

Instruments électriques

Chapitre 1 : Programmer avec des variables

Chapitre 2 : Une boîte à musique électrique

Chapitre 3 : Faire une guitare

Chapitre 4 : Une meilleure guitare

Prénom : _____ Nom : _____ Classe : _____

Les programmes nécessaires à la réalisation des robots sont disponibles en téléchargement sur le site www.ecolerobots.com.

Toutes les boîtes et les pièces détachées sont aussi disponibles sur le site www.ecolerobots.com.

Instruments électriques

Montage, programmation, robotique
Ecole Robots – Cursus Éducation Nationale

Sommaire

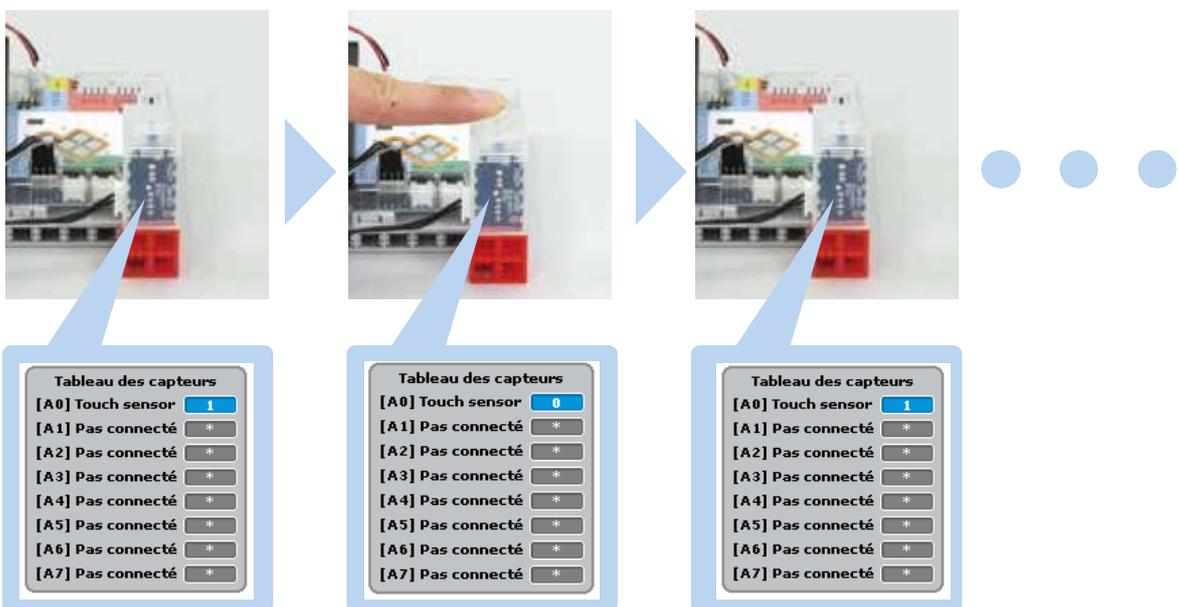
Chapitre 1 : Programmer avec des variables	1
1. Comment utiliser des variables ?	2
2. Comment fonctionnera ton feu ?	5
3. Faire un feu qui s'active au bruit	6
4. Définir tes ports	8
5. Compter les claquements de mains	8
6. Programmer ton feu	11
Chapitre 2 : Une boîte à musique électrique	13
1. Ajouter un buzzer	14
2. Définir tes ports	15
3. Choisir la longueur de la note	16
4. Brille, brille, petite étoile	17
5. La touche finale	19
Chapitre 3 : Faire une guitare	23
1. Construire une guitare	24
2. Définir tes ports	26
3. Jouer de ton instrument	26
4. Notes et valeurs du photorélecteur IR	28
5. Faire des conditions pour les notes	30
6. Jouer plus de notes	35
Chapitre 4 : Une meilleure guitare	38
1. Définir les ports	39
2. Notes et nombres	39
3. Programmer deux octaves	42
4. Un petit concert	47

Chapitre 1

Programmer avec des variables

As-tu déjà voulu compter à l'intérieur d'un programme ? Pourquoi ne pas compter, par exemple, le nombre de fois que tu appuies sur ton capteur de pression, par exemple ? Bien que le capteur de pression puisse indiquer s'il est pressé ou non, **il ne peut pas compter le nombre exact de fois qu'il a été pressé.**

Exemple Comment fonctionne un capteur de pression ?

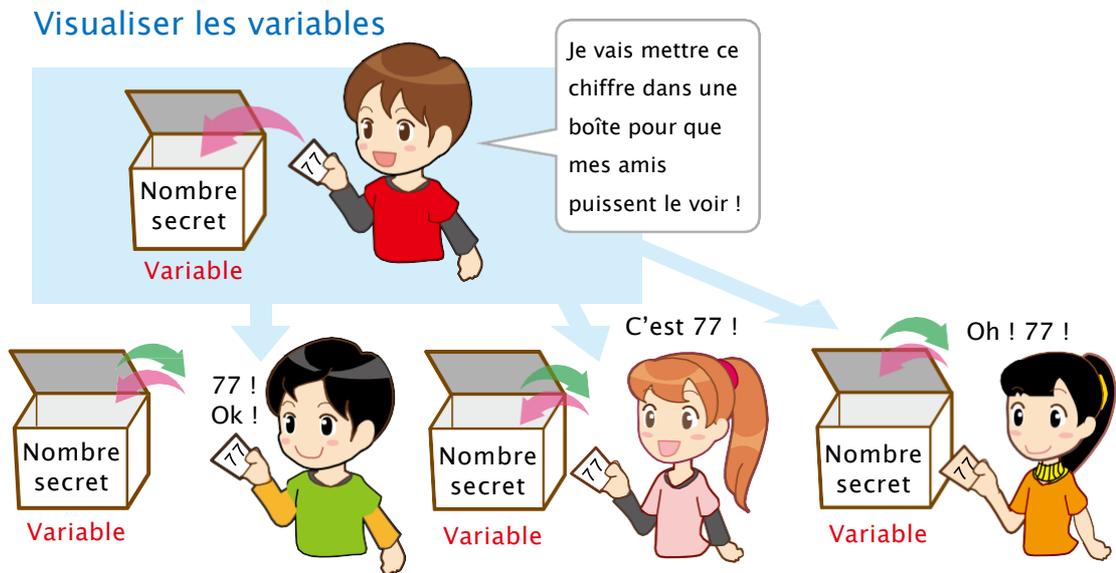


On peut seulement savoir s'il est pressé ou non !

C'est là qu'interviennent les **variables** qui s'avèrent très utiles dans ce genre de situations. Les variables sont utilisées pour stocker des nombres pendant que le programme s'exécute. On peut donc l'utiliser pour enregistrer un cumul. Nous allons apprendre à utiliser des variables avec un robot qui s'allume lorsqu'on frappe dans nos mains.

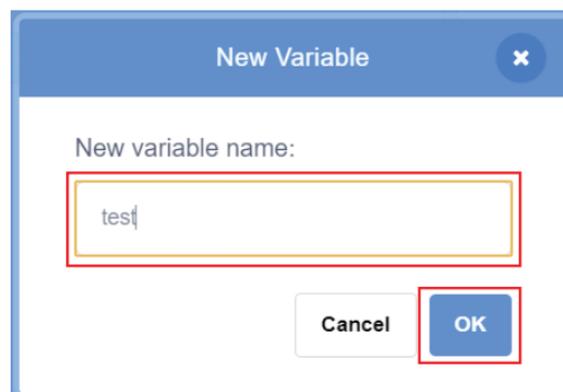
1 Comment utiliser des variables ?

Les variables sont comme des boîtes dans lesquelles on stocke des nombres. Tu peux donner un nom à cette boîte, retirer les nombres qui ont été stockés et les vérifier pratiquement de n'importe où !

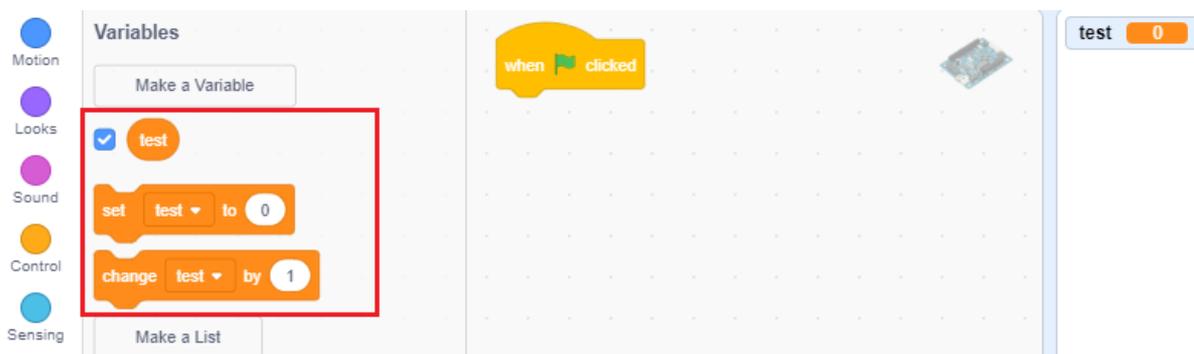


Pour faire des variables, utilise la palette « Variables ». Apprenons à en faire une !

1 Cliquez sur **Variables**, puis sur **Make a Variable** (*Créer une variable*).
Nomme la variable « test » et clique sur OK.

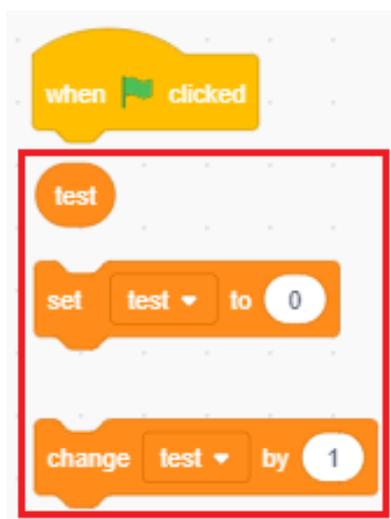


2 Jetons un œil sur les blocs qu'on peut utiliser avec notre nouvelle variable. On peut également regarder `test 0` à droite de l'écran qui indique le nombre stocké dans la variable.



	Ce bloc récupère le nombre dans la boîte et l'utilise. Clique dessus pour voir quel nombre est stocké à l'intérieur.
	Ce bloc stocke dans la variable le nombre que tu as tapé.
	Ce bloc ajoute le nombre que tu as tapé au nombre stocké dans la variable, puis stocke ce nouveau nombre.

3 Glisse chacun des blocs dans le champ de script.



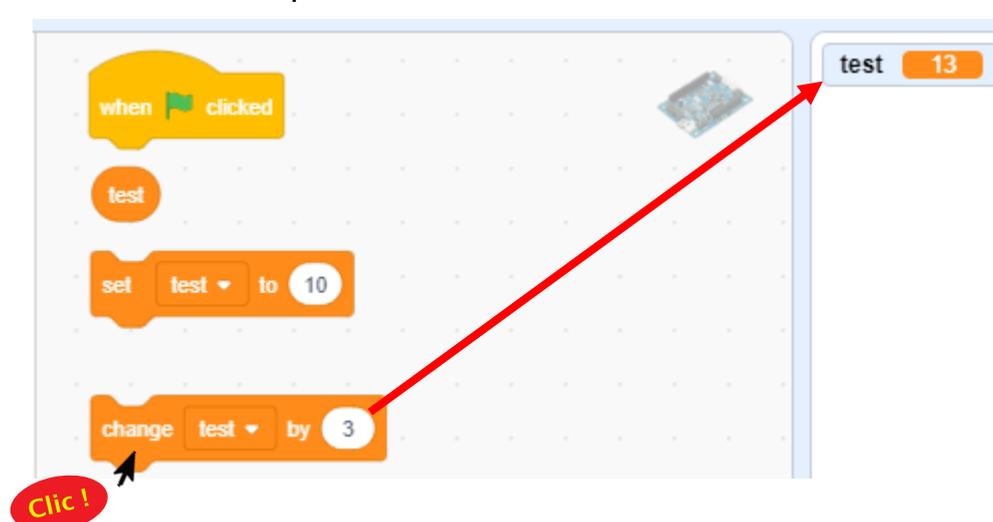
- 4 Clique sur **test** pour voir quel nombre est stocké dans la variable. Chaque variable que tu crées stocke au départ le chiffre 0.



- 5 Change le nombre dans le bloc **set test to 0** par 10 et clique dessus. Cela change le nombre dans ta variable par 10.

