

0.1 Présentation du projet

C'est un projet qui permet de dessiner et manipuler des formes géométriques à l'aide d'un interpréteur de commandes qui permet d'interpréter les différentes commandes prédéfinies introduite par l'utilisateur afin d'afficher les résultats sur une fenêtre. La réalisation de ce projet nécessite l'utilisation d'un ensemble d'objets, fonctions et types définis dans la bibliothèque SFML.

0.2 Bibliothèque SFML

SFML fournit un ensemble de classes représentant des formes simples. Chaque forme possède sa propre classe, mais toutes dérivent de la même classe de base de sorte qu'elles partagent un ensemble de fonctionnalités communes. Chaque classe ajoute ensuite ses propres spécificités : un rayon pour la classe de cercles, une taille pour la classe de rectangles, des points pour la classe des polygones, etc. le module de fenêtrage de SFML fournit tout ce qu'il faut pour créer une fenêtre OpenGL et gérer ses événements, mais n'est d'aucune aide pour dessiner quoique ce soit. La seule option qu'il vous offre est d'utiliser OpenGL. Heureusement, SFML fournit un module graphique avec plein d'entités 2D, beaucoup plus simples à manipuler qu'OpenGL.

0.3 Exécution du projet

Pour voir l'ensemble des commandes possibles on fait appel à la commande help.

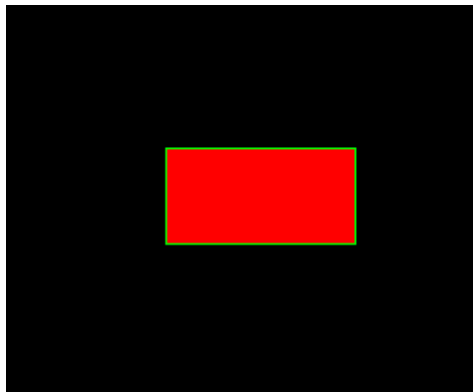
```
> help
[COMMANDES]
help
create [Forme_Type] [Forme_ID] [Coul_Fond] [Coul_Contour] [Epaisseur] [Position_x] [Position_y] <metriques*>
move [Forme_ID] [position_x] [position_y]
rotate [Forme_ID] {+|-} [angle_rotation]
color {background|contour} [Forme_ID] [Coul_ID]
contour [Forme_ID] {++|--}
scaleup [Forme_ID] [facteur_echelle]
scaledown [Forme_ID] [facteur_echelle]
perimetre [Forme_ID]
superficie [Forme_ID]
duplicate [Forme_ID] [New_Forme_ID] [position_x] [position_y]
list
test [Forme_ID_1] {>|<|=} [Forme_ID_2]
dist [Forme_ID_1][Forme_ID_2]
remove [Forme_ID]
removeall
cls
exit

[FORMES ET METRIQUES]
Forme_Type = rectangle|cercle|polygone
Metriques d'un rectangle : [Longeur] [Largeur]
Metriques d'un cercle : [Rayon]
Metriques d'un polygone : [Rayon] [Nb_Sommets]

[COULEURS]
{Coul_Fond|Coul_Contour|Coul_ID} = rouge|vert|bleu|noir|blanc|jaune
```

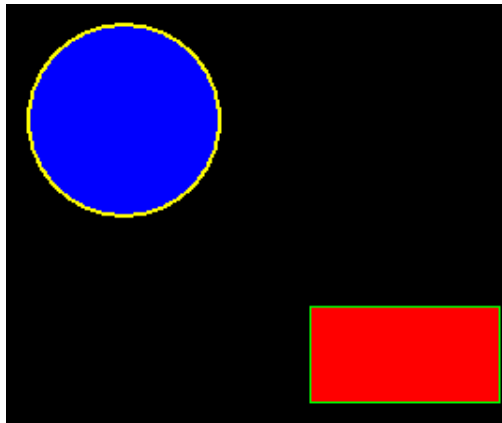
Comme montré dans la figure ci-dessus, pour créer une forme, il suffit d'utiliser la commande « create ». Cette commande prend en paramètres le type de la forme, l'identifiant qui doit être unique, la couleur du fond, la couleur du contour, l'épaisseur, la position en x, la position en y et la métrique. Si on choisit de créer un rectangle il faut donc fournir l'ensemble des paramètres nécessaires.

```
> create rectangle 1 rouge vert 1 200 200 100 50
Constructeur : Forme
Constructeur : Rectangle
Constructeur par copie : Forme
Affectation par copie: Rectangle
>
```



