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 4](#_Toc124702703)

[1.1 Diagrammes UML 4](#_Toc124702704)

[1.1.1 Diagrammes UML de cas d’utilisation 4](#_Toc124702705)

[1.1.2 Diagrammes UML de classes 5](#_Toc124702706)

[1.1.3 Diagramme UML de séquences 6](#_Toc124702707)

[1.1.4 Diagramme UML états-transitions 6](#_Toc124702708)

[1.2 Pseudocode fonction insertion de la classe vecteur 7](#_Toc124702709)

[1.3 Plan de test 8](#_Toc124702710)

Liste des figures

[Figure 1 : Diagrammes UML de cas d’utilisation 4](#_Toc124701996)

[Figure 2 : Diagrammes UML de classes 5](#_Toc124701997)

[Figure 3 : Diagramme UML de séquences 6](#_Toc124701998)

[Figure 4 : Diagramme UML États-transitions 6](#_Toc124701999)

Liste des tableaux

[Tableau 1: Plan de test 8](#_Toc124702005)

# Développement

## Diagrammes UML

### Diagrammes UML de cas d’utilisation

Figure  : Diagrammes UML de cas d’utilisation

### Diagrammes UML de classes

Figure : Diagrammes UML de classes

### Diagramme UML de séquences

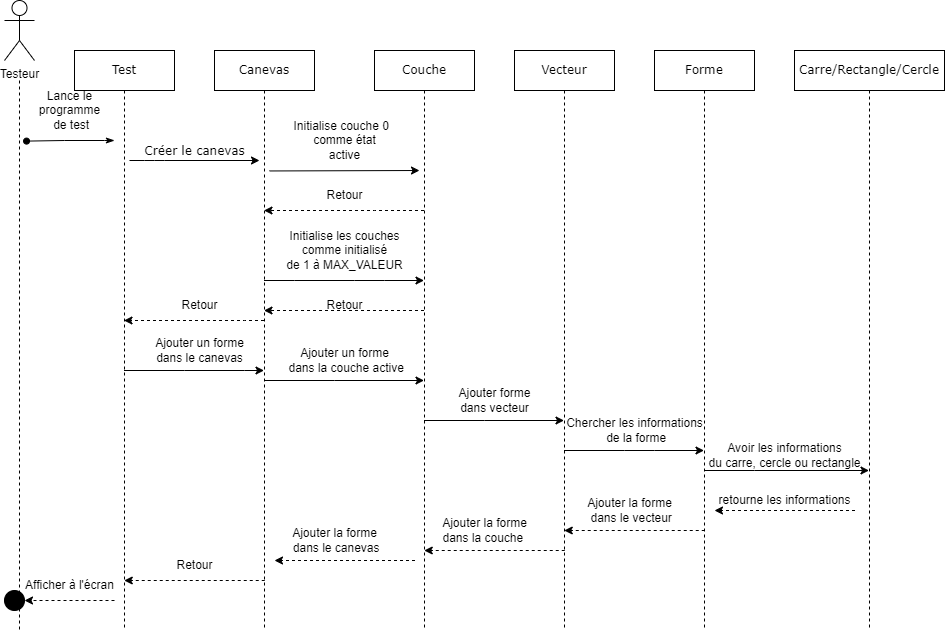
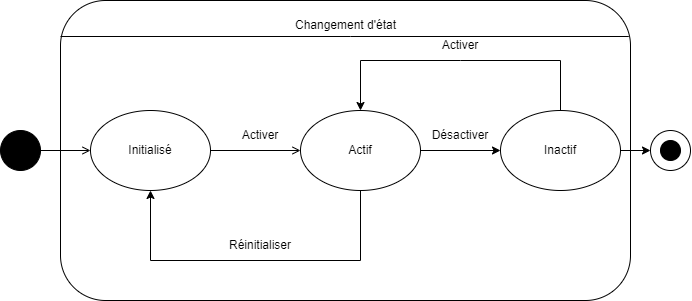


Figure  : Diagramme UML de séquences

### Diagramme UML états-transitions



**Figure 4 : Diagramme UML États-transitions**

## 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**

## Plan de test

Tableau 1: Plan de test

|  |  |  |  |
| --- | --- | --- | --- |
| Fonction et paramètre | | Attendue | Résultat obtenue |
| **TEST 1**  Canevas canevas;  Canevas.activerCouche(3);  Forme\*rectangle1=new rectangle(0,1,4,3);  Forme\* carre1 = new carre(4,3,5);  Forme\* cercle1 = new cercle(4,9,2);  canevas.ajouterForme(rectangle1);  canevas.ajouterForme(carre1);  canevas.ajouterForme(cercle1);  canevas.activerCouche(2);  Forme\* rectangle2 = new rectangle(0,3,7,5);  canevas.ajouterForme(rectangle2);  canevas.afficher(cout);  cout<<"Aire totale: "<< canevas.aire() <<endl; | ----Couche 0  Couche initialisee  ----Couche 1  Couche initialisee  ----Couche 2  Rectangle(x=0, y=3, l=7, h=5, aire=35)  ----Couche 3  Rectangle(x=0, y=1, l=4, h=3, aire=12)  Carre(x=4, y=3, c=5, aire=25)  Cercle(x=4, y=9, r=2, aire=12.5664)  ----Couche 4  Couche initialisee  Aire totale: 84.5664 | | ----Couche 0  Couche initialisee  ----Couche 1  Couche initialisee  ----Couche 2  Rectangle(x=0, y=3, l=7, h=5, aire=35)  ----Couche 3  Rectangle(x=0, y=1, l=4, h=3, aire=12)  Carre(x=4, y=3, c=5, aire=25)  Cercle(x=4, y=9, r=2, aire=12.5664)  ----Couche 4  Couche initialisee  Aire totale: 84.5664 |
| **TEST 2**  canevas.activerCouche(0);  Forme\* rectangle3 = new rectangle(0,0,1,1);  Forme\* carre2 = new carre(0,0,3);  canevas.ajouterForme(rectangle3);  canevas.ajouterForme(carre3);  canevas.activerCouche(3);  canevas.translater(3,3);  canevas.afficher(cout);  cout << "" <<endl;  cout<<"Aire totale: "<< canevas.aire() <<endl; | ----Couche 0  Rectangle(x=0, y=0, l=1, h=1, aire=1)  Carre(x=0, y=0, c=3, aire=9)  ----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 : 94.5664 | | ----Couche 0  Rectangle(x=0, y=0, l=1, h=1, aire=1)  Carre(x=0, y=0, c=3, aire=9)  ----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 : 94.5664 |
| **TEST 3**  canevas.activerCouche(0);  canevas.retirerForme(1);  canevas.afficher(cout);  cout<<"Aire totale: "<< canevas.aire() <<endl; | ----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 | | ----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 |