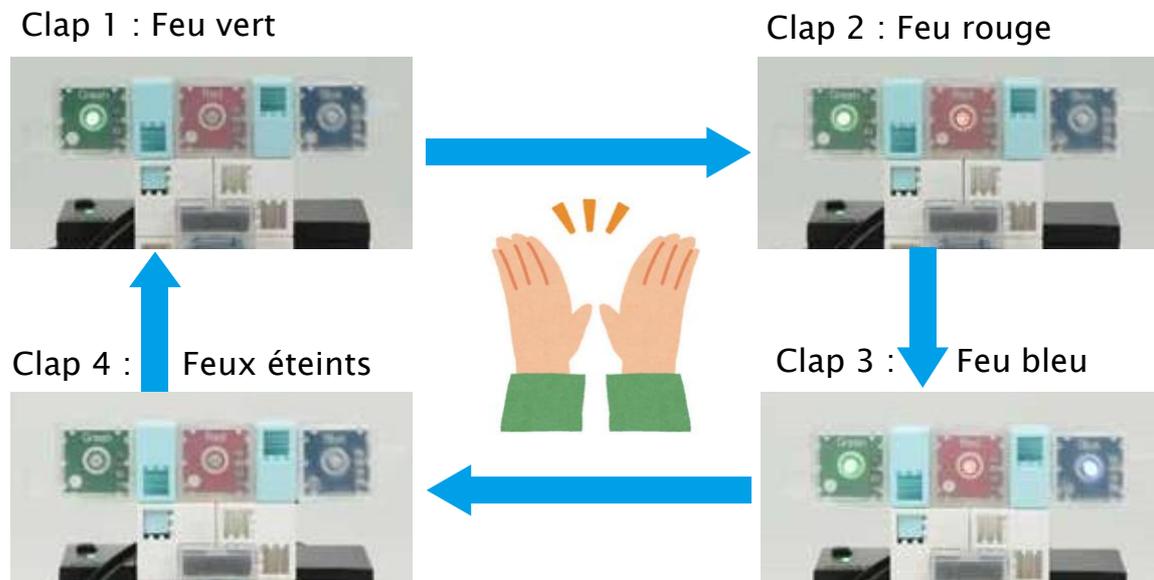


- 6 Change le nombre dans le bloc **change test by 1** par 3 et clique dessus. Cela ajoutera 3 au nombre dans ta variable pour faire 13 et augmentera le nombre de 3 à chaque clic.

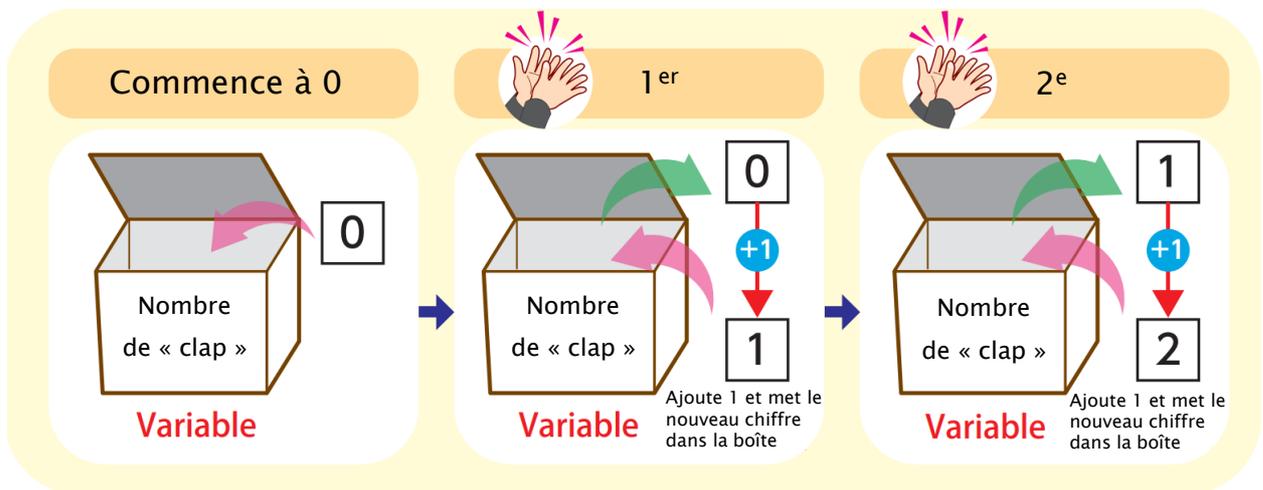


2 Comment fonctionnera ton feu ?

Nous allons utiliser les variables pour faire un feu qui s'allume de différentes façons en fonction du nombre de fois que tu tapes dans tes mains.



Mais le capteur de son ne peut pas compter de lui-même ! Tu dois utiliser une variable pour stocker le nombre de sons que ton capteur de son détecte, ce qui permettra à ta lumière de détecter l'ordre des sons.



3 Faire un feu qui s'active au bruit

Utilise un capteur de son et des LED pour construire ton feu.

Tu auras besoin de...



Studuino x 1



Câble USB x 1



Batterie x 1



Câble de connexion de capteur (S) x 4 (3 fils, 15 cm)



LED (verte) x 1



LED (rouge) x 1



LED (bleue) x 1



Capteur de son x 1

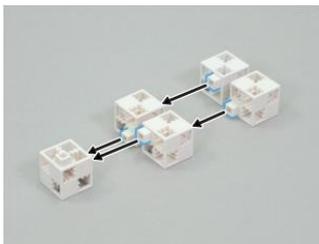


Cube basique (blanc) x 5



Demi-cube C (bleu) x 3

1 Connecte ces blocs.



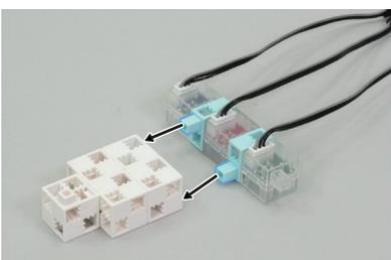
2 Ajoute ce bloc au capteur de son.



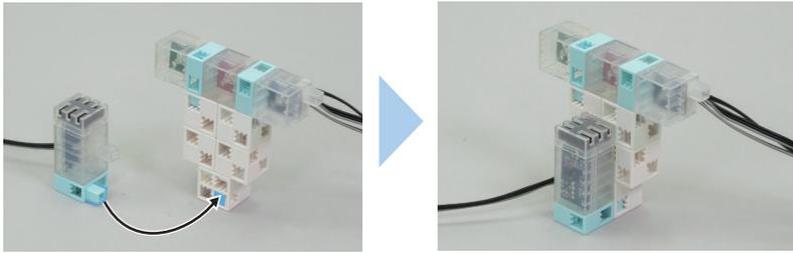
3 Cale les blocs ci-dessous entre les LED.



4 Ajoute l'élément 3 à l'élément 1.

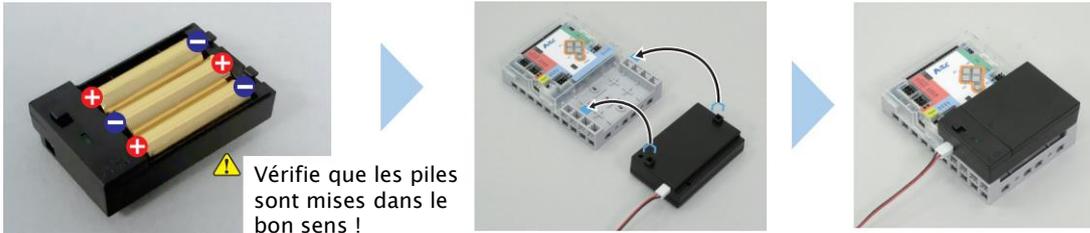


5 Ajoute l'élément **4** à l'élément **2**.

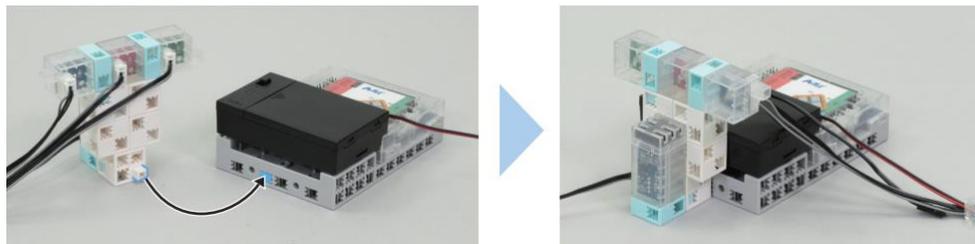


6 Mets dans la batterie 3 piles (AA/LR6).

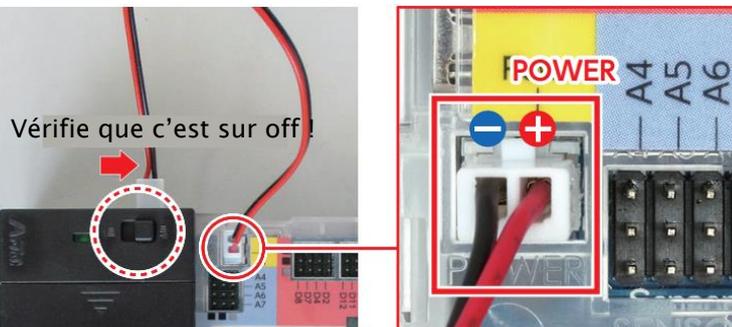
Puis ajoute la batterie à ton Studuino.



7 Ajoute l'élément **5** à l'élément **6**.

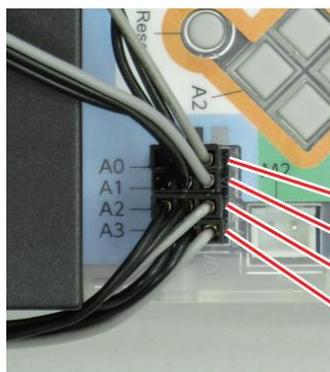


8 Branche ta batterie sur le connecteur **Power** de ton Studuino.



⚠ Vérifie que c'est bien inséré !

9 Branche tes câbles sur les connecteurs comme ci-dessous.



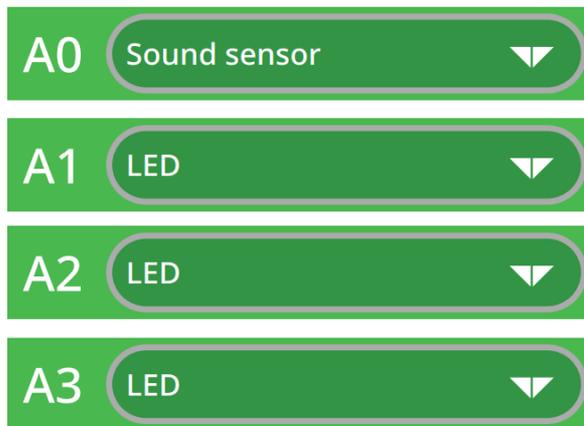
A0: Capteur de son
A1 : LED verte
A2: LED rouge
A3: LED bleue

10 Fini !



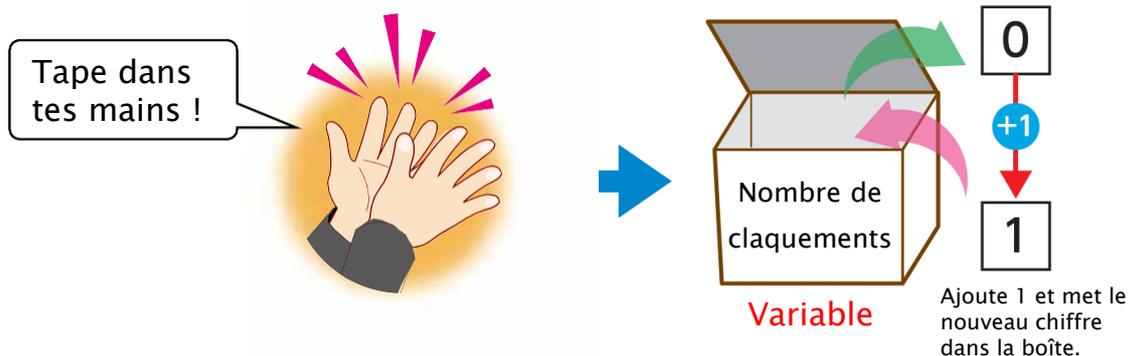
4 Définir tes ports

Sélectionne **Sound Sensor** pour **A0** et **LED** de **A1** à **A3** !



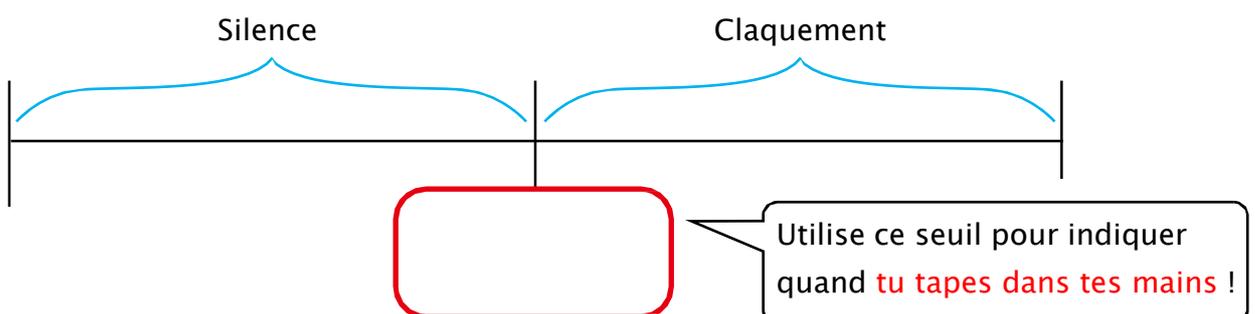
5 Compter les claquements de mains

Fais un programme qui ajoute 1 à la variable à chaque fois que le capteur de son détecte un claquement.

