

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

RAPPORT APP1

Modélisation et programmation orientées objets
APP1 : GEN241

Présenté à
Charles-Antoine Brunet
Domingo Palao Munoz
Jean Lavoie

Présenté par
Équipe numéro 15
Pascal-Emmanuel Lachance – lacp3102
Olivier Rivard – rivo3201

Sherbrooke, 19 janvier 2021

TABLE DES MATIÈRES

| | | |
|-----------|--------------------------------|-----------|
| 1. | Diagrammes UML | 3 |
| 1.1 | Diagramme de classes (Forme) | 3 |
| 1.2 | Diagramme de classes | 4 |
| 1.3 | Diagramme de cas d'utilisation | 5 |
| 1.4 | Diagramme de séquences | 6 |
| 1.5 | Diagramme d'états-transitions | 7 |
| 2. | Pseudocode | 8 |
| 2.1 | Ajout Élément | 8 |
| 2.2 | Réallouer | 9 |
| 2.3 | Taille | 9 |
| 3. | Plan de tests | 10 |

1. DIAGRAMMES UML

1.1 DIAGRAMME DE CLASSES (FORME)

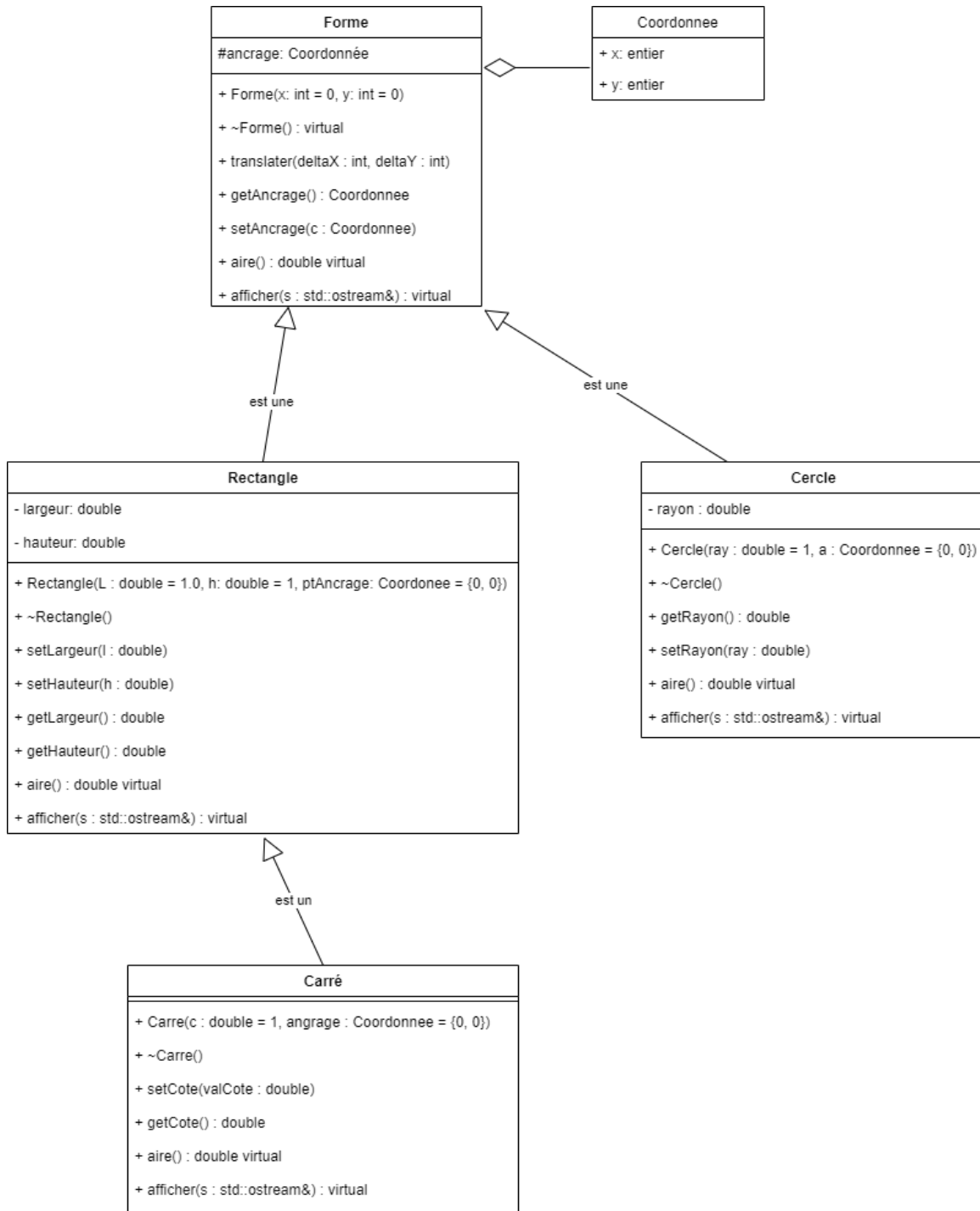


Figure 1 - Diagramme de classes centré sur la classe Forme

1.2 DIAGRAMME DE CLASSES

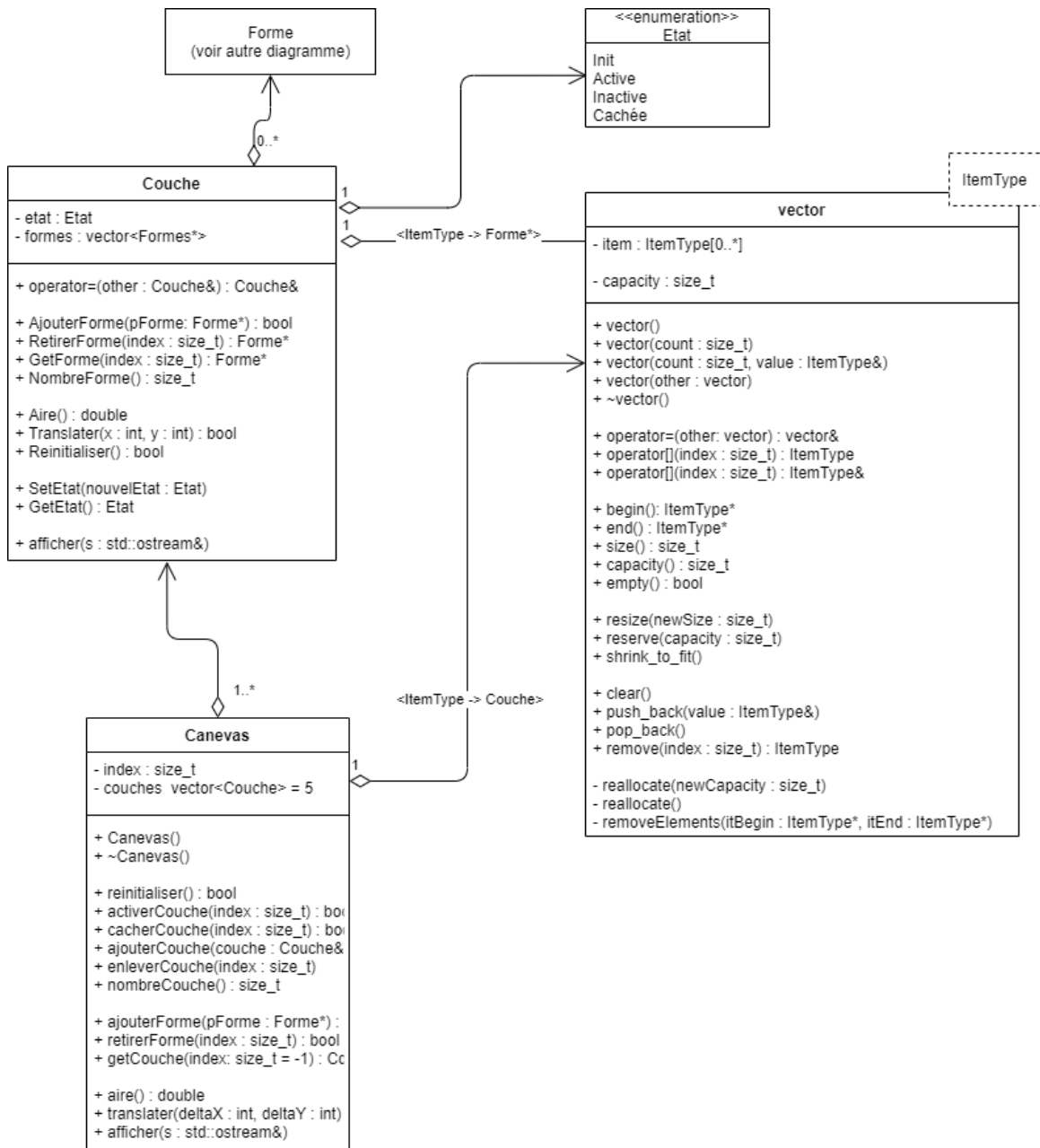
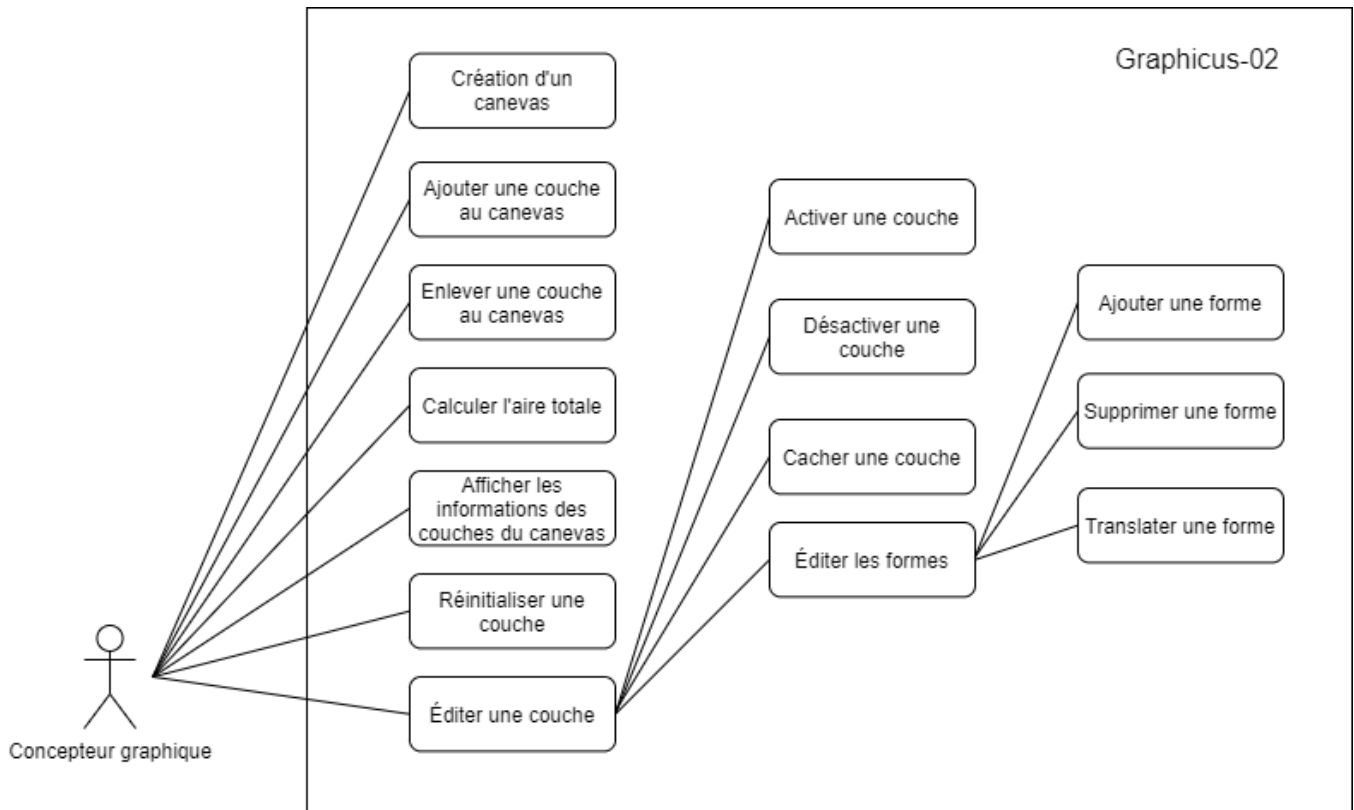
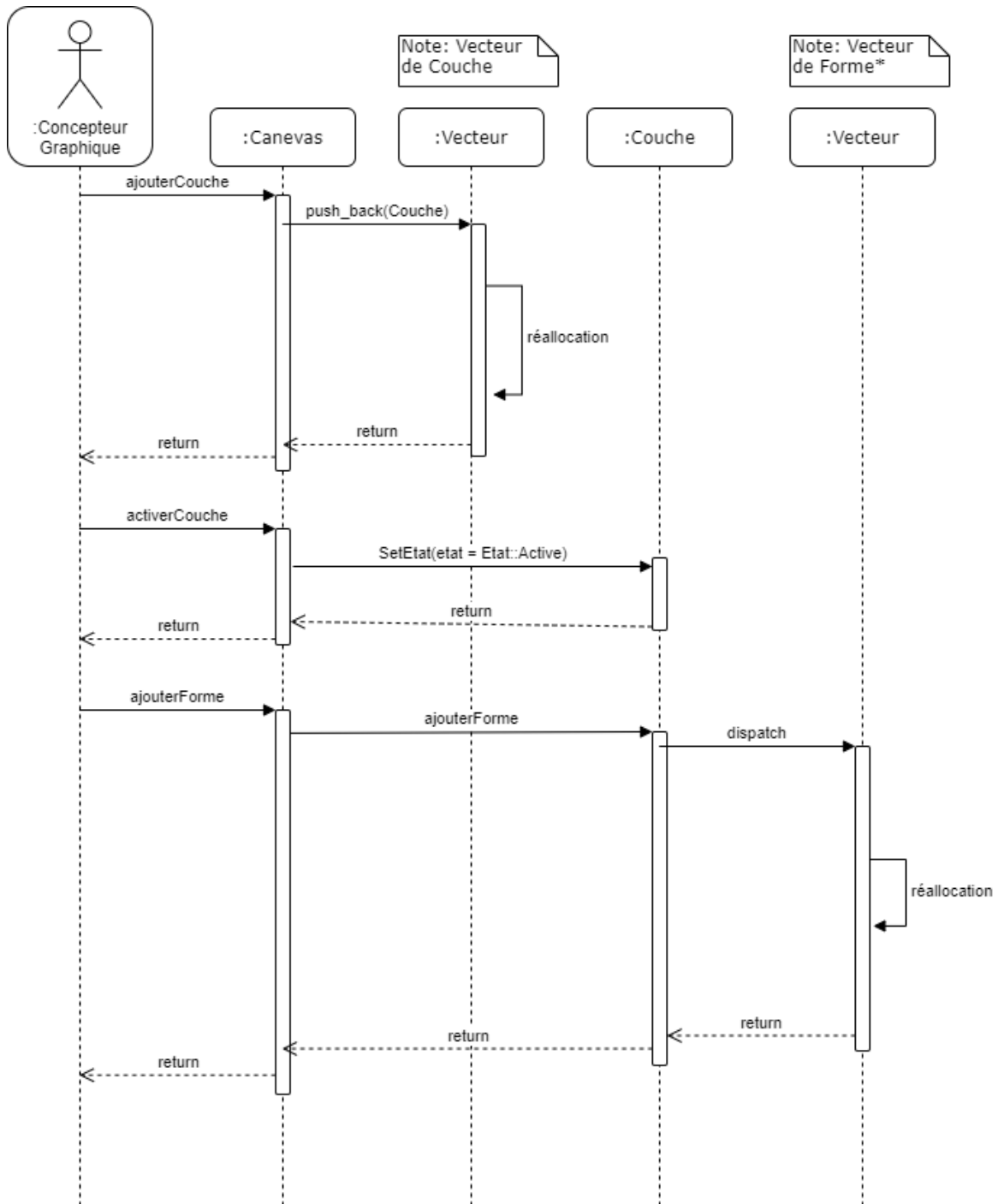


