

TP noté (1h30min)

Aucun document autorisé.

TP noté réalisé sur C5.univ-lyon1.fr.

Un code de base vous est fourni dans la partie centrale de C5. Il contient les structures de base (`vec2` et `color`) ainsi que la plupart des opérateurs dont vous aurez besoin pour réaliser ce TP. Vous pourrez rajouter des opérateurs au besoin. Il contient également le programme principal qui permettra la gestion des menus. Avant de commencer le TP noté, vérifiez que le code fourni compile et s'exécute correctement. Vous devriez avoir une fenêtre graphic qui s'affiche (fig 1.).

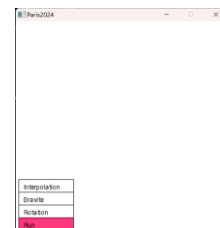


Figure 1 : Application de base.

Vous allez coder 3 fonctionnalités différentes.

- **Interpolation** de la position des anneaux olympiques depuis le centre de la fenêtre vers leur position finale et retour à leur position initiale (fig. 2.)
- Chute des anneaux par **gravité** (fig. 3) et rebonds sur le sol jusqu'à stabilisation.
- **Rotation** des anneaux (fig. 4. Et fig. 5) autour du centre de la fenêtre.

Préambule → Définition de la structure Ring

- 1- Dans un premier temps vous allez définir la structure de données nécessaire à la réalisation des 3 applications. Définissez une structure `Ring` comportant les champs suivants.
 - `RingCenter` : un tableau de `MAXRING` `Particules` représentant les centres des anneaux.
 - `Rayon` : le rayon des anneaux (identique pour les `MAXRING` anneaux).
 - `nbRing` : le nombre d'anneaux utilisés pour l'application.
 - `RingCol` : un tableau de `MAXRING` `color` représentant les couleurs des différents anneaux.
 - `Source` et `Dest` : 2 tableaux de `MAXRING` `Vec2` représentant les positions initiales et finales des anneaux.

Application #1 → Interpolation

- 2- Ecrivez une procédure `InitRingInterpole` qui initialise la structure `Ring`. Cette procédure prendra en paramètres la structure `Ring`, le rayon des anneaux (60 par exemple) et le nombre d'anneaux (ici 5). Dans le sous-programme, les centres des anneaux seront positionnés au centre de la fenêtre de taille `DIMW`, sans vitesse ni force avec une masse de 1.0. Pour les couleurs et la position finale des anneaux, vous utiliserez le code commenté dans le fichier.
- 3- Ecrivez une procédure `Draw` qui à partir de la structure `Ring` permettra d'afficher les anneaux avec la bonne couleur. On commencera par afficher un cercle plein noir de diamètre 3 pixels pour matérialiser le centre de la fenêtre, puis chacun des anneaux. Pour obtenir un cercle plus épais, on dessinera 6 cercles concentriques de rayons allant de r à $r+5$.
- 4- Ecrivez à présent une procédure `RingInterpole` qui à partir d'un paramètre t effectue l'interpolation de la position du centre des anneaux entre les 2 positions `Source` et `Dest`.
- 5- Testez cette première application. Vous devriez obtenir les mêmes résultats que sur la figure 2.

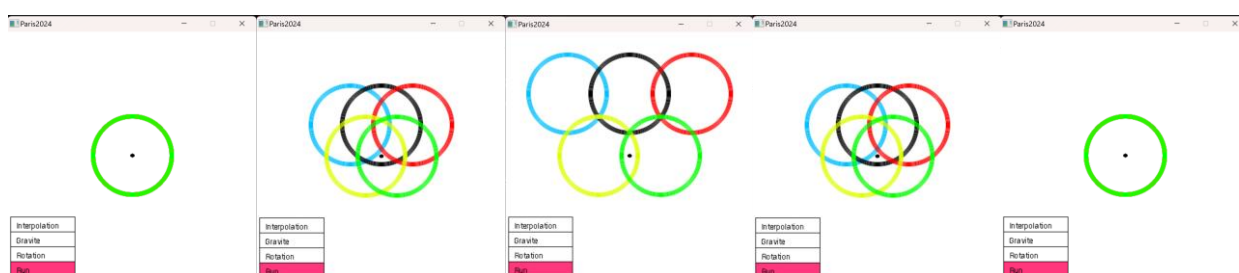


Figure 2 : Interpolation des anneaux olympiques.