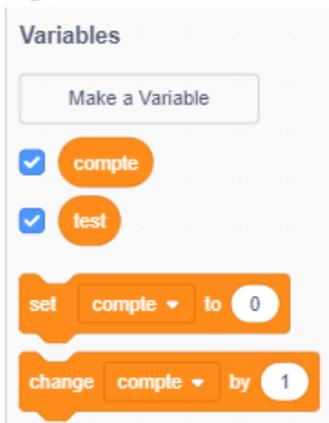


1 Jette un œil aux valeurs de ton capteur de son et choisit un seuil.

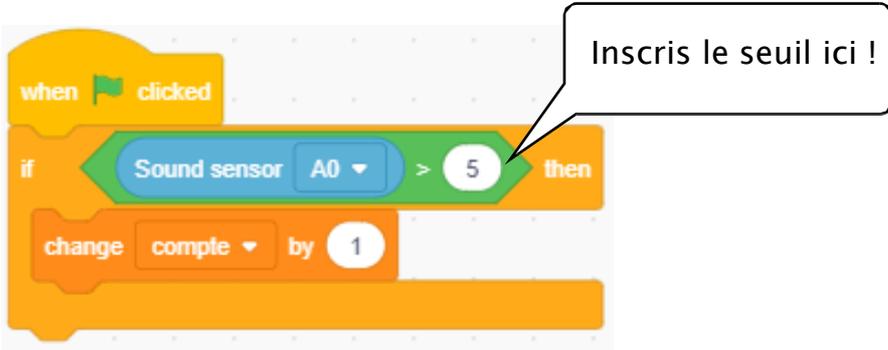
État	Quand tu tapes...	Quand tu ne tapes pas...
Valeur du capteur		



2 Crée une variable appelée **compte**.

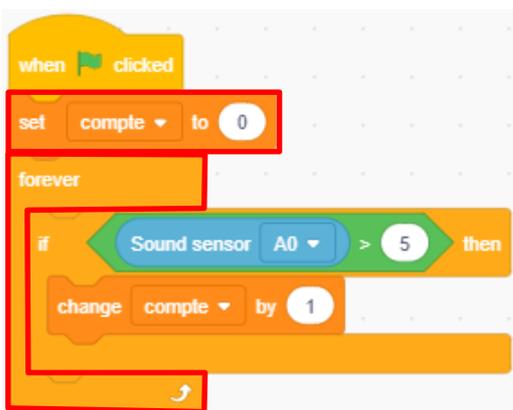


3 Fais en sorte que ton programme ajoute 1 à la variable **compte** à chaque fois que tu frappes dans tes mains.

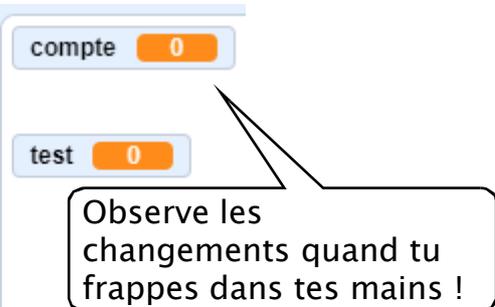
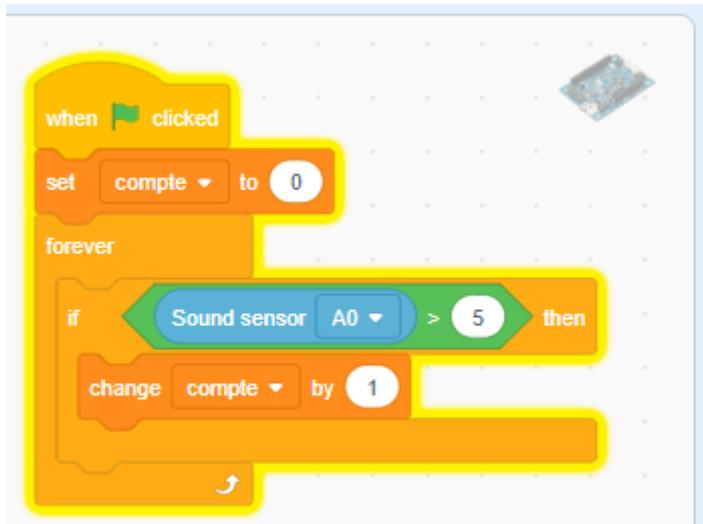


4 Assure-toi que ton programme continuera de vérifier le capteur de son en l'insérant dans un bloc **forever**.

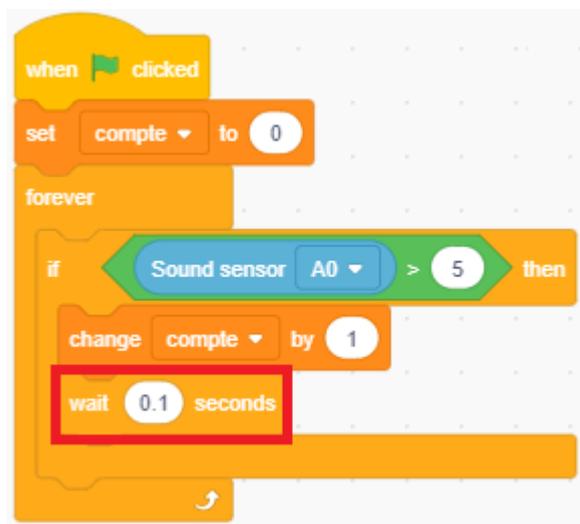
Fais démarrer la variable **compte** à 0 en ajoutant un bloc **set count to 0** au début de ton programme.



5 Après avoir transféré ce programme, tu trouveras que ça ne fonctionne pas très bien. C'est parce que ton **Studuino exécute les programmes très rapidement et taper dans tes mains une fois fera s'exécuter les blocs à l'intérieur de**  **en continu.** Ouvre le mode test et vois par toi-même



Ajoute un bloc  ici. Ajouter ce bloc met en pause le programme pendant un temps et fait s'exécuter les blocs à l'intérieur de  à chaque fois que tu frappes dans tes mains !



6 Voyons maintenant si ça fonctionne ! Lance le mode test pour vérifier si 1 est ajouté à la variable compte à chaque fois que tu frappes dans tes mains.

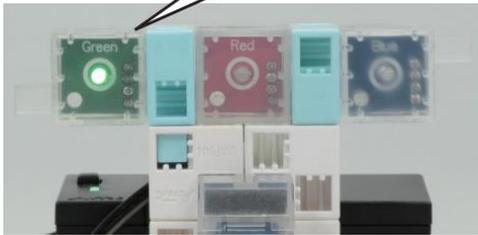
6 Programmer ton feu

Ajoute à ton programme qui compte les applaudissements une partie qui allume différents nombres de lumières en réponse.

1 Programme ce qui doit arriver au premier applaudissement.

Clap 1

Feu vert

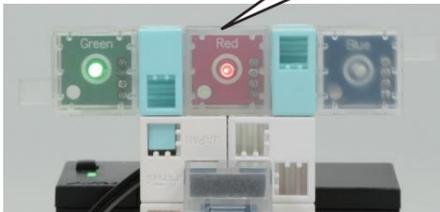


```
when clicked
  set compte to 0
  forever
    if Sound sensor A0 > 5 then
      change compte by 1
      wait 0.1 seconds
      if compte = 1 then
        turn LED A1 ON
```

2 Programme ce qui doit arriver aux 2e et 3e applaudissements.

Clap 2

Feu rouge



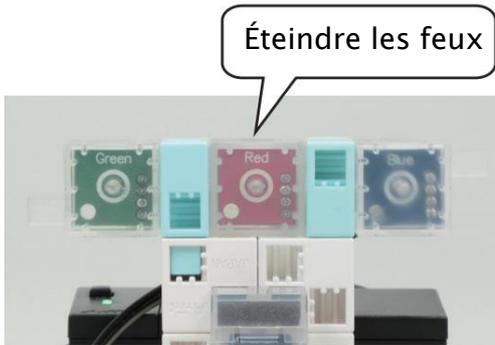
Clap 3

Feu bleu



```
when clicked
  set compte to 0
  forever
    if Sound sensor A0 > 5 then
      change compte by 1
      wait 0.1 seconds
      if compte = 1 then
        turn LED A1 ON
      if compte = 2 then
        turn LED A2 ON
      if compte = 3 then
        turn LED A3 ON
```

3 Programme ce qui doit arriver au quatrième applaudissement ! Dès que le 4e applaudissement fait recommencer le programme à 0, remets la variable compte à 0.



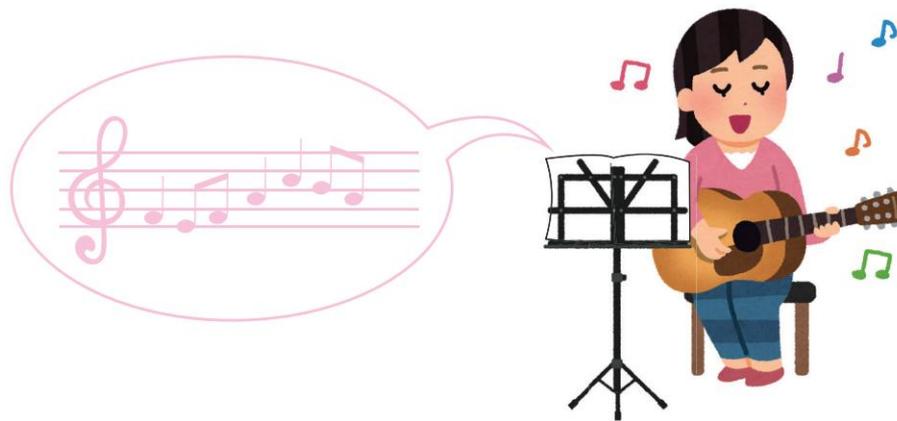
```
when clicked
  set compte to 0
  forever
    if Sound sensor A0 > 5 then
      change compte by 1
      wait 0.1 seconds
    if compte = 1 then
      turn LED A1 ON
    if compte = 2 then
      turn LED A2 ON
    if compte = 3 then
      turn LED A3 ON
    if compte = 4 then
      turn LED A1 OFF
      turn LED A2 OFF
      turn LED A3 OFF
      set compte to 0
```

4 À présent, transfère ton programme et vois si ça fonctionne.

Chapitre 2

Une boîte à musique électrique

As-tu déjà remarqué que beaucoup de musiciens regardent une partition de musique quand ils jouent ?



Les partitions n'indiquent pas seulement quelles notes jouer, mais combien de temps les jouer pour que la chanson sonne bien ! La longueur de ces notes est déterminée en multipliant ou en divisant la longueur d'une note standard.

	 Diviser par 4 !
	 Diviser par 2 !
	 La note standard !
	 Multiplier par 2 !
	 Multiplier par 4 !

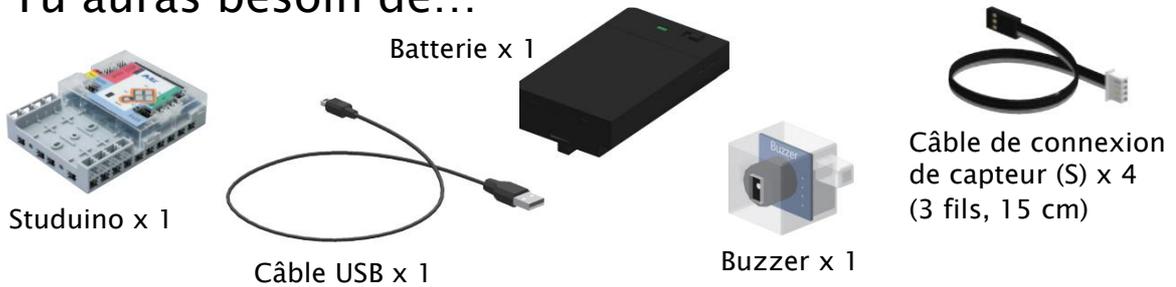
Tu verras qu'un bon nombre de notes dans la chanson ont la même longueur. Tu peux utiliser les variables que tu as vues au chapitre 1 pour régler facilement la longueur de ces notes !

Utilise les variables pour programmer une boîte à musique qui joue Brille, brille, petite étoile.

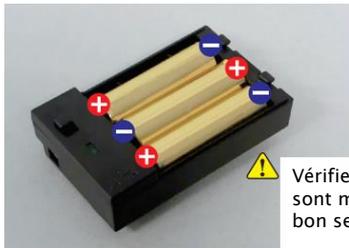
1 Ajouter un buzzer

Ajoute un buzzer à ton Studuino.

Tu auras besoin de...

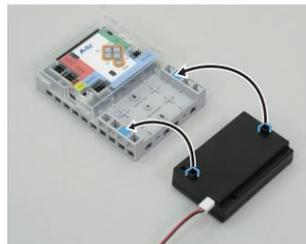


1 Mets dans la batterie 3 piles (AA/LR6).



⚠ Vérifie que les piles sont mises dans le bon sens !

Ajoute la batterie à ton Studuino comme montré ci-dessous.

