

Objectifs

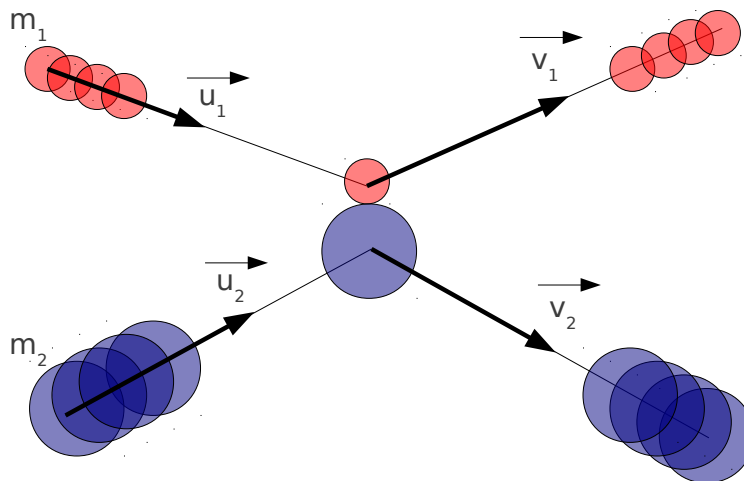
- Gestion des collisions entre la balle et un joueur
- Gestion des collisions entre les joueurs

Il est impératif de valider chaque étape avant de passer à la suivante.

En cas de doute ou de difficulté, consultez votre enseignant.

Introduction

Nous allons nous intéresser à la collision en nous inspirant de la physique pour déterminer le comportement des corps (balle et joueur) au moment de la collision. La position (X_1, Y_1, X_2 et Y_2), la vitesse (u_1 et u_2) et la masse (m_1 et m_2) des corps sont connues, nous allons déterminer le vecteur vitesse des corps (v_1 et v_2) après la collision.

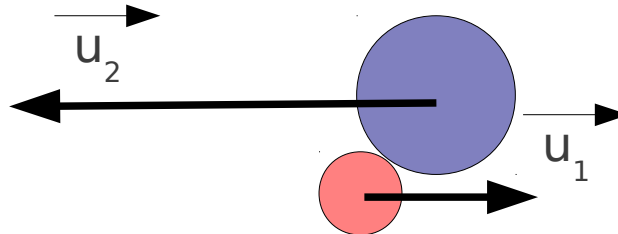


Collisions

Ajouter la fonction suivante dans la bibliothèque ball.h, mais n'écrivez pas le code:

```
// Calcule les vitesses et directions lors d'un choc
void CollisionBallPlayer(BallProperties *Ball, PlayerProperties *Player);
```

Nous allons tester notre fonction sur un cas particulier: les deux corps ont des directions opposées et se heurtent avec un angle de 45° .

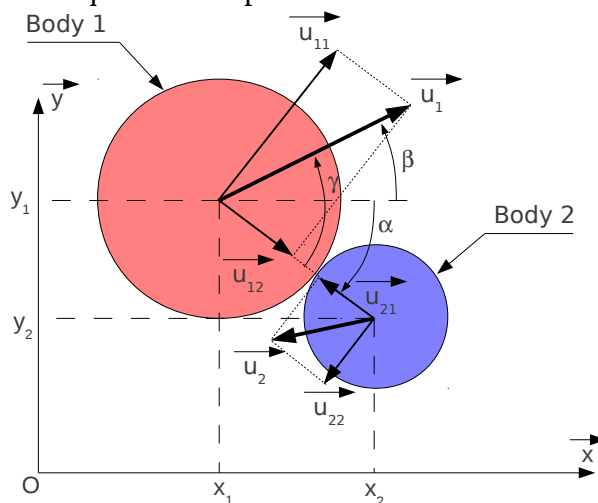


Au début du programme principal insérez temporairement le test décrit ci-dessus:

```
BallProperties B={0,0,0,1};
PlayerProperties P={28.28,28.28,M_PI,5};
CollisionBallPlayer(&B,&P);
return 0;
```

Calcul des angles

Au moment où les corps entrent en contact, la vitesse (u_1) est décomposée en deux vecteurs: le premier est utilisé dans la transmission au second corps (u_{12}) et le second est conservé (u_{21}). En vous aidant du schéma ci-dessous, déterminer les angles α , β et δ en fonction de X_1 , Y_1 , X_2 , Y_2 , u_1 et u_2 . Vous utiliserez la fonction atan2 qui vous simplifiera la tâche.



