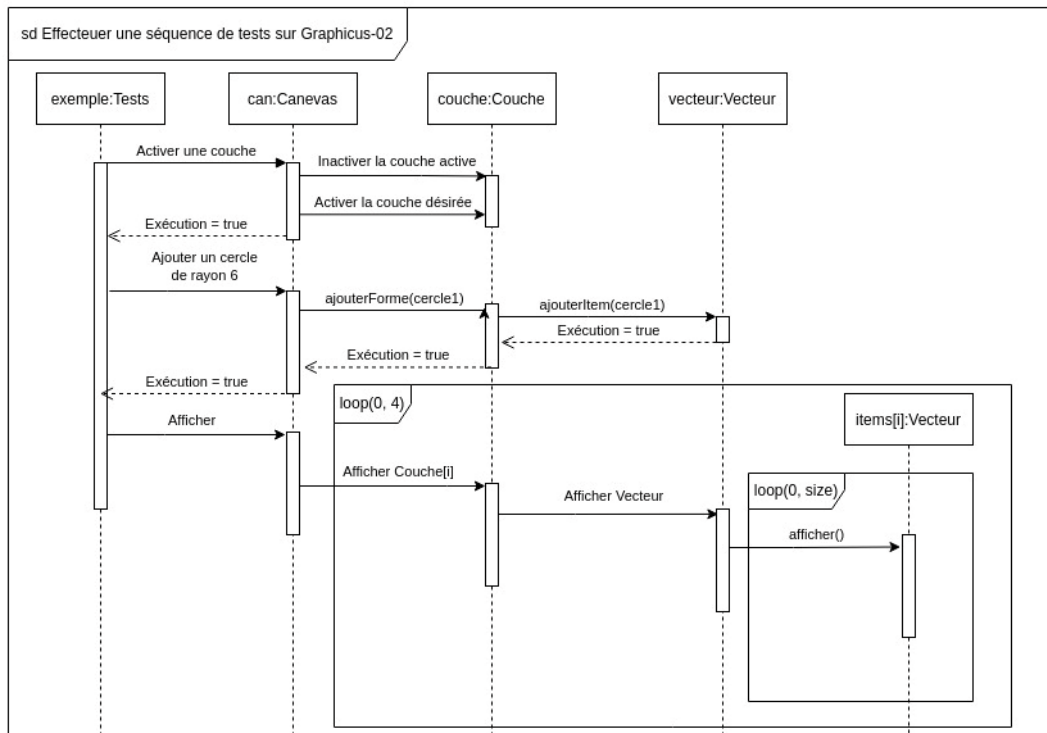


Figure 3: Diagramme de séquences

1.4 DIAGRAMME D'ÉTATS-TRANSITIONS

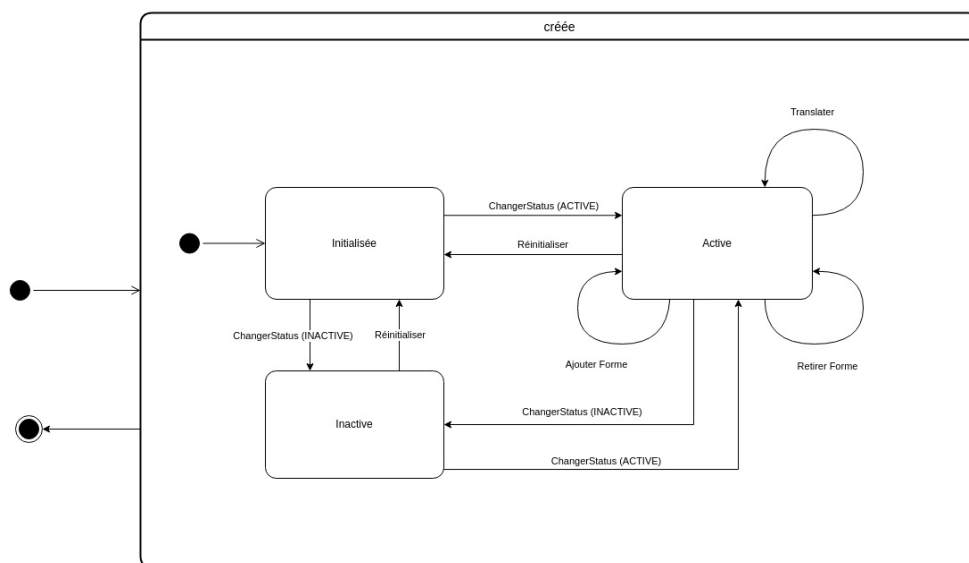


Figure 4: Diagramme d'états-transitions

2. PSEUDOCODE

FONCTION AddItem(Forme it): résultat

// it (forme)