Figure 2 - Diagramme de classes général

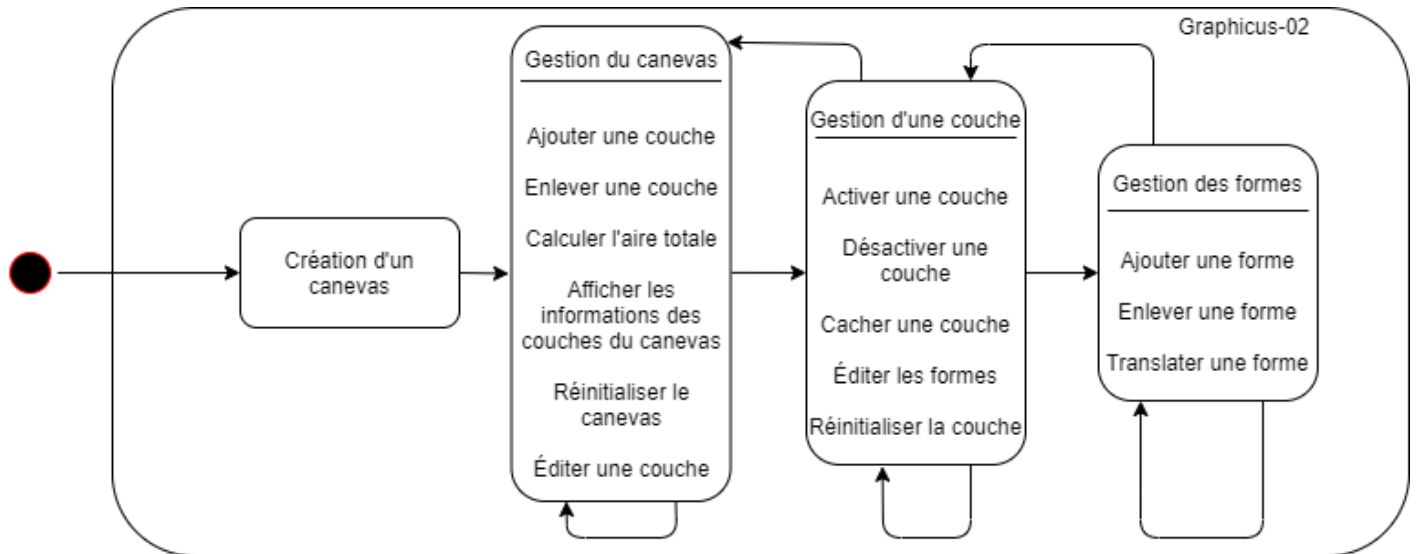
1.3 DIAGRAMME DE CAS D'UTILISATION



1.4 DIAGRAMME DE SÉQUENCES



1.5 DIAGRAMME D'ÉTATS-TRANSITIONS



2. PSEUDOCODE

2.1 AJOUTÉLÉMENT

```
1  // Variables globales:
2  // débutPointeur(pointeur) : Pointeur vers le début de l'espace mémoire alloué
3  // finPointeur(pointeur) :   Pointeur vers l'élément suivant le dernier élément
4  //                           dans l'espace mémoire alloué
5  // capacité(entier) :       Taille totale de l'espace mémoire présentement
6  //                           alloué
7
8  FONCTION AjoutElement(valeur) : succès
9      // valeur(pointeur) : Pointeur vers le nouvel élément à ajouter au vecteur
10     // succès(booléen) : Résultat si l'élément a réussi à être ajouté
11  DÉBUT
12     // Il n'y a pas la place pour le nouvel élément
13     SI Taille() + 1 > capacité ALORS
14         // On alloue un bloc de mémoire 2 fois plus grand
15         succès = Réallouer(capacité * 2)
16
17         // En cas d'échec de réallocation, retourner immédiatement
18         SI succès = faux
19             Retourner succès
20
21     // Incrimer la nouvelle valeur à la fin du tableau
22     Écrire valeur dans finPointeur
23     finPointeur:= finPointeur + 1
24     succès = vrai
25     Retourner succès
26  FIN
```