Application #2 → Gravité

- 6- Ecrivez la procédure `InitRingGravity` qui initialise la structure `Ring` pour gérer l'application de gravité. Les couleurs des anneaux seront les mêmes que pour l'interpolation, les positions initiales des centres seront les positions `Dest` de l'interpolation, vitesses et forces seront nulles et masse = 1.0.
- 7- Ecrivez les sous-programmes `collision` (uniquement avec le sol) et `updateParticule` des TD / TP. Rappels : ces sous programmes permettent la mise à jour des positions, vitesses, et accélérations des particules en fonction de la gravité. La force de gravité $(0, -9.81)$ sera appliquée à la particule. La vitesse sera mise à jour suivant l'équation : $v(t+1) = v(t) + dt * (f/ m)$ et la position sera mise à jour suivant l'équation : $p(t+1) = p(t) + dt * v(t+1)$.
- 8- Ecrivez enfin la procédure `RingGravity` qui applique la gravité aux anneaux (figure 3).

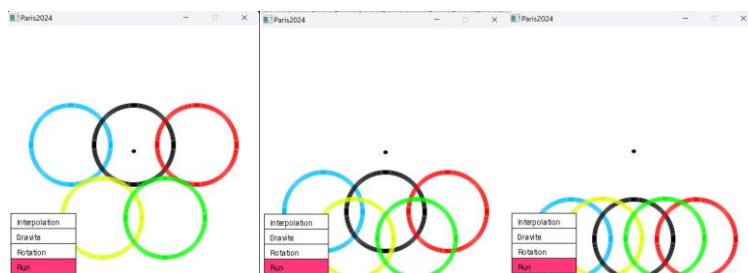


Figure 3 : Gravité.

Application #3 → Rotation

- 9- Ecrivez la procédure `InitRingRotate` qui initialise la structure `Ring` pour permettre la rotation des `nb` anneaux (`nb` passé en paramètre) par rapport au centre. Les centres des anneaux seront à une distance de 100 pixels du centre de la fenêtre. On calculera leur position par rotation par rapport à ce centre. Les couleurs des 5 premiers anneaux pourront être récupérées ou bien choisies aléatoirement comme les couleurs des autres anneaux (figure 4).

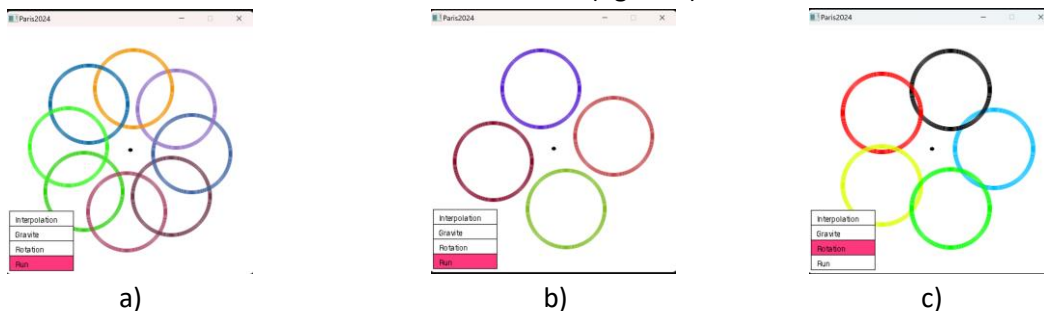


Figure 4 : a) initialisation de 8 anneaux de couleurs aléatoires, b) initialisation de 4 anneaux de couleurs aléatoires, c) initialisation des 5 anneaux avec les couleurs olympiques.

- 10- Ecrivez la procédure `RingRotate` qui fait tourner les anneaux par rapport au centre de 3 degrés à chaque itération (figure 5).

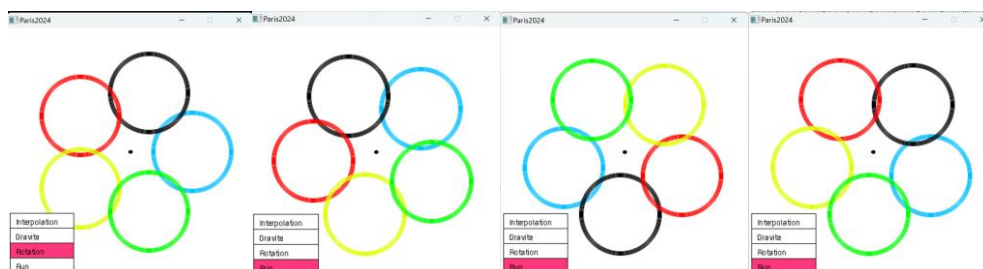


Figure 5 : Rotation des anneaux par rapport au centre de la fenêtre avec 5 anneaux des couleurs prédéterminées.