// résultat (booléen) : valeur de retour

DÉBUT

 // size (entier)

 // capacity (entier)

 // cpyItems[capacity] (tableau de formes)

 // items[capacity] (tableau de formes)

SI (it = NULL) **ALORS**

 Retourner faux

SI (size > capacity) **ALORS**

 capacity := capacity*2

POUR i:=0 **À** i:=size-2

 cpyItems[i] := items[i]

 items[capacity] := cpyItems

 items[size-1] := it

 Retourner vrai

FIN

3. PLANS DE TEST SOMMAIRE

3.1 TESTS UNITAIRES FORME

Tableau 1: Tests unitaires de la classe Forme

Entrées	Résultats attendus	Résultats obtenus
Rectangle(x=0, y=0, l=5, h=8)	Rectangle(x=0, y=0, l=5, h=8, aire=40)	Rectangle(x=0, y=0, l=5, h=8, aire=40)
Cercle(x=2, y=7, r=3)	Cercle(x=2, y=7, r=3, aire=28.2743)	Cercle(x=2, y=7, r=3, aire=28.2743)
Carre(x=-5, y=1, c=1)	Carre(x=-5, y=1, c=1, aire=1)	Carre(x=-5, y=1, c=1, aire=1)
Carre(x=0, y=0, c=2.46)	Carre(x=0, y=0, c=2, aire=4)	Carre(x=0, y=0, c=2, aire=4)
Rectangle(x=-1, y=-1, l=6, h=2)	Rectangle(x=-1, y=-1, l=6, h=2, aire=12)	Rectangle(x=-1, y=-1, l=6, h=2, aire=12)

3.2 TESTS UNITAIRES VECTEUR

Tableau 2: Tests unitaires de la classe Vecteur

Entrées	Résultats attendus	Résultats obtenus
Rectangle(x=0, y=0, l=5, h=8, aire=40)	Taille du vecteur: 6	Taille du vecteur: 6
Cercle(x=0, y=0, r=1, aire=3.14159)	Capacité du vecteur: 8	Capacité du vecteur: 8
Carre(x=0, y=0, c=1, aire=1)		
Carre(x=0, y=0, c=1, aire=1)		
Carre(x=0, y=0, c=1, aire=1)		
Carre(x=0, y=0, c=1, aire=1)		

3.3 TESTS UNITAIRES COUCHE

Tableau 3: Tests unitaires de la classe Couche

Entrées	Résultats attendus	Résultats obtenus
Ajout de deux items à l'état active	L'aire de la couche est: 2	L'aire de la couche est: 2
Rectangle(x=0, y=0, l=1, h=1, aire=1)		
Carre(x=0, y=0, c=1, aire=1)		

3.4 TESTS UNITAIRES CANEVAS

Tableau 4: Tests unitaires de la classe Canevas

Entrées	Résultats attendus	Résultats obtenus
Activer la couche 2 et ajouter un rectangle unitaire	----- Couche 0	----- Couche 0
	Couche inactive	Couche inactive
	----- Couche 1	----- Couche 1
	Couche initialisee	Couche initialisee
	----- Couche 2	----- Couche 2
	Rectangle(x=0, y=0, l=1, h=1, aire=1)	Rectangle(x=0, y=0, l=1, h=1, aire=1)
	----- Couche 3	----- Couche 3
	Couche initialisee	Couche initialisee
	----- Couche 4	----- Couche 4
	Couche initialisee	Couche initialisee
	L'aire du canevas est: 1	L'aire du canevas est: 1