2.2 RÉALLOCUER

```
1  // Cette fonction en pseudocode ne s'attaque qu'au scénario où la capacité
2  // augmente, mais elle pourrait diminuer.
3  FONCTION Réallouer(nouvelleCapacité) : succès
4      // Alloue un nouveau bloc de mémoire, copie les éléments de l'ancien bloc
5      // vers le nouveau bloc, et supprime l'ancien bloc
6
7      // nouvelleCapacité(entier) : Taille du nouveau bloc mémoire
8      // succès(booléen) :      Résultat si le nouveau bloc mémoire a pu
9      //                          être alloué
10 DÉBUT
11     // nouveauBloc(pointeur) : Pointeur vers le nouveau bloc mémoire
12
13     nouveauBloc := Alloue nouvelleCapacité dans la mémoire dynamique
14
15     SI nouveauBloc = 0 ALORS
16         // Il y a eu une erreur dans l'allocation du bloc mémoire
17         succès = faux
18         Retourne succès
19
20     // Copie les données de l'ancien bloc vers le nouveau bloc
21     POUR toutes les valeurs de l'ancien bloc
22         Copie la valeur dans le nouveau bloc
23
24     Désalloue débutPointeur          // Supprime l'ancien bloc mémoire
25     débutPointeur := nouveauBloc
26     finPointeur := nouveauBloc + Taille()
27     capacité := nouvelleCapacité
28     succès = vrai
29     Retourne succès
30 FIN
```

2.3 TAILLE

```
1  FONCTION Taille() : taille
2      // Retourne le nombre d'éléments dans le vecteur
3      // taille(entier) : nombre d'éléments dans le vecteur
4  DÉBUT
5      taille := finPointeur - débutPointeur
6      Retourner taille
7  FIN
```

3. PLAN DE TESTS

| Nom du test | Condition(s) du test | Résultat attendu | Réussi (O/N) |
|---|---|--|--------------|
| Ajout de formes dans un vecteur vide | Instancier un vecteur, créer des formes et utiliser la méthode <code>push_back()</code> . | La console affiche toutes les bonnes formes à l'intérieur du vecteur (Valide donc la création des formes avec les constructeurs selon les mesures données). | O |
| Retirer des formes du vecteur qui en contient déjà | Avoir un vecteur contenant différentes formes afin d'utiliser les méthodes <code>pop_back()</code> et <code>remove()</code> . | Pour <code>pop_back()</code> , le nouveau vecteur s'est fait retirer sa dernière forme. Pour <code>remove()</code> , le nouveau vecteur contient un élément en moins et cet élément est celui spécifié dans les paramètres de la méthode. | O |
| Construire une copie d'un vecteur | Avoir un vecteur initial contenant déjà des formes. Utilisation du constructeur de copie du vecteur. | La console affiche le même contenu pour les deux vecteurs. | O |
| Capacité d'un vecteur | Réserver une capacité plus grande que la taille du vecteur et lui ajouter une forme. | Si la capacité du vecteur est plus grande que sa taille, la capacité ne doit pas augmenter. | O |
| Calculer l'aire totale d'une couche | Avoir une couche contenant plusieurs formes. Utiliser la méthode <code>Aire()</code> . | La console affiche l'aire total en sommation justement. | O |
| Translater une couche | Avoir une couche contenant plusieurs formes. Utiliser la méthode <code>Translater()</code> . | La console affiche les nouvelles informations des formes avec les points d'ancrage modifiés selon les paramètres de la méthode. | O |
| Changer l'état de la couche | Avoir une couche active afin de pouvoir changer son état avec une des méthodes (ex.: | La console affiche le nouvel état de la couche (selon l'exemple, cachée). | O |

| | | | |
|--|---|--|---|
| cacherCouche()). | | | |
| Création d'un canevas | Utiliser le constructeur par défaut pour instancier un canevas. | La console affiche le nombre de couches par défaut du canevas qui doit être de 5. | O |
| Ajout d'une couche au canevas | Utiliser la méthode ajouterCouche() avec un canevas. | La console affiche le nouveau nombre de couches, soit 6 dans notre cas. | O |
| Affichage d'une couche du canevas | Avoir un canevas avec au moins une couche. | La console affiche, s'il y en a, les formes dans la couche spécifiée ainsi que leurs informations. | O |