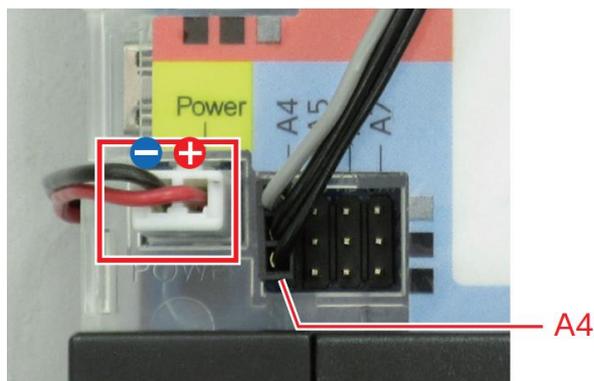


2 Ajoute le buzzer à ton Studuino comme illustré.

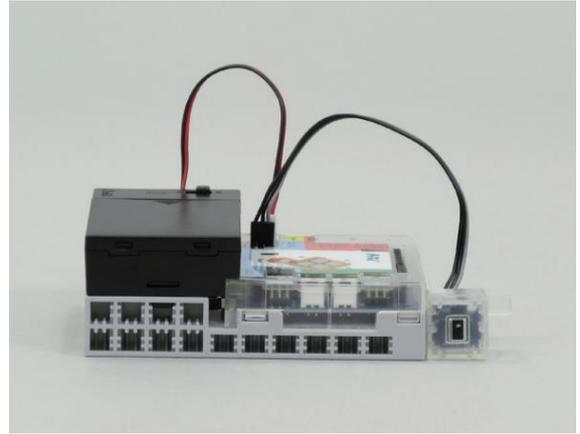


3 Branche la batterie sur le connecteur **Power** de ton Studuino et ton buzzer sur **A4**.

⚠ Vérifie que c'est bien inséré !



4 Fini !



2 Définir tes ports

Choisis **Button** pour **A0** et **Buzzer** pour **A4**.

A0

Button



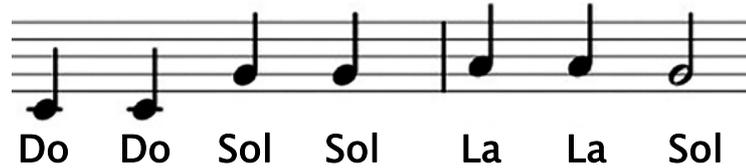
A4

Buzzer

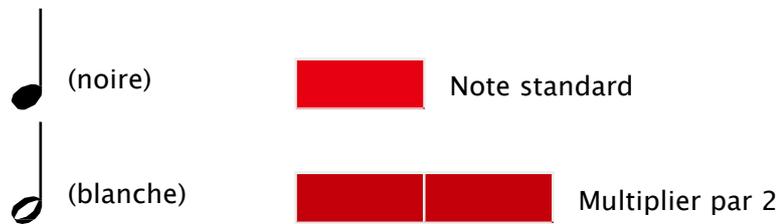


3 Choisir la longueur de la note

Nous allons jouer l'ouverture de Brille, brille, petite étoile en utilisant les variables pour choisir la longueur des notes !



Jette un œil sur la partition ci-dessus. Tu y verras deux types de notes : les notes noires  et les notes blanches . Regarde ci-dessous pour voir la relation entre ces deux notes. Tu verras que **dès que tu trouveras la longueur de la noire, tu pourras déterminer la longueur de la blanche automatiquement.**

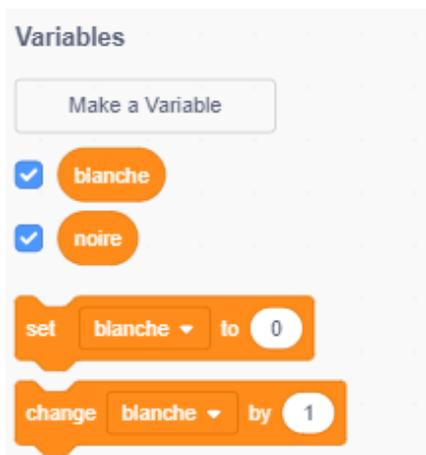


Mets le nombre de secondes pour tes notes noires et blanches dans des variables et fais une formule mathématique qui ressemble à celle-ci-dessous.

$$\text{blanche} = \text{noire} \times 2$$

Faisons un programme qui reproduit cette formule !

1 Crée tes variables **noire** et **blanche**.

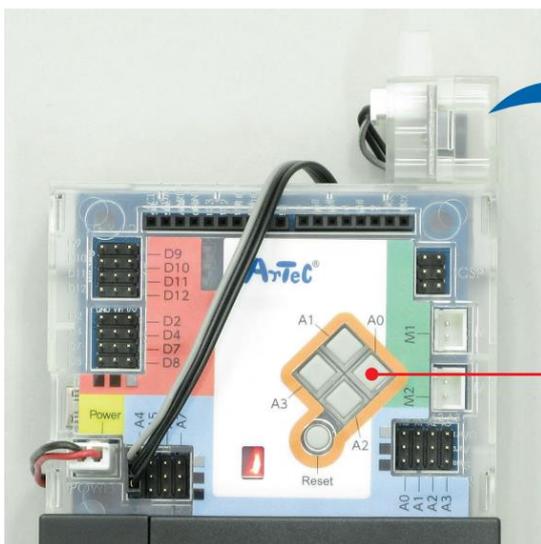


2 Programme le nombre de secondes de ta noire et de ta blanche. Pour cela, nous allons faire jouer ta noire pendant **0,5 secondes**. Comme une note blanche joue deux fois le temps d'une noire, nous pouvons utiliser une formule pour la calculer automatiquement. C'est là qu'intervient le bloc de multiplication .



4 Jouer Brille, brille, petite étoile

Programme la chanson pour qu'elle se joue lorsque tu appuies sur le bouton A0 de ton Studuino.



Appuyer sur A0



1 Crée une condition pour presser A0.



2 Programme le buzzer à jouer la mélodie de la partition.

The image shows a Scratch script on the right and a musical notation on the left. The script is a 'when clicked' event that sets two variables: 'noire' to 0.5 and 'blanche' to 'noire * 2'. It then enters a 'forever' loop with an 'if' condition: 'if Button A0 = 0 then'. Inside the 'if' block, there are two paths: one for a half note (frequency 60) and one for a whole note (frequency 67). The script uses 'buzzer A4 on frequency' blocks, 'wait' blocks, and 'buzzer A4 off' blocks to produce the melody. The musical notation on the left shows a sequence of notes: Do, Do, Sol, Sol, La, La, Sol. The notes are grouped into pairs, and the script blocks are aligned with these pairs. The first pair (Do, Do) is highlighted in blue, and the script blocks for this pair are also highlighted in blue. The second pair (Sol, Sol) is highlighted in blue, and the script blocks for this pair are also highlighted in blue. The third pair (La, La) is highlighted in blue, and the script blocks for this pair are also highlighted in blue. The fourth pair (Sol, Sol) is highlighted in blue, and the script blocks for this pair are also highlighted in blue. The fifth pair (La, La) is highlighted in blue, and the script blocks for this pair are also highlighted in blue. The sixth pair (Sol, Sol) is highlighted in blue, and the script blocks for this pair are also highlighted in blue. The seventh pair (La, La) is highlighted in blue, and the script blocks for this pair are also highlighted in blue. The eighth pair (Sol, Sol) is highlighted in blue, and the script blocks for this pair are also highlighted in blue.

when clicked

set **noire** to 0.5

set **blanche** to $\text{noire} * 2$

forever

if **Button A0** = 0 then

buzzer A4 on frequency: 60

wait **noire** seconds

buzzer A4 off

buzzer A4 on frequency: 60

wait **noire** seconds

buzzer A4 off

buzzer A4 on frequency: 67

wait **noire** seconds

buzzer A4 off

buzzer A4 on frequency: 67

wait **noire** seconds

buzzer A4 off

buzzer A4 on frequency: 69

wait **noire** seconds

buzzer A4 off

buzzer A4 on frequency: 69

wait **noire** seconds

buzzer A4 off

buzzer A4 on frequency: 67

wait **blanche** seconds

buzzer A4 off

Do Do Sol Sol La La Sol

3 Transfère ton programme et vois si ça fonctionne.

5 La touche finale

Le programme que tu viens de faire jouer la 1^{ère} mesure de Brille, brille, petite étoile. Maintenant regarde la partition ci-dessous et finis le reste !

The image shows a musical score for the song 'Brille, brille, petite étoile'. It consists of three staves of music, each with a corresponding line of lyrics below it. The notes are represented by quarter notes on a five-line staff. The lyrics are:
Staff 1: Do Do Sol Sol La La Sol
Staff 2: Fa Fa Mi Mi Ré Ré Do
Staff 3: Sol Sol Fa Fa Mi Mi Ré
Staff 4: Do Do Sol Sol La La Sol
Staff 5: Fa Fa Mi Mi Ré Ré Do

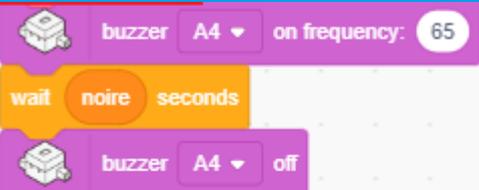
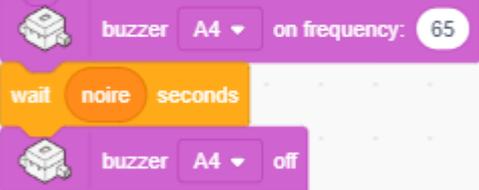
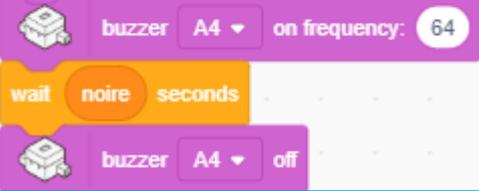
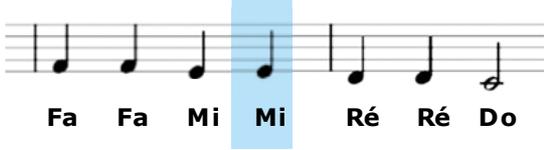
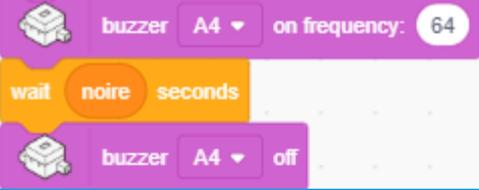
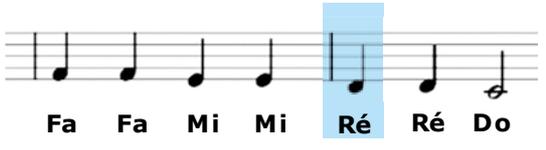
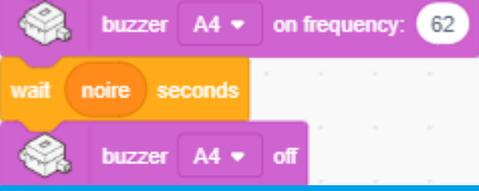
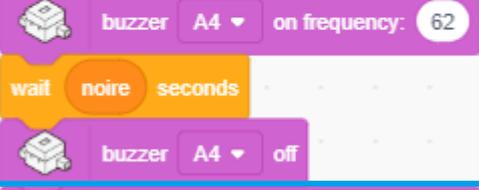
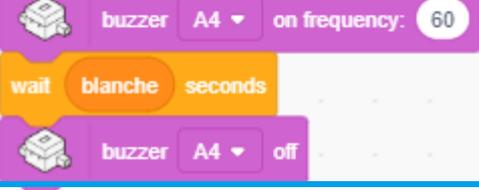
Tu peux faire la chanson en associant ensemble trois mélodies différentes. Mettre ces mélodies dans des fonctions t'épargnera beaucoup de temps pour reproduire la chanson en entier ! Ajoute des fonctions à ton programme et finis de faire Brille, brille, petite étoile.

1 Mets la première mesure de la chanson dans une fonction appelée 1.



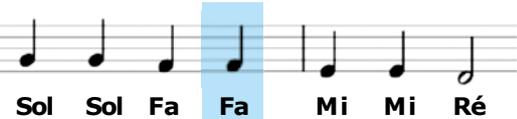
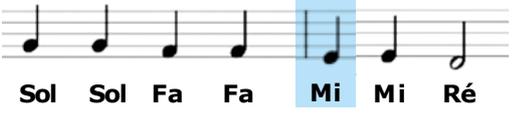
The image shows a Scratch code editor with a program to play the first measure of the song. The program starts with a 'when clicked' event, followed by setting 'noire' to 0.5 and 'blanche' to 'noire + 2'. A 'forever' loop contains an 'if' statement that checks if 'Button A0' is equal to 0. If true, it calls 'fonction 1'. The 'fonction 1' block contains a sequence of buzzer and wait blocks:
1. buzzer A4 on frequency: 60
2. wait noire seconds
3. buzzer A4 off
4. buzzer A4 on frequency: 60
5. wait noire seconds
6. buzzer A4 off
7. buzzer A4 on frequency: 67
8. wait noire seconds
9. buzzer A4 off
10. buzzer A4 on frequency: 67
11. wait noire seconds
12. buzzer A4 off
13. buzzer A4 on frequency: 69
14. wait noire seconds
15. buzzer A4 off
16. buzzer A4 on frequency: 69
17. wait noire seconds
18. buzzer A4 off
19. buzzer A4 on frequency: 67
20. wait blanche seconds
21. buzzer A4 off

2 Crée une fonction appelée 2 pour la deuxième mesure. Duplique simplement la fonction 1 et modifie les notes par celles ci-dessous.

