Objectifs

- Ajouter la balle dans le jeu
- Calculer la dynamique de la balle
- Détecter les collisions

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

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

Ajout de la balle

Créez une nouvelle bibliothèque (fichiers ball.h et ball.cpp). Cette bibliothèque est comparable à celle des joueurs, mais destinée à la balle. Dans le fichier ajouter la ligne suivante :

```
typedef     PlayerProperties    BallProperties;
```

Cette ligne permet de définir un nouveau type (BallProperties) qui sera une copie du type existant PlayerProperties. Les deux types auront les mêmes propriétés et seront compatibles.

En vous inspirant de ce qui a déjà été fait pour les joueurs, ajouter la déclaration et l'affichage de la balle. Ne pas ajouter la dynamique pour l'instant. La position initiale de la balle est fixée au centre du terrain.

Dynamique de la balle

Dans la bibliothèque ball ajoutez la fonction ComputeBallDynamic selon le prototype suivant:

```
// Calcule la dynamique de la balle
BallProperties ComputeBallDynamic(BallProperties prop);
```

Dans la fonction, ajoutez le ralentissement de la balle et le modèle dynamique qui sont identiques à ceux des joueurs, mais n'ajouter pas le bridage de la vitesse. Lorsque la balle dépasse le bord du terrain (et non de la fenêtre) deux actions doivent être mises en place:

- replacer la balle dans le terrain (comme pour les joueurs),
- calculer le nouvel angle de la balle de façon à obtenir un rebond.

Ajoutez 4 constantes symboliques (LIMIT_BALL_BOTTOM, LIMIT_BALL_TOP, LIMIT_BALL_LEFT et LIMIT_BALL_RIGHT) pour définir les limites du terrain. Pour l'instant, on ne tiendra pas compte des buts.

Commentez le ralentissement naturel de la balle et vérifier que les rebonds fonctionnent sur les quatre bords avec l'initialisation suivante de la balle :

```
BallProperties PropBall={ WINDOW_WIDTH/2. , WINDOW_HEIGHT/2. , 0.5 , 3.};
```

Boucle principale

Afin de détecter et gérer les collisions, nous allons restructurer la boucle principale de la façon suivante :

1. Lecture des entrées (clavier et joystick)
2. Calcul de la dynamique (pour les joueurs et la balle)
3. Rafraîchissement de l'affichage.

Espacez votre boucle en trois partie distinctes et déplacer les 3 fonctions liées à la dynamique au milieu. Vérifier que l'application fonctionne toujours.

Détection des collisions

Dans la bibliothèque ball, ajoutez une fonction qui retourne la distance entre la balle et un joueur. Le prototype vous est donné ci-dessous:

```
// Calcule la distance entre la balle et un joueur
double  DistanceBallPlayer (BallProperties Ball, PlayerProperties Player);
```

Ajoutez dans la même bibliothèque une fonction qui retourne 1 si la balle et le joueur sont en collision, et 0 sinon:

```
// Teste s'il y a une collision entre la balle et le joueur
unsigned char isCollisionBallPlayer(BallProperties Ball, PlayerProperties
Player);
```

Vous utiliserez bien évidement la fonction `DistanceBallPlayer` lors de la vérification des collisions. Les diamètres des éléments de jeux vous sont fournis:

```
#define         DIAM_BALL               32
#define         DIAM_PLAYER             48
```

Pour tester vos fonctions, vous allez implémenter la règle suivante après le calcul de la dynamique: si une collision a lieu entre la balle et un joueur:

- l'orientation de la balle devient identique à celle du joueur,
- la vitesse de la balle est égale à 150% de la vitesse du joueur.

Vérifiez que les collisions sont bien détectées puis supprimer la partie du programme principal qui a servi au test.

Complétez la bibliothèque `player` avec deux fonctions similaires qui permettront de mesurer la distance et de détecter les collisions entre les deux joueurs. Testez vos fonctions par la méthode de votre choix.

Commentez, indentez et faites valider par l'enseignant.