

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

RAPPORT APP1

Modélisation et programmation orientées objets
GEN 241

Présenté à
Équipe de formateurs de la session S2

Présenté par
Raphael Bouchard – bour0703
Alexis Guérard – guea0902

Sherbrooke – 15 janvier 2023

TABLE DES MATIÈRES

1.	Développement	1
1.1	Diagrammes UML	1
1.1.1	Diagrammes UML de cas d'utilisation	1
1.1.2	Diagrammes UML de classes	2
1.1.3	Diagramme UML de séquences	3
1.1.4	Diagramme UML états-transitions	3
1.2	Pseudocode fonction insertion de la classe vecteur	4
1.3	Plan de test	5

LISTE DES FIGURES

Figure 1 : Diagrammes UML de cas d'utilisation	1
Figure 2 : Diagrammes UML de classes	2
Figure 3 : Diagramme UML de séquences	3
Figure 4 : Diagramme UML États-transitions	3

LISTE DES TABLEAUX

Tableau 1: Plan de test	5
-------------------------	---

1. DÉVELOPPEMENT

1.1 DIAGRAMMES UML

1.1.1 DIAGRAMMES UML DE CAS D'UTILISATION



Figure 1 : Diagrammes UML de cas d'utilisation

1.1.2 DIAGRAMMES UML DE CLASSES

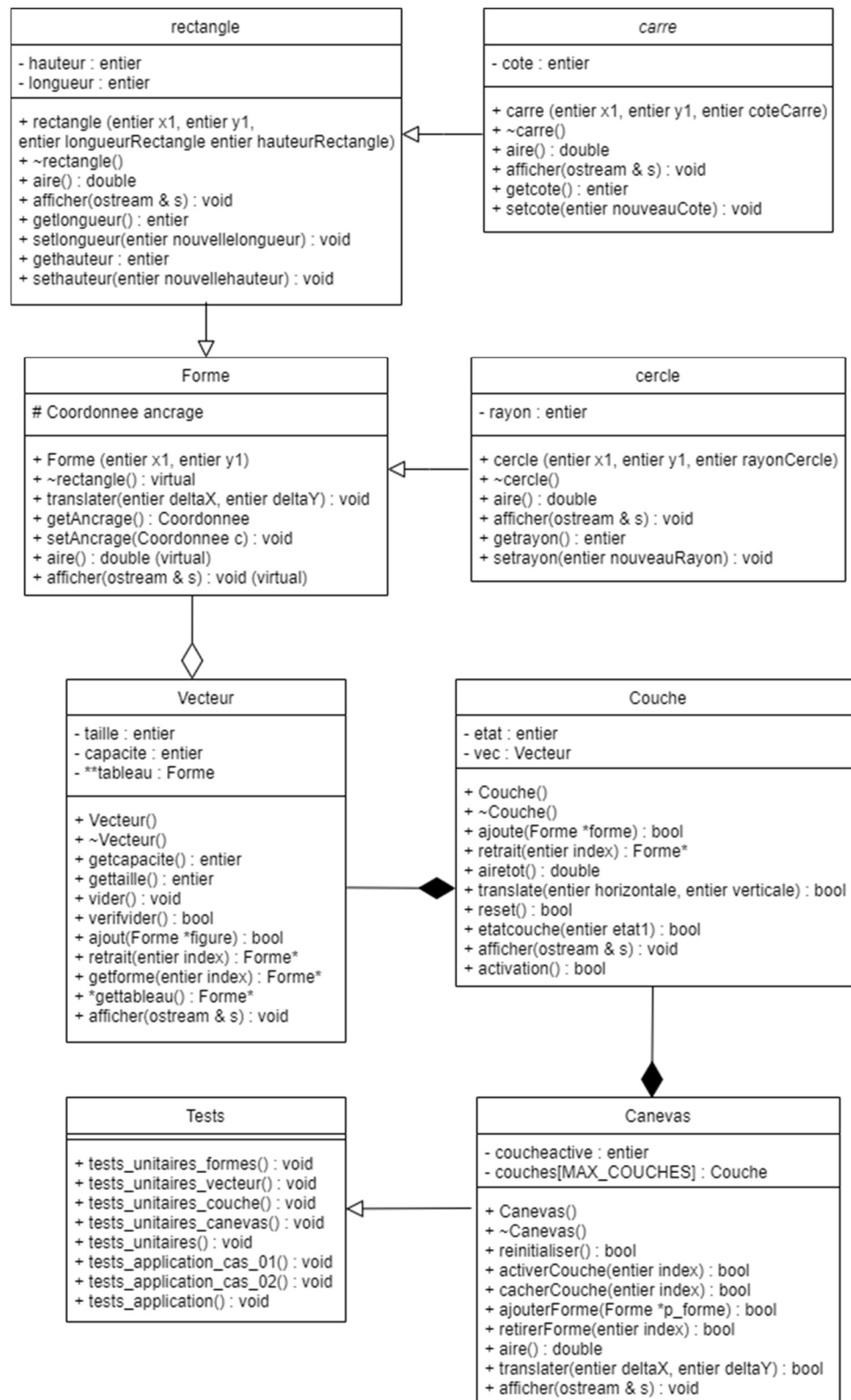


Figure 2 : Diagrammes UML de classes

1.1.3 DIAGRAMME UML DE SÉQUENCES

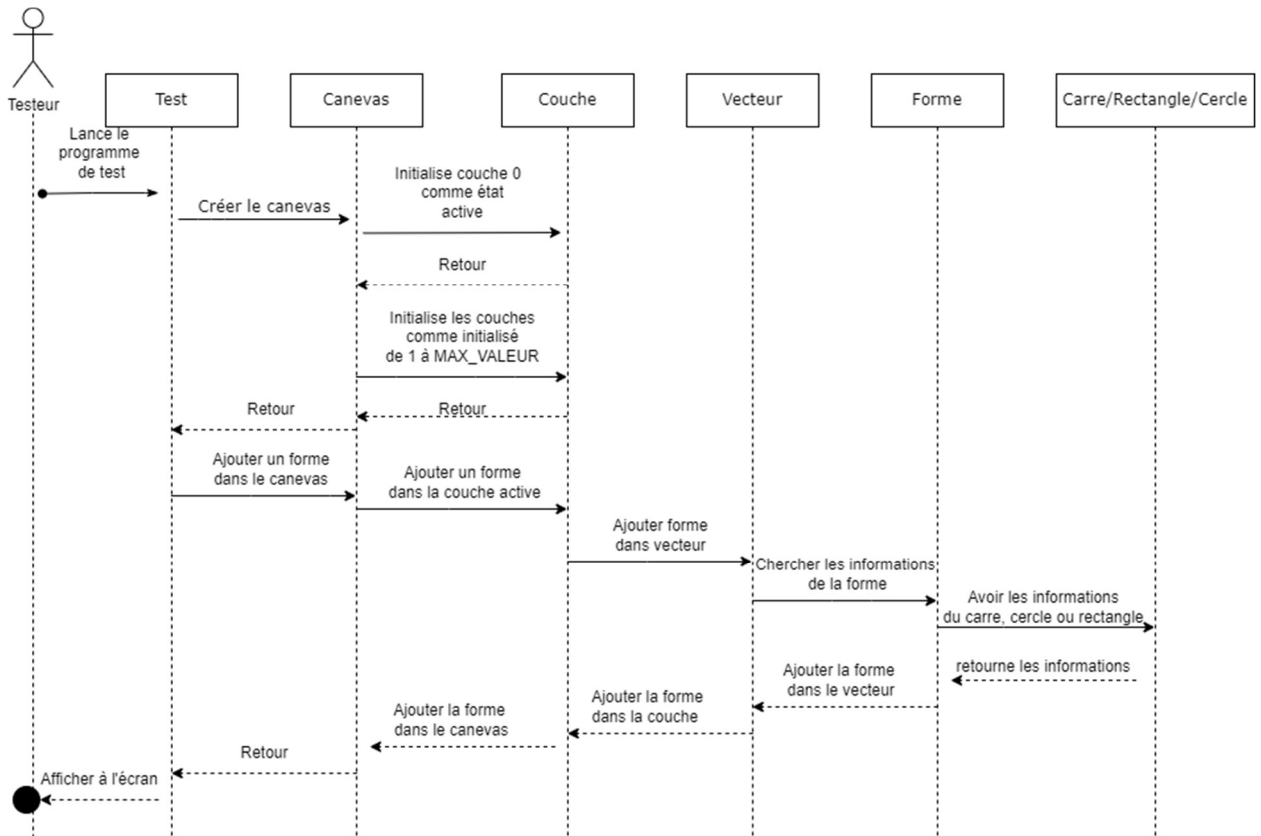


Figure 3 : Diagramme UML de séquences

1.1.4 DIAGRAMME UML ÉTATS-TRANSITIONS

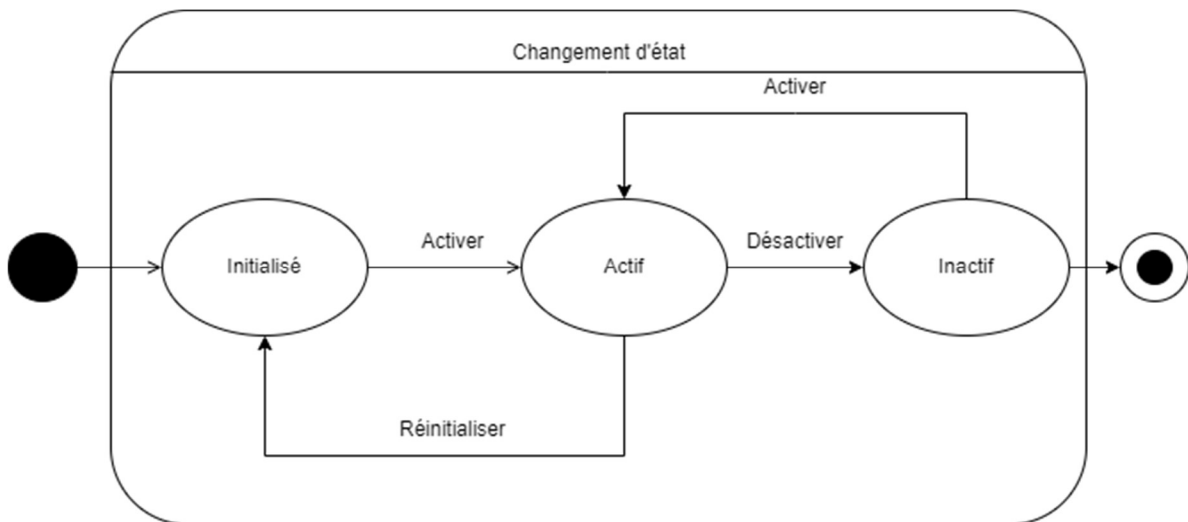


Figure 4 : Diagramme UML États-transitions

1.2 PSEUDOCODE FONCTION INSERTION DE LA CLASSE VECTEUR

FONCTION ajout (figure) : retour

// La fonction permet d'ajouter des formes dans une liste

//figure (Forme) : pointeur de la classe Forme

//retour (booléen) :

DEBUT

//taille (entier) :

//capacite (entier) :

//tableau (Forme) : pointeur de tableau de classe Forme de longueur (capacite)

//tableau2 (Forme) : pointeur de tableau de la classe Forme de longueur (capacite)

// i (entier) :

retour := vrai

SI taille = capacite **ALORS**

 capacite := capacite * 2

POUR i :=0 **À** taille

 tableau2[i] := tableau[i]