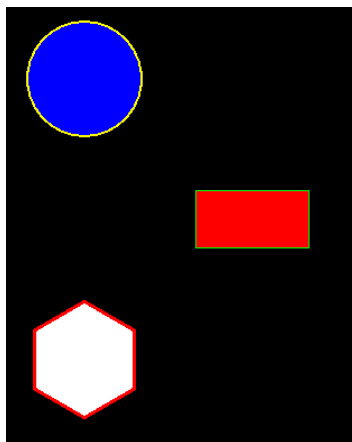
Maintenant on va créer un cercle avec la commande suivante :

```
> create cercle 2 bleu jaune 2 50 50 50
Constructeur : Forme
Constructeur : Cercle
Constructeur par copie : Forme
Affectation par copie: Cercle
>
```



Après on ajoute aussi un rectangle avec la commande suivante :

```
> create polygone 3 blanc rouge 3 50 300 50 6  
Constructeur : Forme  
Constructeur : Polygone regulier  
Constructeur par copie : Forme  
Affectation par copie: Polygone  
>
```

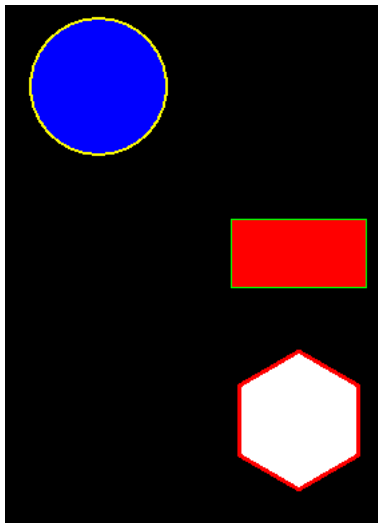


Après avoir créer un formes on peut lui appliquées un ensemble de fonctionnalités.

Move

Pour déplacer une forme, il est nécessaire d'indiquer l'identifiant de la forme, les nouvelles positions en x et y.

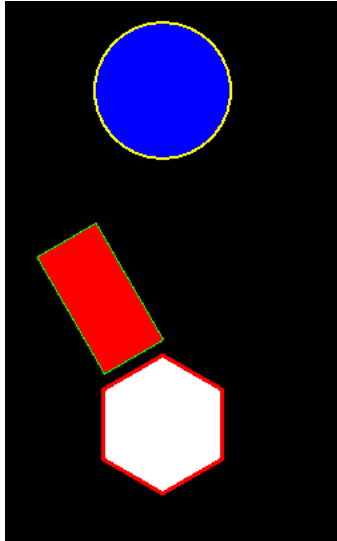
```
> move 3 200 300  
surcharge operateur << Vecteur  
>
```



Rotate

La rotation d'une forme se fait à l'aide de la commande «rotate », tout en indiquant l'identifiant de la forme, le signe de la rotation en plus + ou en moins - et l'angle de rotation.

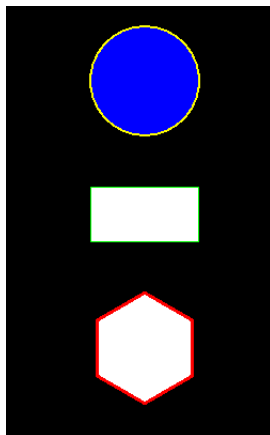
```
> rotate 1 + 60  
surcharge +  
Affectation par copie : Forme  
>
```



Color

La commande « color » permet de changer la couleur de background ou du contour, en indiquant l'identifiant et la couleur à appliquer.