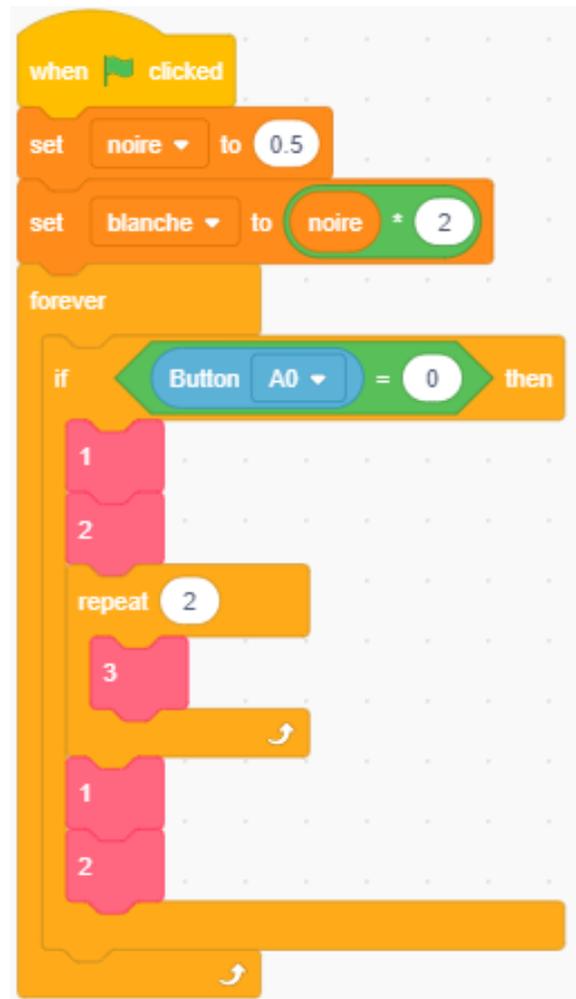
	function 2
 <p>Fa Fa Mi Mi Ré Ré Do</p>	
 <p>Fa Fa Mi Mi Ré Ré Do</p>	
 <p>Fa Fa Mi Mi Ré Ré Do</p>	
 <p>Fa Fa Mi Mi Ré Ré Do</p>	
 <p>Fa Fa Mi Mi Ré Ré Do</p>	
 <p>Fa Fa Mi Mi Ré Ré Do</p>	
 <p>Fa Fa Mi Mi Ré Ré Do</p>	

3 Crée une fonction appelée 3 pour la troisième mesure. Duplique simplement la fonction 2 et modifie les notes par celles-ci-dessous.

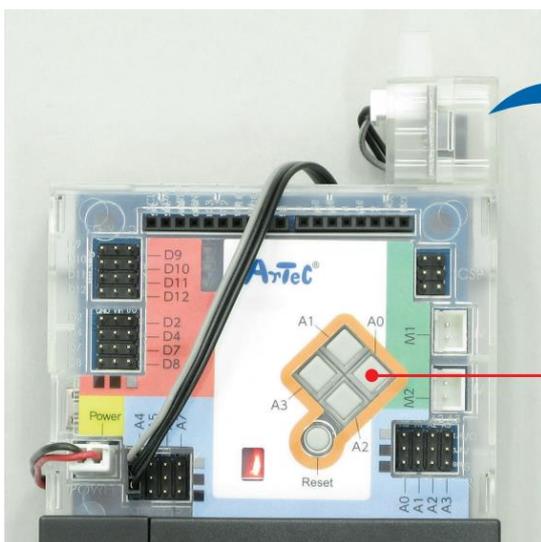
function 3

 <p>Sol Sol Fa Fa Mi Mi Ré</p>	  
 <p>Sol Sol Fa Fa Mi Mi Ré</p>	  
 <p>Sol Sol Fa Fa Mi Mi Ré</p>	  
 <p>Sol Sol Fa Fa Mi Mi Ré</p>	  
 <p>Sol Sol Fa Fa Mi Mi Ré</p>	  
 <p>Sol Sol Fa Fa Mi Mi Ré</p>	  
 <p>Sol Sol Fa Fa Mi Mi Ré</p>	  

4 Pour jouer la chanson correctement, placez les fonctions à l'intérieur du bloc



5 Transfère ton programme et vois si ça fonctionne !

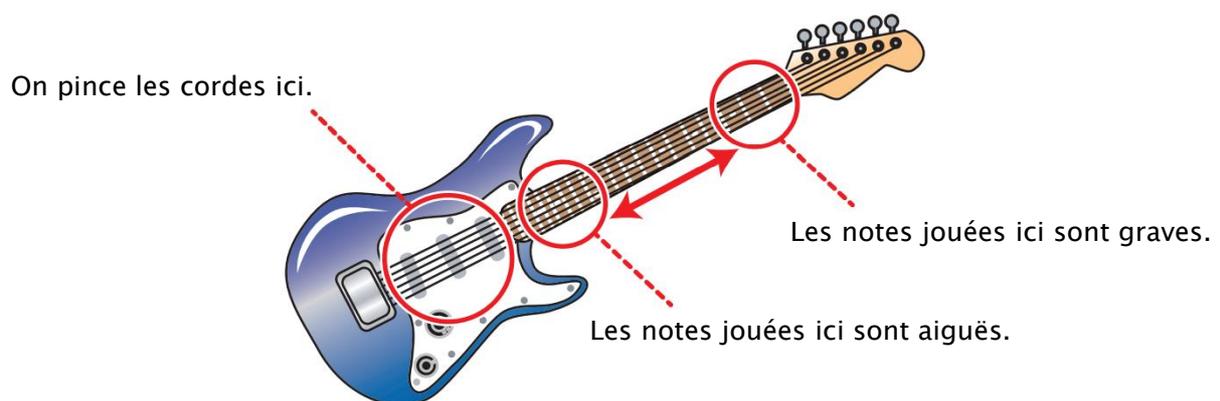


Appuyer sur A0

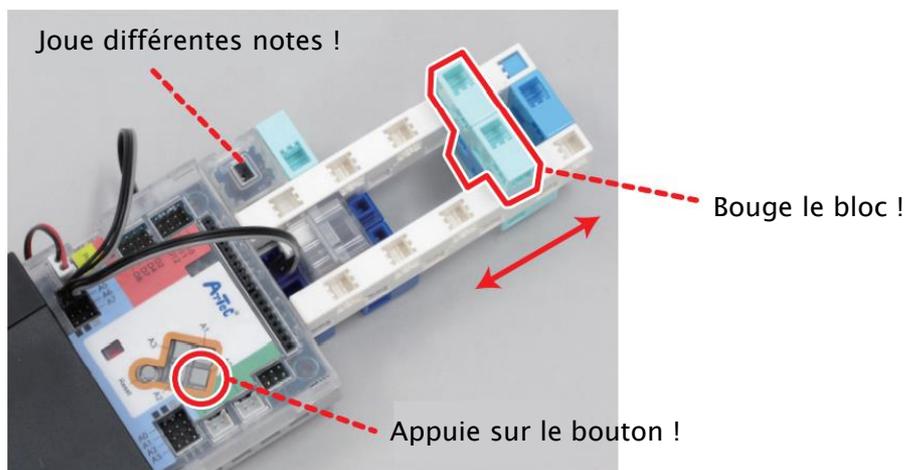
Chapitre 3

Faire une guitare

Si tu as déjà joué de la guitare, tu dois savoir qu'il faut pincer ses cordes pour émettre des sons. Tu utilise ton autre main pour appuyer sur les cordes afin de changer de notes. Si tu appuies sur les cordes vers le bas de ta guitare, les notes seront plus aiguës, mais si tu appuies dessus vers le haut, les notes seront plus graves !



Cette distance entre le haut et le bas de la guitare contrôle la hauteur de la note. Savais-tu qu'on pouvait utiliser un photoréflexeur infrarouge pour faire un instrument qui joue comme une guitare ? Nous allons faire une guitare à laquelle tu peux jouer en utilisant les boutons sur ton Studuino et un bloc.



1 Construire une guitare

Nous allons utiliser ton photoréflexeur IR pour mettre au point une guitare.

Tu auras besoin de...



Studuino x 1



Câble USB x 1



Batterie x 1



Câble de connexion de capteur (S) x 2 (3 fils, 15 cm)



Barre x 2



Demi-cube B (bleu) x 4



Demi-cube D (turquoise) x 1



Demi-cube C (bleu clair) x 5

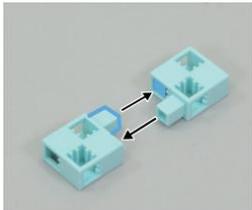


Buzzer x 1

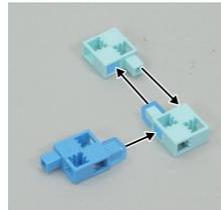


Photoréflexion IR x 1

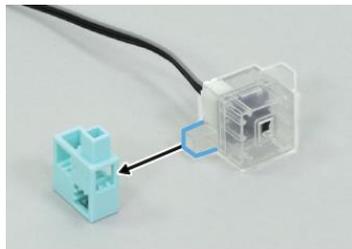
1 Connecte ces blocs.



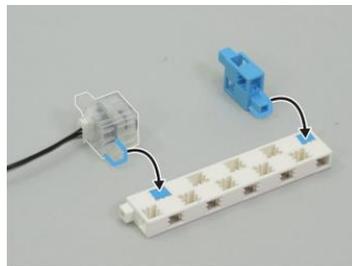
2 Connecte ces blocs.



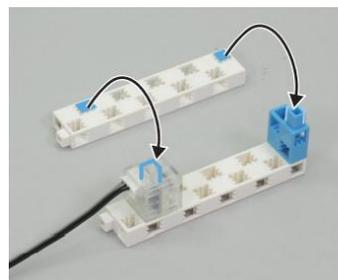
3 Ajoute ce bloc à ton buzzer.



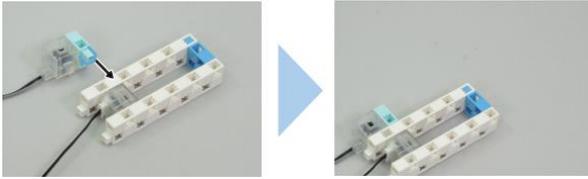
4 Ajoute le capteur IR et le bloc montré ci-contre à la barre.



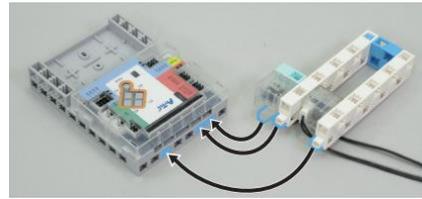
5 Ajoute une autre barre à la partie **4** comme montré ci-contre.



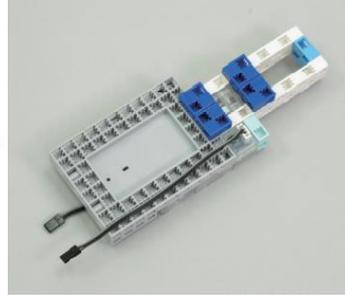
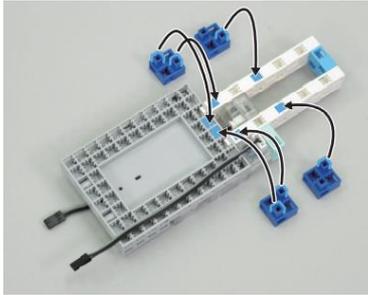
6 Ajoute la partie **3** à la partie **5**.



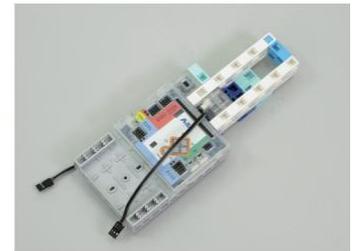
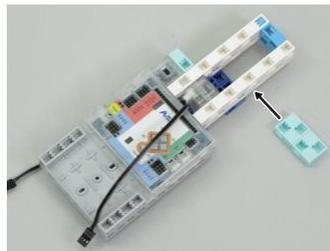
7 Ajoute la partie **6** à ton Studuino.



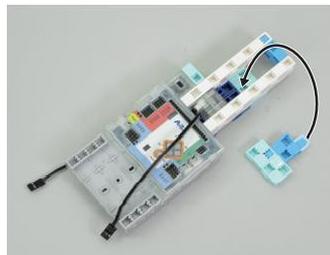
8 Ajoute les blocs ci-dessous sous la partie **7**.



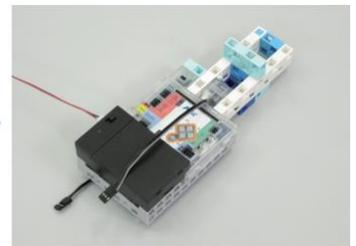
9 Ajoute la partie **1** à la partie **8**.



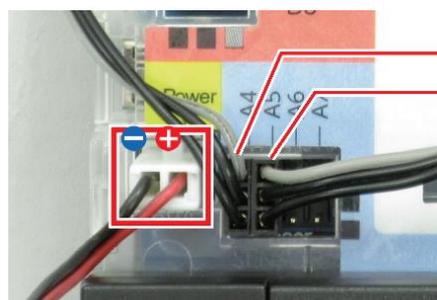
10 Ajoute la partie **2** au-dessus de la partie **9**.



