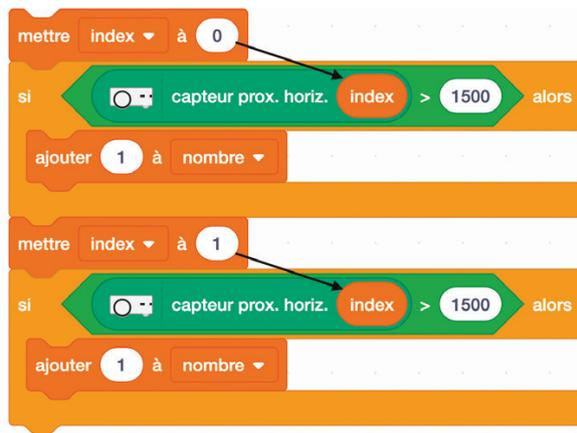


Etape 2 : D'un point de vue programmation, il n'est pas satisfaisant de répéter 5 fois le même test avec juste la référence du capteur qui change. Heureusement, on peut remplacer le nom du capteur par un nombre entier appelé **index** avec la correspondance : 0 : gauche ; 1 : centre gauche ; 2 : centre ; 3 : centre droite ; 4 : droite

Dans un premier temps, gardons les 5 tests en effectuant le remplacement.

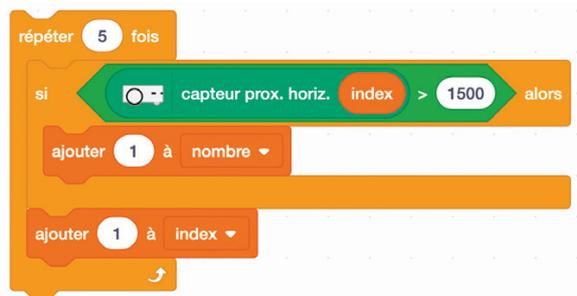
La figure ci-contre montre un fragment du code avec 2 tests enchaînés :

Modifier le script et le faire fonctionner.



Etape 3 : On remplace le bloc **< mettre [index] à (...) >** par **< ajouter (1) à [index] >**.

Modifier le script et le faire fonctionner.



Etape 4 : En lisant le script, on observe que l'on répète strictement 5 fois les mêmes blocs. D'où l'idée d'utiliser une boucle de répétition automatique.

Le bloc **< dire le nombre >**

Une fois le nombre **nombre** calculé, on crée un bloc **< dire le nombre (nombre) >** qui prend en entrée **nombre**. Dans ce bloc, on répète **nombre** fois la séquence :

- Allumer les leds rvb.
- Jouer une note pendant un temps court.
- Attendre le même temps.
- Éteindre les leds.
- Attendre un autre instant pour que les bips ne soient pas trop proches.

Le programme complet

Il est constitué simplement :

- Du bloc chapeau pour démarrer le script.
- Du bloc **< compter >**.
- Du bloc **< dire le nombre >**.

Enrichir le programme

- Dans le programme précédent, si il n'y a pas d'objet, **nombre** = 0 donc il n'y pas de son. On peut ajouter dans ce cas un bip différent (un son système) et une couleur différente.
- **[plus difficile]** Une autre idée intéressante est de déclencher un comptage automatique si on ajoute ou enlève un objet. Pour cela, dans le programme principal, on ajoute une variable **ancien_nombre** qui vaut **nombre** au début. Puis on ajoute une boucle infinie qui compte en permanence. (ajouter un bloc **< attendre (0,5) secondes >** pour éviter des blocages de Scratch). Quand on ajoute ou enlève un objet, **nombre** devient différent de **ancien_nombre**. Donc mettre dans la boucle un test **< Si ... >** qui exécute **< dire le nombre >** si **nombre** est différent de **ancien_nombre**. Puis remettre **ancien_nombre** à **nombre** pour éviter d'exécuter à nouveau **< dire le nombre >**.