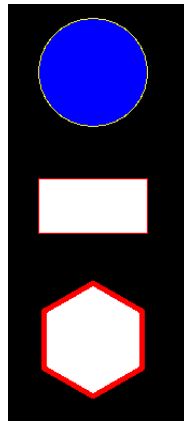
```
> color background 1 blanc
Surcharge operateur << Couleur
>
```



Contour

Pour incrémenter ou décrémenter l'épaisseur du contour en mentionnant l'identifiant de la forme et le signe ++ pour incrémenter et – pour décrémenter.

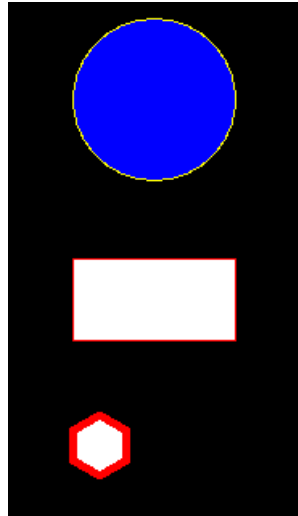
```
> contour 3 ++  
surcharge ++  
>
```



Scaleup et scaledown

On peut à l'aide de ces deux commandes augmenter ou diminuer la taille d'une forme.

```
> scaledown 3 3  
Constructeur par copie : Forme  
Affectation par copie: Polygone  
surcharge /  
Affectation par copie : Forme  
>
```



Superficie et perimetre

On peut calculer le périmètre et la surface d'une forme à l'aide des commandes au-dessous.

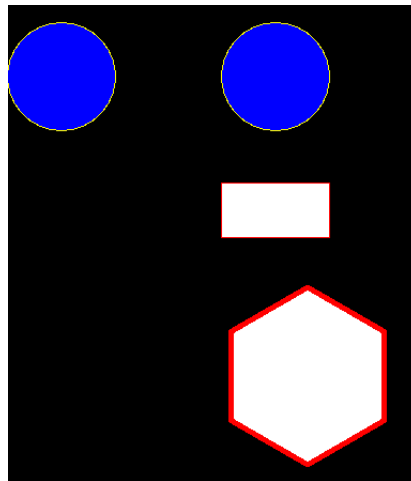
```
> perimetre 2  
157  
>
```

```
> superficie 2  
7850  
>
```

Duplicate

La commande « duplicate » est utilisée pour dupliquer n'importe qu'elle forme déjà créée en indiquant l'identifiant de la forme souhaitée à dupliquer, l'identifiant pour la nouvelle forme et ses positions en x et y.

```
> duplicate 2 4 0 50  
Constructeur par copie : Forme  
Affectation par copie: Cercle  
Constructeur par copie : Forme  
Affectation par copie: Cercle  
Surcharge operateur << Vecteur  
Affectation par copie : Forme  
>
```



List

Avec cette commande on peut lister l'ensemble des formes enregistrées dans la persistante du projet.


```

> list
surcharge <<
-----
FORME_ID :: FORME_TYPE, COUL_FOND, COUL_BORDURE, EPAI_BORD, POSITION, METRIQUE(S)
-----
1 :: rectangle , rouge , vert , 1 , x = 200 y = 200 , longueur : 100 largeur : 50
2 :: cercle , bleu , jaune , 2 , x = 200 y = 50 , rayon : 50
3 :: polygone , blanc , rouge , 3 , x = 200 y = 300 , nb points: 6 rayon : 50
-----
3 element(s)
-----
>

```

Test

La commande « test » permet de tester l'égalité ou l'inégalité de deux formes en utilisant les signes suivant entre les identifiants de ces dernières : ==, < ou >.

```

> test 1 > 2
surcharge operateur >
faux
> test 1 < 2
surcharge operateur <
vrai
> test 1 == 2
surcharge operateur ==
faux
>

```

Dist

Pour calculer la distance entre deux forme, il suffit d'utiliser la commande « dist » en mettant en paramètres les identifiants des deux formes.

```

> dist 1 2
149
>

```

Remove

Enfin si on veut supprimer une forme on fait remove suivi de l'id de la forme sinon si on veut supprimer tout les formes on fait appel à la commande removeall.

```
> remove 1  
>
```