11 Ajoute la batterie (avec des piles) à la partie **10**.

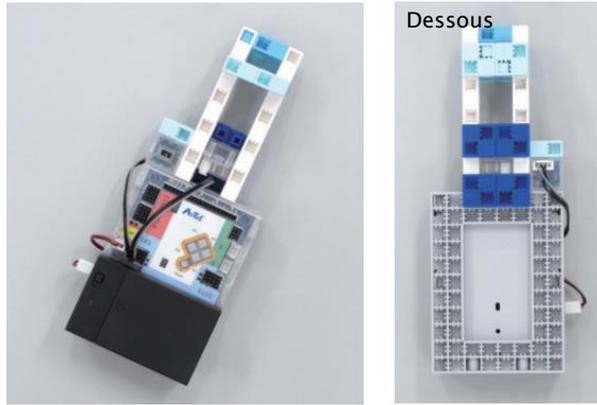


12 Branche la batterie sur le connecteur **Power** du Studuino, le buzzer sur **A4** et le photoréflexeur sur **A5**.



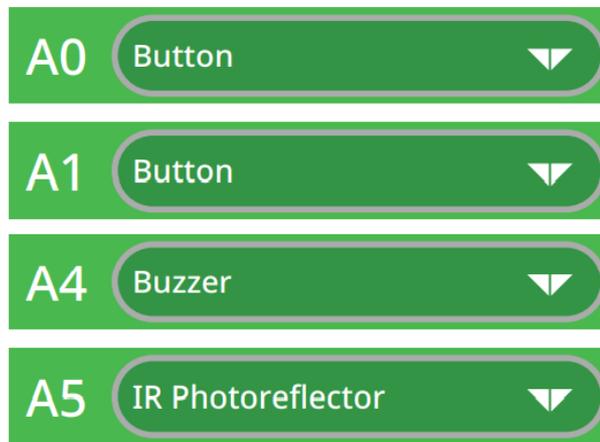
⚠ Vérifie que c'est bien branché !

13 Fini !



2 Définir tes ports

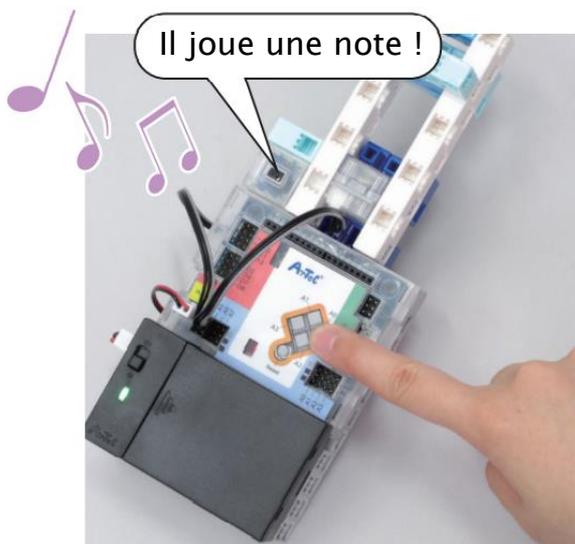
Choisis **Button** pour **A0** et **A1**, **Buzzer** pour **A4** et **IR Photoreflector** pour **A5**.



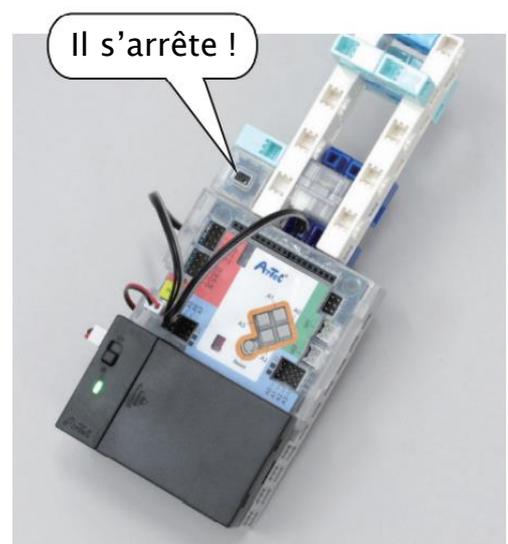
3 Jouer de ton instrument

1 Appuie sur le bouton A0 pour jouer une note.

Appuie dessus...



Relâche-le !

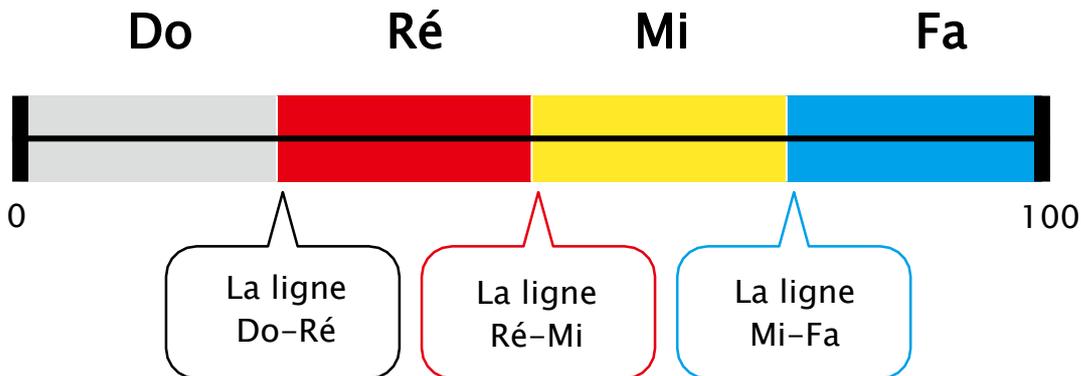


2 Ton instrument jouera les notes Do (60), Ré (62), Mi (64) ou Fa (65) selon la position du bloc.

Note	Position
Do (60)	
Ré (62)	
Mi (64)	
Fa (65)	

4 Notes et valeurs du photoréfecteur IR

Les notes de ton instrument changeront de hauteur en fonction de la zone dans laquelle se trouve le bloc. Puisqu'il joue quatre notes, tu auras besoin de trois lignes de délimitation pour faire chaque zone. Observe les valeurs de ton photoréfecteur IR pour les zones allant de Do à Ré, de Ré à Mi et de Mi à Fa.



1 Lance le mode test et observe les valeurs de ton photoréfecteur IR lorsque tu es sur chacune des lignes.

Ligne	Position	Photoréfecteur IR
Do - Ré		①
Ré - Mi		②
Mi - Fa		③

2 Utilise les valeurs de ton photorélecteur IR pour trouver une zone pour chaque note.

Note	Range
Do (60)	Moins que <input type="text" value="①"/>
Ré (62)	<input type="text" value="①"/> à <input type="text" value="②"/>
Mi (64)	Ajoute ① à ② ici ! <input type="text" value="①"/> à <input type="text" value="③"/>
Fa (65)	Plus grand que <input type="text" value="③"/>

5 Faire des conditions pour les notes

Tu peux utiliser un bloc  quand tu souhaites définir une plage entre deux valeurs pour une condition. Le bloc-même  crée une condition qui est vraie **lorsque les deux parties de la condition sont vraies** !

Afficher la plage

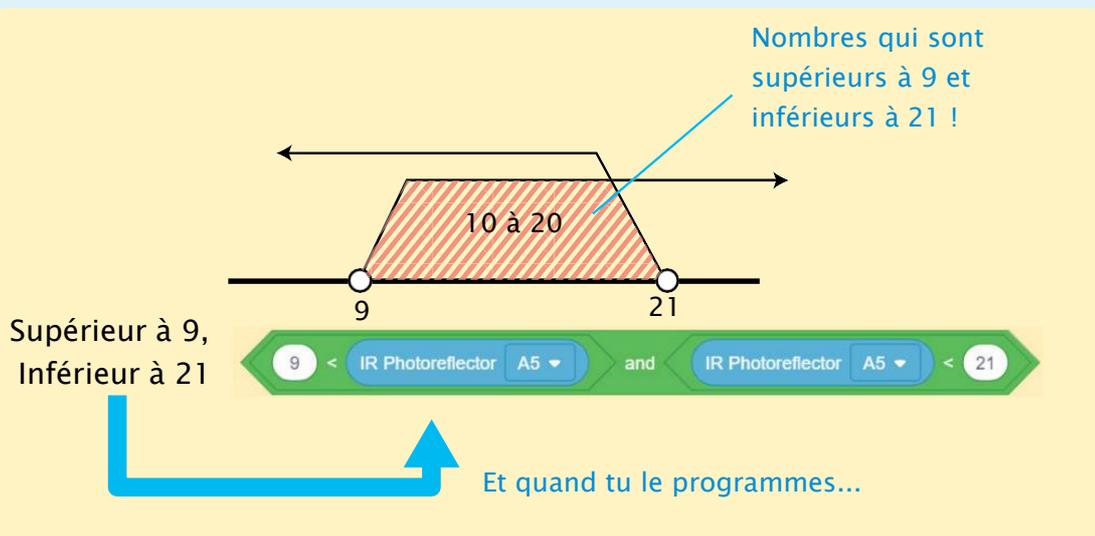
En guise d'exemple, utilisons un bloc  pour afficher une plage allant de 10 à 20. Pense à cette plage comme étant supérieure à une valeur et inférieure à une autre valeur, ce qui pourrait donner par exemple **supérieure à 9, inférieure à 21** !

10 à 20

donne...

Supérieur à 9, inférieur à 21

Une plage supérieure à une valeur A et inférieure à une valeur B n'inclura pas les valeurs A et B. C'est pourquoi tu dois soustraire 1 à la valeur la plus petite et ajouter 1 à la valeur la plus grande, ce qui donne une plage **supérieure à 9 (10 - 1) et inférieure à 21 (20 + 1)**. Mets ces deux conditions dans le bloc .



Le bloc  utilise le mot « and » pour vérifier si les deux conditions sont vraies. Cela signifie que ce bloc sera vrai pour tous les nombres allant de 10 à 20 parce qu'ils sont tous supérieurs à 9 et inférieurs à 21 !

1 Transformons les plages que tu as trouvées à la page 26 en expressions « supérieur à..., inférieur à... ». Attention, ces plages n'incluent pas les nombres apparaissant dans l'expression !

Note	Plage	
Do (60)	Inférieur à	<input type="text"/>
Ré (62)	Supérieur à	<input type="text"/>
	Inférieur à	<input type="text"/>
Mi (64)	Supérieur à	<input type="text"/>
	Inférieur à	<input type="text"/>
Fa (65)	Supérieur à	<input type="text"/>

2 Programme la note Do (60) !

The image shows a Scratch code block with an orange background. It starts with an 'if' block containing 'IR Photorelector A5' and '< 11'. Below it is a 'buzzer A4 on frequency: 60' block. A callout bubble labeled 'Do' points to the frequency value '60'. Another callout bubble says 'C'est la valeur qui se trouve dans [] !' with a black-bordered box.

3 Programme la note Ré (62) !

The image shows a Scratch code block with an orange background. It starts with an 'if' block containing '10 < IR Photorelector A5' and 'and IR Photorelector A5 < 18'. Below it is a 'buzzer A4 on frequency: 62' block. A callout bubble labeled 'Ré' points to the frequency value '62'. Two other callout bubbles say 'C'est la valeur dans [] !' with a grey-bordered box and 'C'est la valeur dans [] !' with a red-bordered box.

4 Programme la note Mi (64) !

The image shows a Scratch code block with an orange background. It starts with an 'if' block containing 'IR Photorelector A5' and '< 11'. Below it is a 'buzzer A4 on frequency: 60' block. The next 'if' block contains '10 < IR Photorelector A5' and 'and IR Photorelector A5 < 18'. Below it is a 'buzzer A4 on frequency: 62' block. The final 'if' block contains '17 < IR Photorelector A5' and 'and IR Photorelector A5 < 30'. Below it is a 'buzzer A4 on frequency: 64' block. A callout bubble labeled 'Mi' points to the frequency value '64'. Three other callout bubbles say 'C'est la valeur dans [] !' with a yellow-bordered box, 'C'est la valeur dans [] !' with a blue-bordered box, and another 'C'est la valeur dans [] !' with a blue-bordered box.

5 Programme la note Fa (65) !

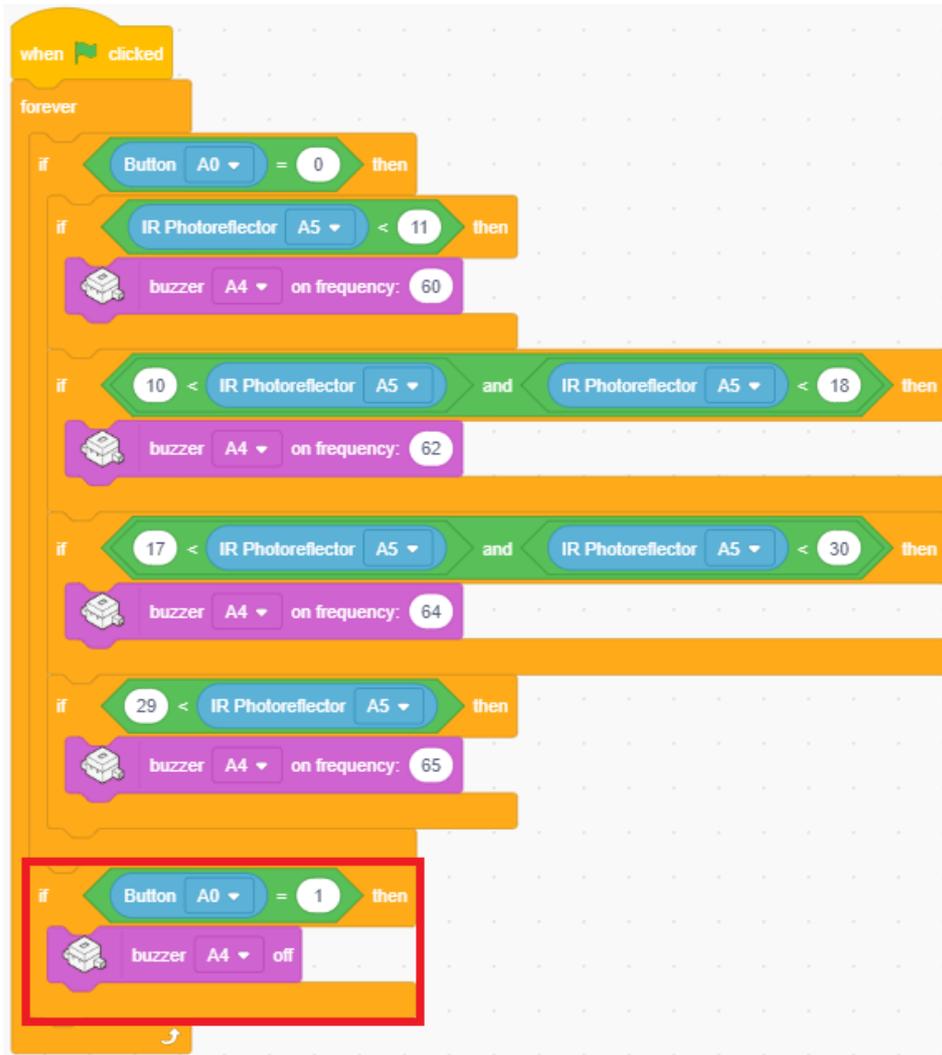
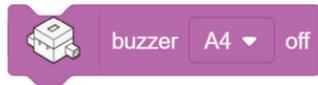
C'est la valeur dans [] !

