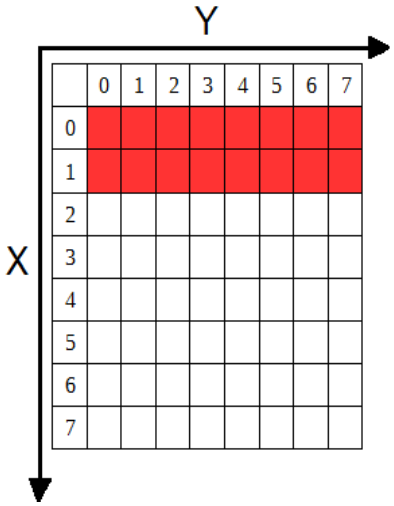
Le jeu du casse briques

Nous souhaitons réaliser le jeu du casse briques sur notre matrice de LED. Pour cela, nous allons décomposer le problème en plusieurs étapes.

Fabriquez un nouveau projet de type ***application console*** en langage **C** nommé ***Briques***.

Étape n°1 : Afficher les briques

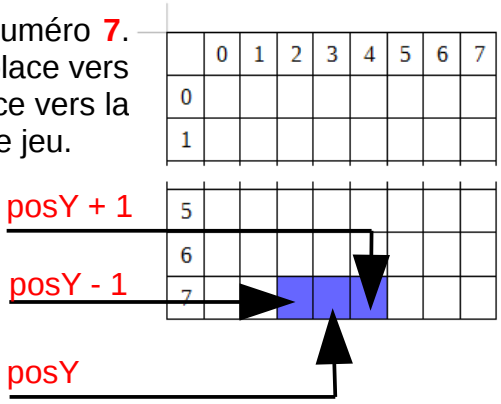
Les briques sont constituées de deux lignes de LED rouges, la ligne 0 et 1, comme le montre la figure ci-dessous :



Questions à se poser :
- Comment réaliser ces deux lignes ?
- Quel type de boucle faut-il utiliser ?

Étape n°2 : Afficher la raquette et la faire se déplacer

La raquette occupe 3 LED bleues sur la ligne numéro **7**. Le bouton gauche du joystick **KEY_LEFT** la déplace vers la gauche, le bouton droit **KEY_RIGHT** la déplace vers la droite. Le bouton central **KEY_ENTER** quittera le jeu.



- Il faut allumer les 3 LED
- Animer la raquette avec les boutons :
  Il faut effacer la LED qui se trouve en posY+1 pour KEY_LEFT et posY-1 pour KEY_RIGHT.
  Puis faire posY ← posY -1 dans le 1er cas et posY ← posY +1 dans le 2éme.

Attention : posY doit rester entre 1 et 6 sinon la raquette sort du jeu.

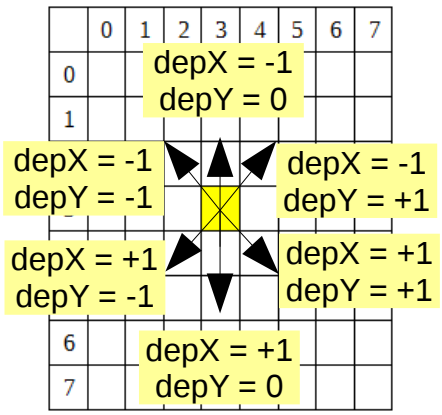
Étape n°3 : Animer la balle.

Pour la balle, nous pouvons choisir d'utiliser une LED jaune, dont les coordonnées sont balX et balY. Pour assurer son déplacement sur la matrice, il faut allumer la LED en NOIR en balX et balY pour effacer la balle précédente puis recalculer balX et balY, les nouvelles coordonnées, à partir des formules ci-dessous et allumer la balle en JAUNE :

balX ← ***balX + depX*** (depX déplacement en X)

balY ← ***balY + depY*** (depY déplacement en Y)

Les valeurs de depX et depY représentent le déplacement que doit faire la balle (-1, 0 ou 1) sur chaque axe. Elles sont données par la figure suivante.



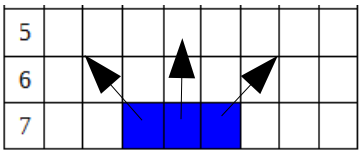
Le rebond ne peut se faire que sur la raquette, ou sur le coté gauche **posY = 0**, ou sur le coté droit **posY = 7**, ou sur le mur de briques ou encore sur le haut **posX = 0** si un trou a été réalisé dans le mur.

Pour savoir si la balle est sur le mur ou sur la raquette, il est possible avec la fonction ***uint16_t ObtenirPixel(int X, int Y)*** de la librairie ***senseHat*** de connaître la couleur d'un pixel.

La fonction prend en paramètre les coordonnées du point dont on veut connaître la couleur. Elle retourne sous la forme d'un entier non signé sur 16 bits cette couleur. De 0X0000 pour NOIR … ROUGE, BLEU, jusqu'à 0xFFFF pour BLANC.

Pour déterminer le type de rebond, il y a plusieurs cas de figure :

- **Sur la raquette** : le déplacement en X est toujours négatif. Le déplacement en Y est donnée par la figure.



- **Sur les murs** : côté droit et côté gauche, on inverse le sens du déplacement en Y (depY ← - depY). Le déplacement en X ne change pas.

- **Sur les briques ou sur le mur du haut** : on propose de renvoyer aléatoirement la balle avec un déplacement en Y égale soit à -1, soit à 0, soit à 1. Le déplacement en X est toujours positif.

  Pour fournir un nombre entier compris entre [-1 et 1] on peut écrire :

  depY ← (rand() % 3) – 1

  donne un nombre entier entre [0 et 2]

Pour finir :

Le déplacement de la raquette et l'animation de la balle doivent se faire dans la même boucle. Elle s'arrête lorsque l'utilisateur appuie sur le bouton central du joystick, KEY_ENTER ou que le joueur n'a pas rattrapé la balle avec la raquette. Il est nécessaire de ralentir le traitement avec une temporisation d'environ 200 ms, sinon la balle va trop vite.

Et maintenant à vous de jouer …