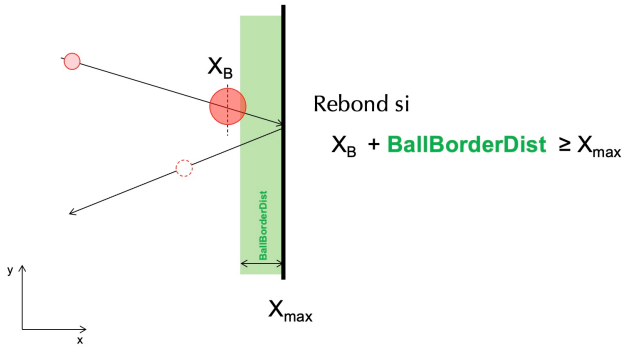
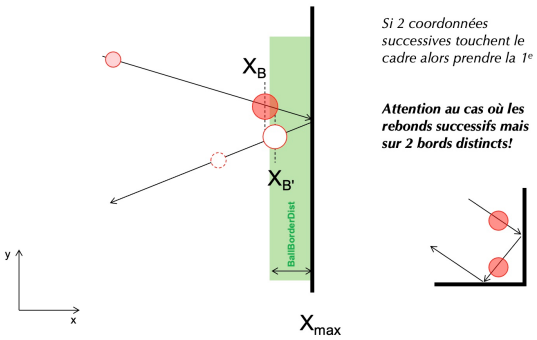
Projet programmation 2025 – Ex. 3

But

Ecrire la fonction **GetTouchIdx**() qui retourne les indices correspondants aux chocs entre la boule et le cadre du billard, i.e. quand la distance boule-cadre < BallBorderDist avec BallBorderDist= 9.

```
function [IdxTouch]=GetTouchIdx(X,Y,Xmin, Xmax, Ymin, Ymax, BallBorderDist)
```





- La boule peut ne jamais toucher le cadre, dans ce cas retourner [].
- Omettre l'indice **1** si celui-ci touche le cadre.
- Il se peut que plusieurs indices successifs satisfassent la condition < BallBorderDist, dans ce cas garder uniquement le 1e indice. Attention aux cas ou deux bords différents sont touchés successivement.
- *Hint* : traiter les bords séparément, puis les mettre ensemble.

Ecrire la fonction **GetBallMoveOrder**() qui retourne l'indice de la 1e boule ayant bougé, idem 2e et 3e boules ayant bougé. La boule *Red* à l'indice 1, *Yellow* : 2, *White* : 3. La fonction retourne également le nombre de boules qui ont bougé. Si deux boules ont le même indice, vous devez prendre celle qui a la plus grande distance initiale parcourue, i.e. 1e segment.

```
function [FirstBall,SecondBall,LastBall, NbBallsMoved]
                = GetBallMoveOrder(Xr, Yr,  Xy, Yy,  Xw, Yw,  MoveDistPx)
```

- *Hint* : donner un indice impossible (ex. length(X)+1) aux boules qui ne bougent pas.

Utiliser la fonction **GetFirstMoveIdx** () de la semaine passée, vous devrez la modifier pour qu'elle retourne également le vecteur des distances parcourues, i.e. segments.

```
function [FirstMoveIdx, MoveDist]=GetFirstMoveIdx(X,Y, MoveDistPx)
```

Fonctions utiles

- `diff()`
- `find()`
- `unique()`
- `length()`