 supprimer tableau

 tableau := tableau2

tableau[taille] = figure

taille := taille + 1

SI tableau[taille] est différent de nulle **ALORS**

 retour := faux

FIN

1.3 PLAN DE TEST

Tableau 1: Plan de test

Fonction et paramètre	Attendue	Résultat obtenue
TEST 1	----Couche 0	----Couche 0
Canevas canevas;	Couche initialisee	Couche initialisee
Canevas.activerCouche(3);	----Couche 1	----Couche 1
Forme*rectangle1=new rectangle(0,1,4,3);	Couche initialisee	Couche initialisee
	----Couche 2	----Couche 2
Forme* carre1 = new carre(4,3,5);	Rectangle(x=0, y=3, l=7, h=5, aire=35)	Rectangle(x=0, y=3, l=7, h=5, aire=35)
Forme* cercle1 = new cercle(4,9,2);	----Couche 3	----Couche 3
canevas.ajouterForme(rectangle1);	Rectangle(x=0, y=1, l=4, h=3, aire=12)	Rectangle(x=0, y=1, l=4, h=3, aire=12)
canevas.ajouterForme(carre1);	Carre(x=4, y=3, c=5, aire=25)	Carre(x=4, y=3, c=5, aire=25)
canevas.ajouterForme(cercle1);	Cercle(x=4, y=9, r=2, aire=12.5664)	Cercle(x=4, y=9, r=2, aire=12.5664)
canevas.activerCouche(2);	----Couche 4	----Couche 4
Forme* rectangle2 = new rectangle(0,3,7,5);	Couche initialisee	Couche initialisee
canevas.ajouterForme(rectangle2);	Aire totale: 84.5664	Aire totale: 84.5664
canevas.afficher(cout);		
cout<<"Aire totale: "<< canevas.aire() <<endl;		
TEST 2	----Couche 0	----Couche 0
canevas.activerCouche(0);	Rectangle(x=0, y=0, l=1, h=1, aire=1)	Rectangle(x=0, y=0, l=1, h=1, aire=1)
Forme* rectangle3 = new rectangle(0,0,1,1);	Carre(x=0, y=0, c=3, aire=9)	Carre(x=0, y=0, c=3, aire=9)
Forme* carre2 = new carre(0,0,3);	----Couche 1	----Couche 1
canevas.ajouterForme(rectangle3);	Couche initialisee	Couche initialisee
canevas.ajouterForme(carre3);	----Couche 2	----Couche 2
canevas.activerCouche(3);	Rectangle(x=0, y=3, l=7, h=5, aire=35)	Rectangle(x=0, y=3, l=7, h=5, aire=35)
canevas.translater(3,3);	----Couche 3	----Couche 3
canevas.afficher(cout);	Rectangle(x=3, y=4, l=4, h=3, aire=12)	Rectangle(x=3, y=4, l=4, h=3, aire=12)
cout << "" <<endl;	Carre(x=7, y=6, c=5, aire=25)	Carre(x=7, y=6, c=5, aire=25)
cout<<"Aire totale: "<< canevas.aire() <<endl;	Cercle(x=7, y=12, r=2, aire=12.5664)	Cercle(x=7, y=12, r=2, aire=12.5664)
	----Couche 4	----Couche 4
	Couche initialisee	Couche initialisee

TEST 3

```
canevas.activerCouche(0);
canevas.retirerForme(1);
canevas.afficher(cout);
cout<<"Aire totale: "<< canevas.aire()
<<endl;
```

Aire totale : 94.5664

```
----Couche 0
Rectangle(x=0, y=0, l=1, h=1,
aire=1)
----Couche 1
Couche initialisee
----Couche 2
Rectangle(x=0, y=3, l=7, h=5,
aire=35)
----Couche 3
Rectangle(x=3, y=4, l=4, h=3,
aire=12)
Carre(x=7, y=6, c=5, aire=25)
Cercle(x=7, y=12, r=2,
aire=12.5664)
----Couche 4
Couche initialisee
Aire totale: 85.5664
```

Aire totale : 94.5664

```
----Couche 0
Rectangle(x=0, y=0, l=1, h=1,
aire=1)
----Couche 1
Couche initialisee
----Couche 2
Rectangle(x=0, y=3, l=7, h=5,
aire=35)
----Couche 3
Rectangle(x=3, y=4, l=4, h=3,
aire=12)
Carre(x=7, y=6, c=5, aire=25)
Cercle(x=7, y=12, r=2,
aire=12.5664)
----Couche 4
Couche initialisee
Aire totale: 85.5664
```

TEST 4

```
canevas.reinitialiser();
canevas.afficher(cout);
cout<<"Aire totale: "<< canevas.aire()
<<endl;
```

```
----Couche 0
Couche initialisee
----Couche 1
Couche initialisee
----Couche 2
Couche initialisee
----Couche 3
Couche initialisee
----Couche 4
Couche initialisee
Aire totale : 0
```

```
----Couche 0
Couche initialisee
----Couche 1
Couche initialisee
----Couche 2
Couche initialisee
----Couche 3
Couche initialisee
----Couche 4
Couche initialisee
Aire totale : 0
```