Implémenter le calcul des angles pour la balle (α_1 , β_1 et δ_1) et pour le joueur (α_2 , β_2 et δ_2). Afficher temporairement les valeurs calculées de Alpha, Beta et Gamma. Ne continuez pas tant que vous n'avez pas obtenu les valeurs suivantes:

α_1	$\pi/4$	0,7853
β_1	0	0
δ_1	$-\pi/4$	-0,7853
α_2	$-3\pi/4$	-2,3561
β_2	π	3,1415
δ_2	$-\pi/4$	5,49779

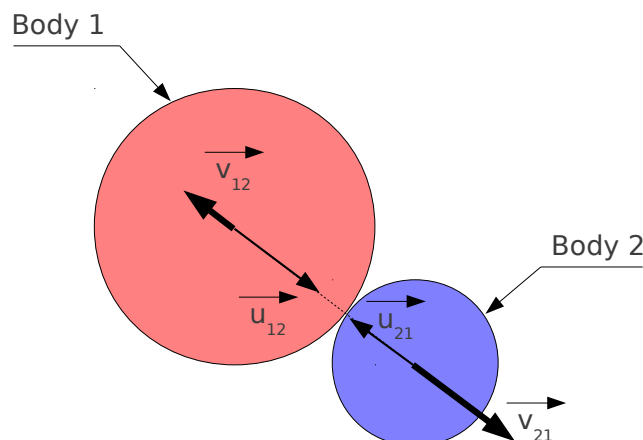
Décomposition des vecteurs

Grâce aux angles calculés, déduire la norme des vecteurs u_{11} , u_{12} , u_{21} et u_{22} . Ne continuez pas tant que vous n'avez pas obtenu les valeurs suivantes:

u_{11}	-0,7071
u_{12}	0,7071
u_{21}	3,5355
u_{22}	-3,5355

Echange d'énergie lors de la collision

Le problème de l'échange d'énergie peut se rapporter au problème d'une collision frontale mettant en œuvre les vecteurs u_{12} et u_{21} :



Une collision élastique est régie par deux principes.

La conservation de l'énergie cinétique:

$$\frac{1}{2} m_1 \|\vec{u}_{12}\|^2 + \frac{1}{2} m_2 \|\vec{u}_{21}\|^2 = \frac{1}{2} m_1 \|\vec{v}_{12}\|^2 + \frac{1}{2} m_2 \|\vec{v}_{21}\|^2$$

La conservation de la quantité de mouvement:

$$m_1 \vec{u}_{12} + m_2 \vec{u}_{21} = m_1 \vec{v}_{12} + m_2 \vec{v}_{21}$$

Ces deux équations forment un système de deux équations à deux inconnues (v_{12} et v_{21}). Résolvez le système, implémentez le résultat. Déclarez les masses des corps dans le fichier tools.h.

```
#define MASS_BALL 1
#define MASS_PLAYER 10
```

Vérifier que vous obtenez bien les valeurs suivantes avant de poursuivre :

v_{12}	-7,00679
v_{21}	-2,76414

Vitesse cartésienne total

La vitesse après collision est donnée par la somme de la vitesse conservée et de la vitesse échangée:

$$\vec{v}_1 = \vec{u}_{11} + \vec{v}_{12} \quad \text{et} \quad \vec{v}_2 = \vec{u}_{22} + \vec{v}_{21}$$

Déterminer les coordonnées des vecteurs v_1 et v_2 . Vérifiez votre résultat avec les données suivantes:

v_1	x	-4,45455
	y	-5,45455
v_2	x	-4,45455
	y	0,54545

Vitesse polaire finale

Connaissant les coordonnées des vecteurs vitesse après collision, déterminez les coordonnées polaire de ces vecteurs et mettre à jour les champs vitesse et angle des structures Ball et Player.

Vous devez obtenir les résultats suivants:

Angle de la balle	-2,25562
Vitesse de la balle	7,04237
Angle du joueur	3,01975
Vitesse du joueur	4,48782

Finalisation

Dans le programme principal, remplacer les lignes de détection de collision par celles-ci:

```
// Vérifie et gère les collisions
if (isCollisionBallPlayer(PropBall, PropPlayer1))
    CollisionBallPlayer(&PropBall,&PropPlayer1);

if (isCollisionBallPlayer(PropBall, PropPlayer2))
    CollisionBallPlayer(&PropBall,&PropPlayer2);
```

Vérifiez que le comportement de la balle lors d'une cas de collision est conforme à la dynamique. Notez qu'il se peut que la balle reste attachée au joueur, ce problème sera rectifié plus tard.

Commentez, indentez et faites valider par l'enseignant.