Fa

6 Puisque tu dois appuyer sur A0 pour jouer une note, crée une condition qui servira lorsque tu presses le bouton et insère ton programme fait en 5 dans cette condition.

```
when clicked
forever
  if Button A0 = 0 then
    if IR Photorelector A5 < 11 then
      buzzer A4 on frequency: 60
    if 10 < IR Photorelector A5 and IR Photorelector A5 < 18 then
      buzzer A4 on frequency: 62
    if 17 < IR Photorelector A5 and IR Photorelector A5 < 30 then
      buzzer A4 on frequency: 64
    if 29 < IR Photorelector A5 then
      buzzer A4 on frequency: 65
```

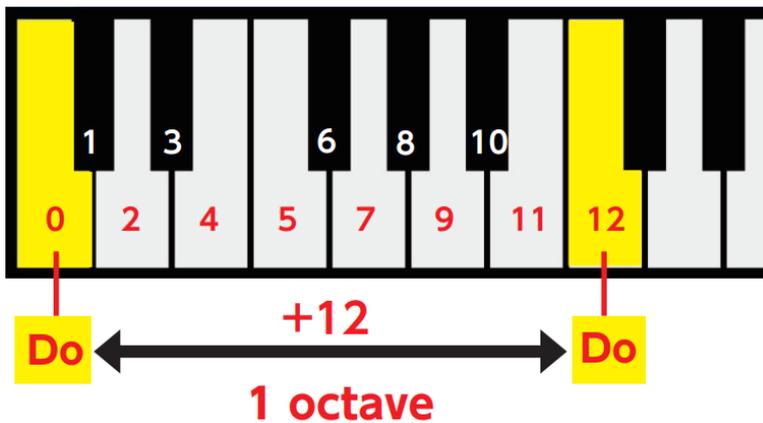
5 Puisque aucune note n'est jouée quand tu n'appuies pas sur A0, crée une condition qui servira lorsque le bouton est relâché et place un bloc à l'intérieur de ta condition.



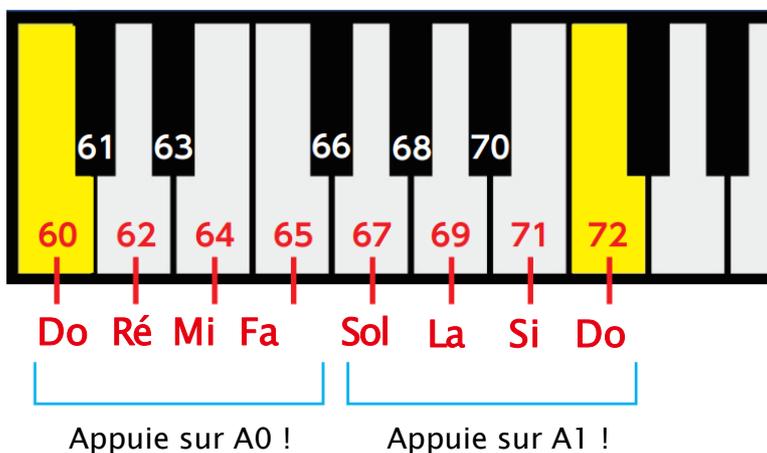
6 Transfère ton programme et vois si ça fonctionne.

6 Jouer plus de notes

Le programme que tu viens de faire jouera seulement les notes Do (60), Ré (62), Mi (64) et Fa (65) mais on peut ajouter une autre octave à ton instrument en utilisant le bouton A1 pour jouer les notes Sol (67), La (69), Si (71) et Do (72) ! Une octave est l'intervalle de 12 notes séparant le premier Do du dernier Do.



En partant de Do (60), les nombres que tu utilises pour faire jouer des notes à ton buzzer correspondent aux numéros des touches du clavier montré ci-dessous. Programme ton instrument à jouer les notes Sol (67), La (69), Si (71) et Do (72) selon la position du bloc quand tu appuies sur A1.



1 Duplique la condition faite pour A0. Fais jouer, dans cette nouvelle condition, les notes Sol (67), La (69), Si (71) ou Do (72) quand tu appuies sur A1 !

```
when clicked
  forever
    if Button A0 = 0 then
      if IR Photorelector A5 < 11 then
        buzzer A4 on frequency: 60
      if 10 < IR Photorelector A5 and IR Photorelector A5 < 18 then
        buzzer A4 on frequency: 62
      if 17 < IR Photorelector A5 and IR Photorelector A5 < 30 then
        buzzer A4 on frequency: 64
      if 29 < IR Photorelector A5 then
        buzzer A4 on frequency: 65
    if Button A1 = 1 then
      IR Photorelector A5 < 11 then
        buzzer A4 on frequency: 67
      if 10 < IR Photorelector A5 and IR Photorelector A5 < 18 then
        buzzer A4 on frequency: 69
      if 17 < IR Photorelector A5 and IR Photorelector A5 < 30 then
        buzzer A4 on frequency: 71
      if 29 < IR Photorelector A5 then
        buzzer A4 on frequency: 72
    if Button A0 = 1 then
      buzzer A4 off
```

Duplique-la !

Change-le en A1 !

Sol

La

Si

Do

2 Crée une condition pour arrêter les notes. Ces notes s'arrêteront à chaque fois que tu relâcheras A0 ou A1. Tu peux le faire en utilisant un bloc **and** pour faire une condition qui indique que **A0 et A1 sont relâchés**.

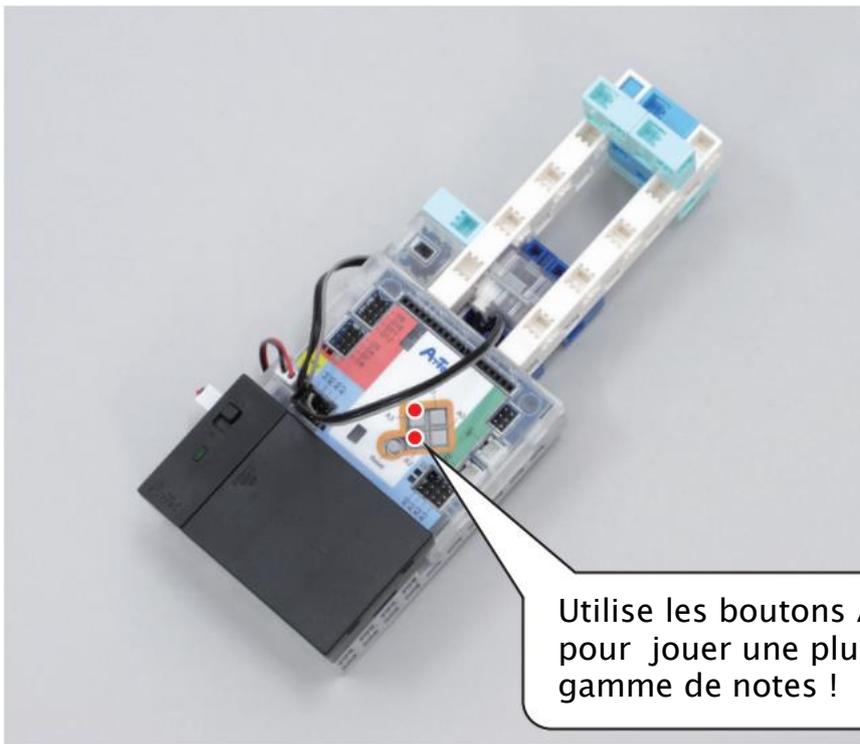
```
when clicked
forever
  if Button A0 = 0 then
    if IR Photorelector A5 < 11 then
      buzzer A4 on frequency: 60
    if 10 < IR Photorelector A5 and IR Photorelector A5 < 18 then
      buzzer A4 on frequency: 62
    if 17 < IR Photorelector A5 and IR Photorelector A5 < 30 then
      buzzer A4 on frequency: 64
    if 29 < IR Photorelector A5 then
      buzzer A4 on frequency: 65
  if Button A1 = 1 then
    if IR Photorelector A5 < 11 then
      buzzer A4 on frequency: 67
    if 10 < IR Photorelector A5 and IR Photorelector A5 < 18 then
      buzzer A4 on frequency: 69
    if 17 < IR Photorelector A5 and IR Photorelector A5 < 30 then
      buzzer A4 on frequency: 71
    if 29 < IR Photorelector A5 then
      buzzer A4 on frequency: 72
  if Button A0 = 1 and Button A1 = 1 then
    buzzer A4 off
```

3 Transfère ton programme et vois si ça fonctionne !

Chapitre 4

Une meilleure guitare

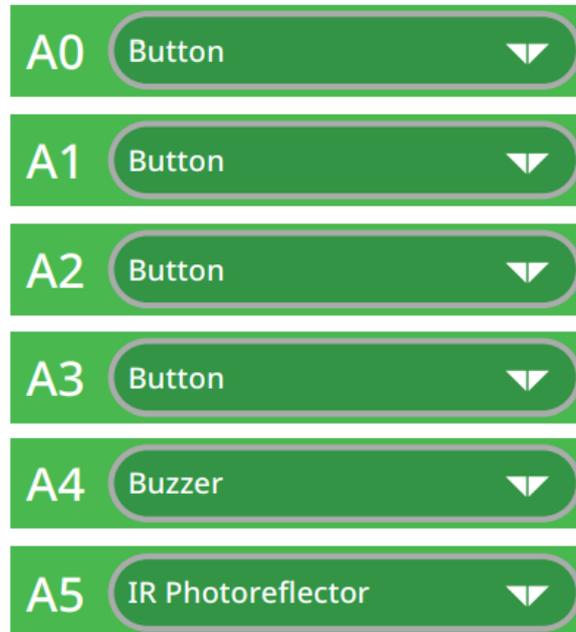
Au chapitre 3, tu as fait une guitare avec laquelle tu pouvais jouer en appuyant sur les boutons A0 et A1 de ton Studuino pour choisir parmi une gamme de notes dans une octave. Tu peux utiliser plus de boutons pour jouer une plus large gamme encore ! Dans ce chapitre, nous allons ajouter les boutons A2 et A3 pour que ta guitare joue encore plus de notes !



Nous allons utiliser autant de fonctions et de variables que possible pour rendre le programme court et facile à comprendre !

1 Définir les ports

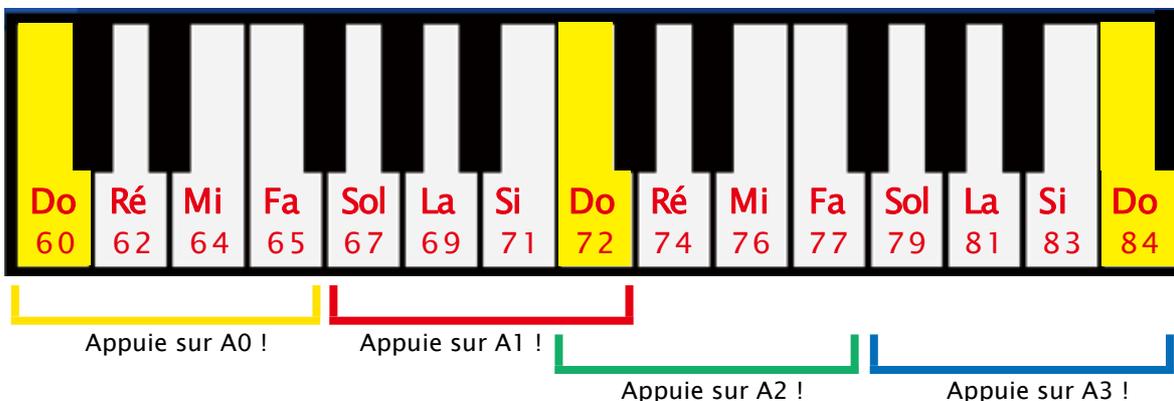
Choisis **Button** pour A0 à A3, **Buzzer** pour A4 et **IR Photorelector** pour A5 !



