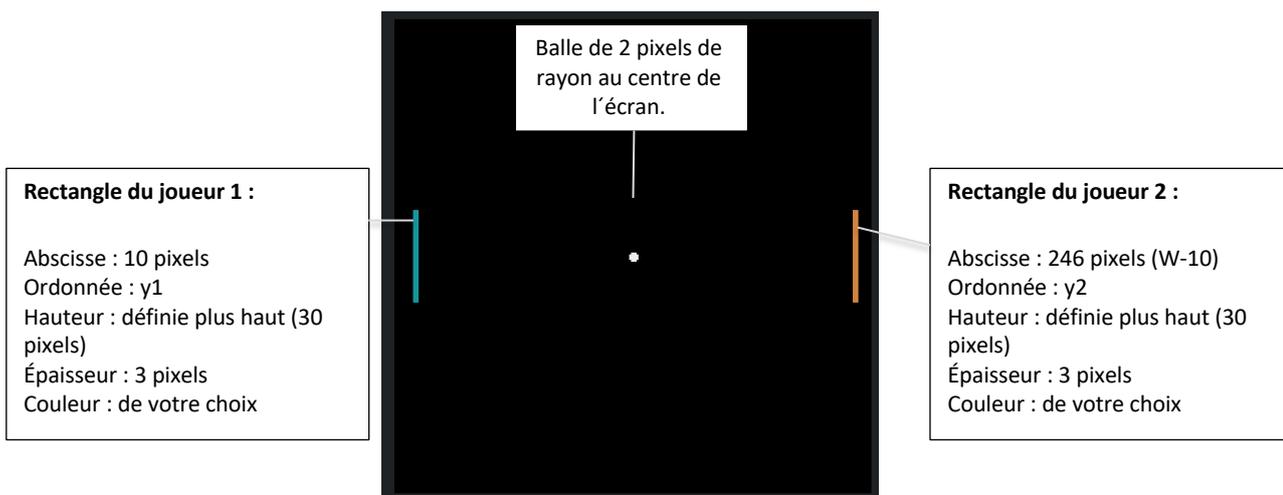




## CRÉATION D'UN JEU: DEUXIÈME PARTIE



La scène est prête. Il faut maintenant mettre le jeu en action !

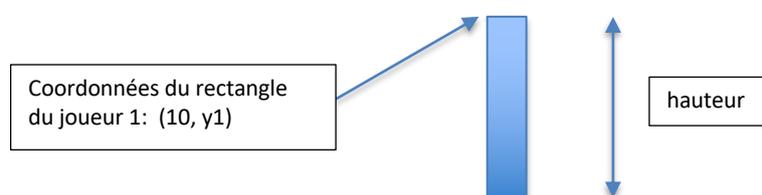
### 1. DÉPLACEMENT DES JOUEURS

Nous avons vu qu'il est possible de déclencher des *événements* clavier dans la fonction `update`.

👉 *Notre mission* : Le **joueur 2** d'ordonnée `y2` doit se déplacer verticalement en pressant les touches UP et DOWN, et le **joueur 1** d'ordonnée `y1` avec les touches W et S.

👉 *En option* : On peut empêcher le rectangle des joueurs de sortir de la scène.

Pour cela, nous devons savoir une chose importante : les coordonnées d'un rectangle repèrent son coin supérieur gauche.



## 2. DÉPLACEMENTS DE LA BALLE

Nous avons défini trois variables pour la balle : les coordonnées de son centre (`x_balle` et `y_balle`), sa vitesse en x (`vx`) et sa vitesse en y (`vy`). On rappelle que son rayon fait 2 pixels. Nous allons coder ses mouvements dans la fonction `update`.

☞ *Mission 1* : Lancer la balle ! A chaque appel de la fonction **update**, ses coordonnées en x et en y doivent respectivement augmenter (ou diminuer, c'est comme on veut) de `vx` et `vy`.

Seul petit problème, la balle ne doit pas sortir de l'écran.

☞ *Mission 2* : Il faut maintenant coder les rebonds de la balle sur le haut et le bas de la scène et aussi bien sûr, sur le rectangle des joueurs. Pour cela, nous allons utiliser des conditions, avec ces indications importantes :

- Lorsque la balle touche un mur, il faut inverser le signe de `vy`.
- Si elle touche un joueur, il faut inverser le signe de `vx`.

☞ *En option* : Pour ajouter un peu de difficulté, on peut augmenter légèrement la vitesse de la balle (`vx`) à chaque rebond sur un joueur, ou diminuer la taille de chaque rectangle...

☞ *Mission 3* : Si un joueur a perdu, on remet la balle au centre et on la relance dans l'autre sens.

- Lorsque la balle touche un mur, il faut inverser le signe de `vy`.
- Si elle touche un joueur, il faut inverser le signe de `vx`.

## 3. AMÉLIORATIONS POSSIBLES

☞ *Amélioration 1* : On peut créer deux nouvelles variables (avant la fonction `update`) pour garder en mémoire le score des joueurs. N'oubliez pas de les ajouter à la liste des variables globales, au début de la fonction `update`.

- On affiche le score de chaque joueur. La documentation de `Pyxel` indique :

```
pyxel.text(x, y, s, col)  
Draw a string s in color col at (x, y)
```

Cette instruction doit apparaître dans la fonction `draw`. Attention, le score doit être converti en chaîne de caractère avec la fonction `str()`.

- Lorsqu'un joueur perd, l'autre gagne un point. Le premier arrivé à 10 gagne et on quitte le jeu avec l'instruction **`pyxel.quit()`**.

☞ *Amélioration 2* : Ajouter un peu de hasard dans la direction initiale de la balle. Pour cela, on peut choisir une valeur au hasard pour `vy`. La documentation de `Pyxel` nous informe :

```
pyxel.rndf(a, b)  
Return a random floating-point number greater than or equal to  
a and less than or equal to b.
```

☞ *Pourquoi ne pas ajouter du son ?* : Rendez-vous sur la documentation de `Pyxel` ...