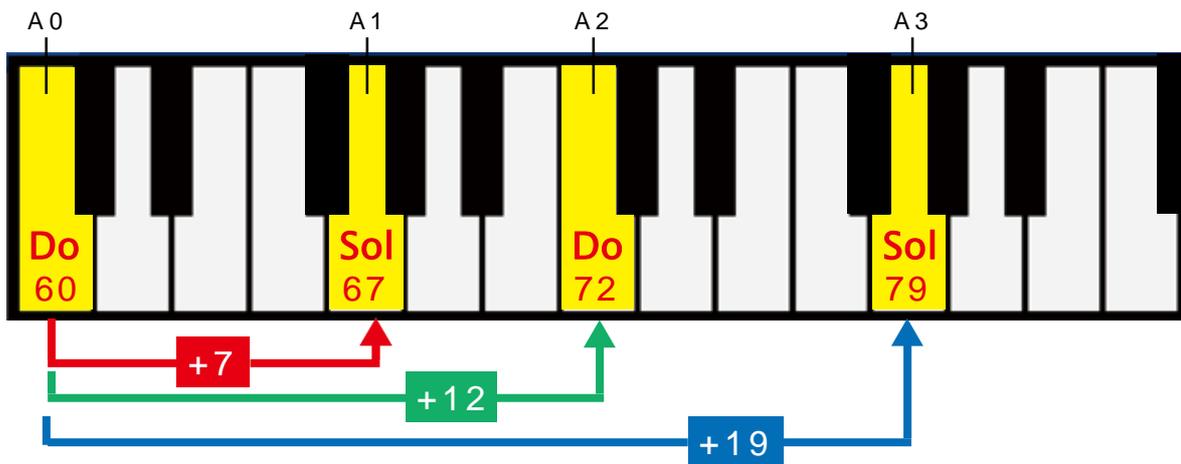
2 Notes et nombres

Améliore ton programme du chapitre 3 en ajoutant les boutons A2 et A3 pour étendre la gamme de ta guitare de deux octaves !

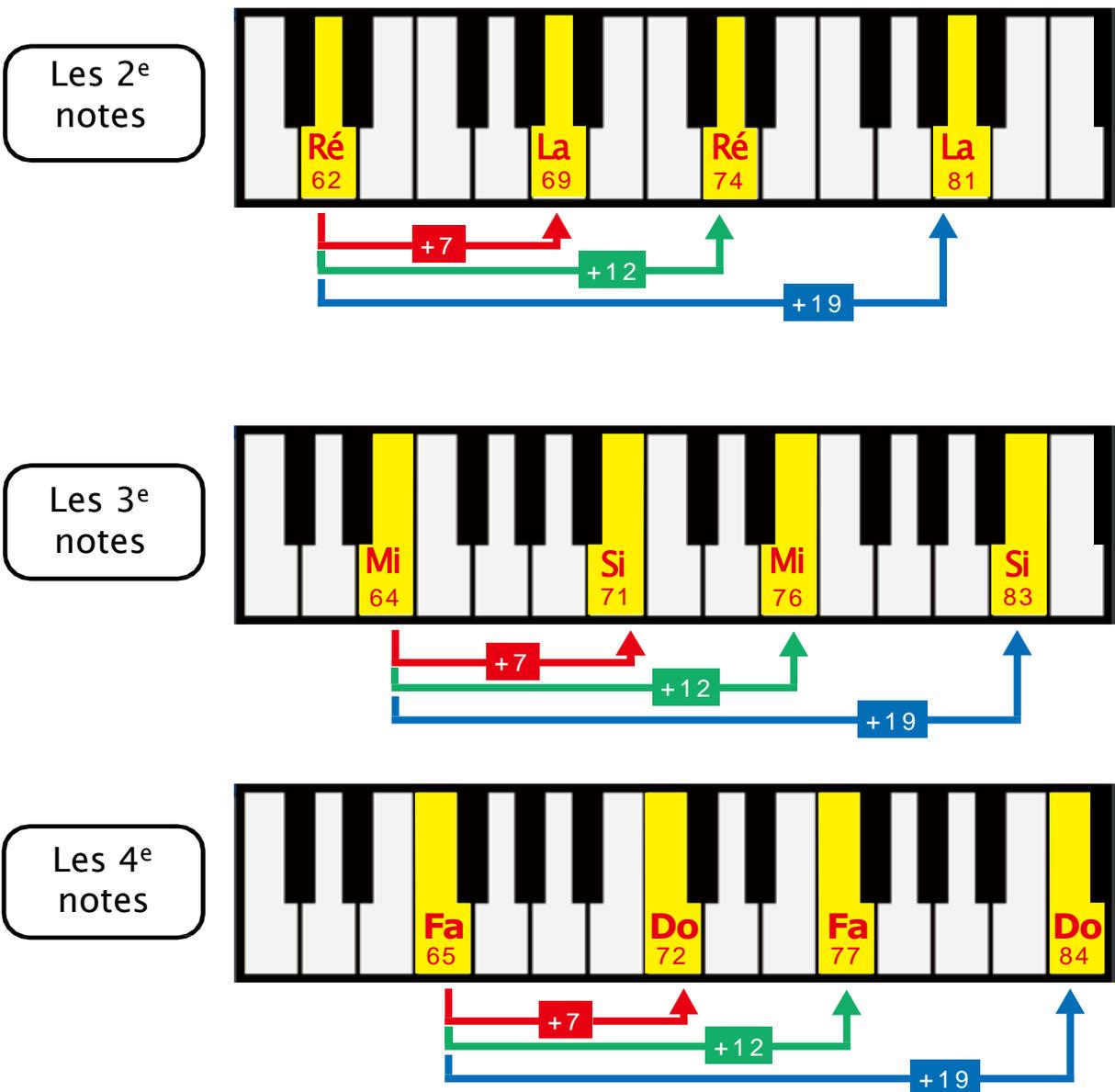
As-tu remarqué que les nombres des notes sont réguliers ? Cette régularité permettra de la programmer très facilement !



1 Regarde la note avec le numéro le plus bas pour chaque bouton.



2 Regarde maintenant les notes restantes. Tu verras que les nombres augmentent de la même façon !



3 Mets en évidence dans un tableau la relation entre ces nombres. Observe les notes du clavier à la page 35. Tu verras que tu peux trouver le reste des notes en ajoutant les nombres aux notes de base Do (60), Ré (62), Mi (64) et Fa (65) qui jouent lorsque tu appuies sur A0.

Boutons	A0	A1	A2	A3
1 ^{er}	Do (60)	Sol (67)	Do (72)	Sol (79)
2 ^e	Ré (62)	La (69)	Ré (74)	La (81)
3 ^e	Mi (64)	Si (71)	Mi (76)	Si (83)
4 ^e	Fa (65)	Do (72)	Fa (77)	Do (84)

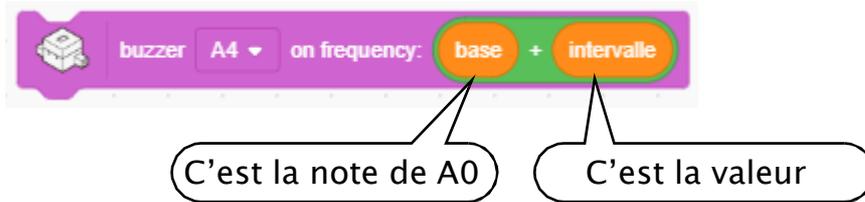
Transforme ce tableau en formules mathématiques et tu obtiendras les notes suivantes quand tu appuies sur chacun des boutons !

Notes de A1	=	Notes de A0	+	7
Notes de A2	=	Notes de A0	+	12
Notes de A3	=	Notes de A0	+	19

Tu peux faire les notes pour A1, A2 ou A3 en ajoutant 7, 12 ou 19 aux notes de A0 ! Cela signifie que tu peux trouver chaque note en utilisant la formule ci-dessous !

Note	=	Notes de A0	+	Valeur
		(60) Do (62) Ré (64) Mi (65) Fa		7 (pour A1) 12 (pour A2) 19 (pour A3)

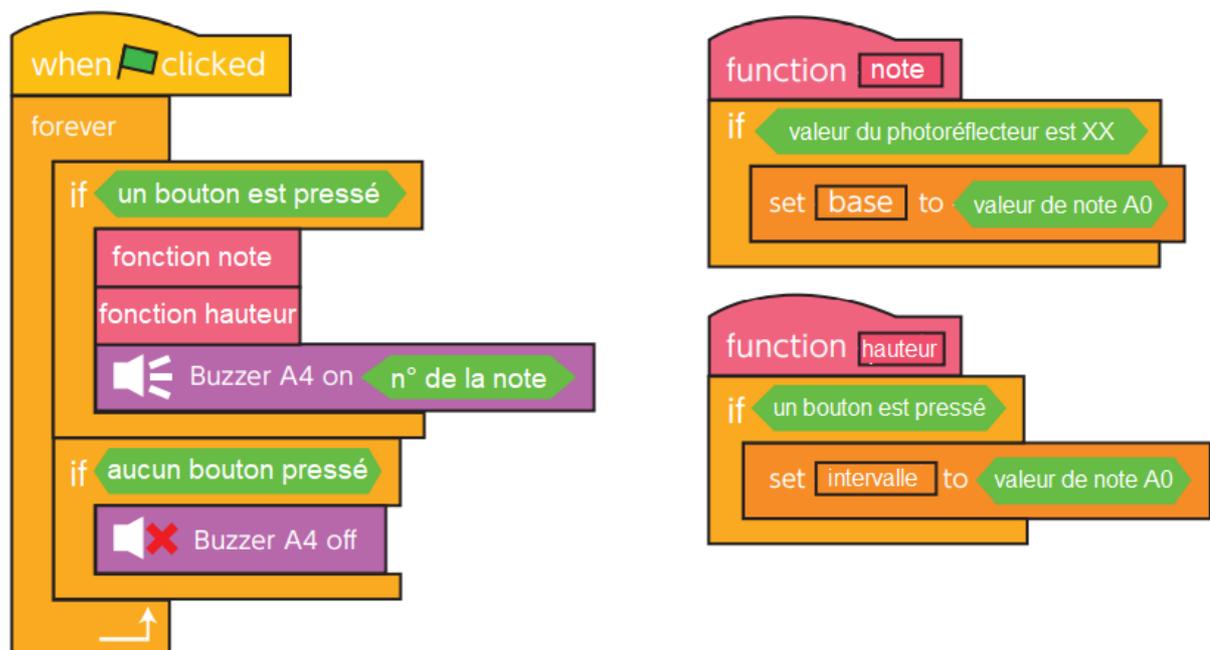
Les notes de A0 changeront à la fois en fonction de la position du bloc et en fonction de la valeur du bouton pressé. On peut utiliser des variables ici pour rendre ton programme beaucoup plus court.



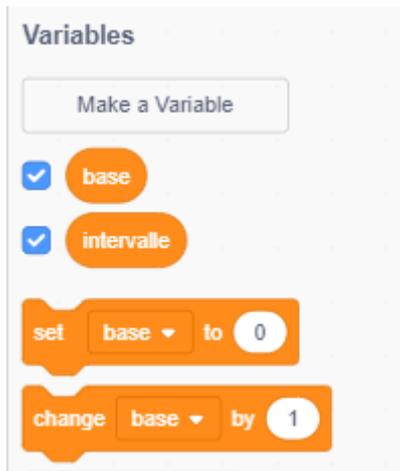
Nous allons créer une variable appelée « base » qui stocke les notes de A0 et une autre variable appelée « intervalle » qui stocke les valeurs dans ton programme.

3 Programmer deux octaves

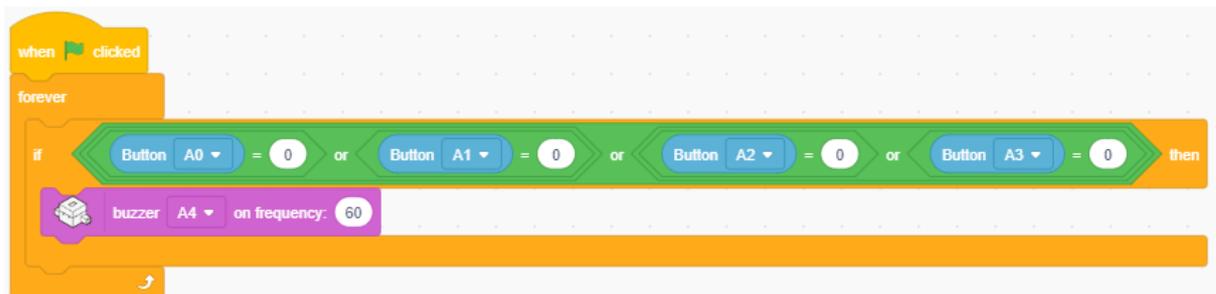
Nous allons faire un programme qui utilise la relation entre les notes que nous avons mise en évidence ! Nous allons devoir faire le même programme que celui du chapitre 3 qui **joue les notes quand tu appuies sur les boutons et les arrête quand tu les relâches !**



- 1 Fais une variable appelée « base » qui stocke les notes de A0 et une autre variable appelée « intervalle » qui stocke les valeurs !

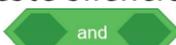


- 2 Programme ton buzzer pour jouer les notes quand tu appuies sur les boutons de ton Studuino ! Utilise un bloc  pour faire une condition qui servira lorsque **une des deux conditions est vraie** !



- 3 Fais en sorte que la note qui se joue soit la somme de la base et de l'intervalle.

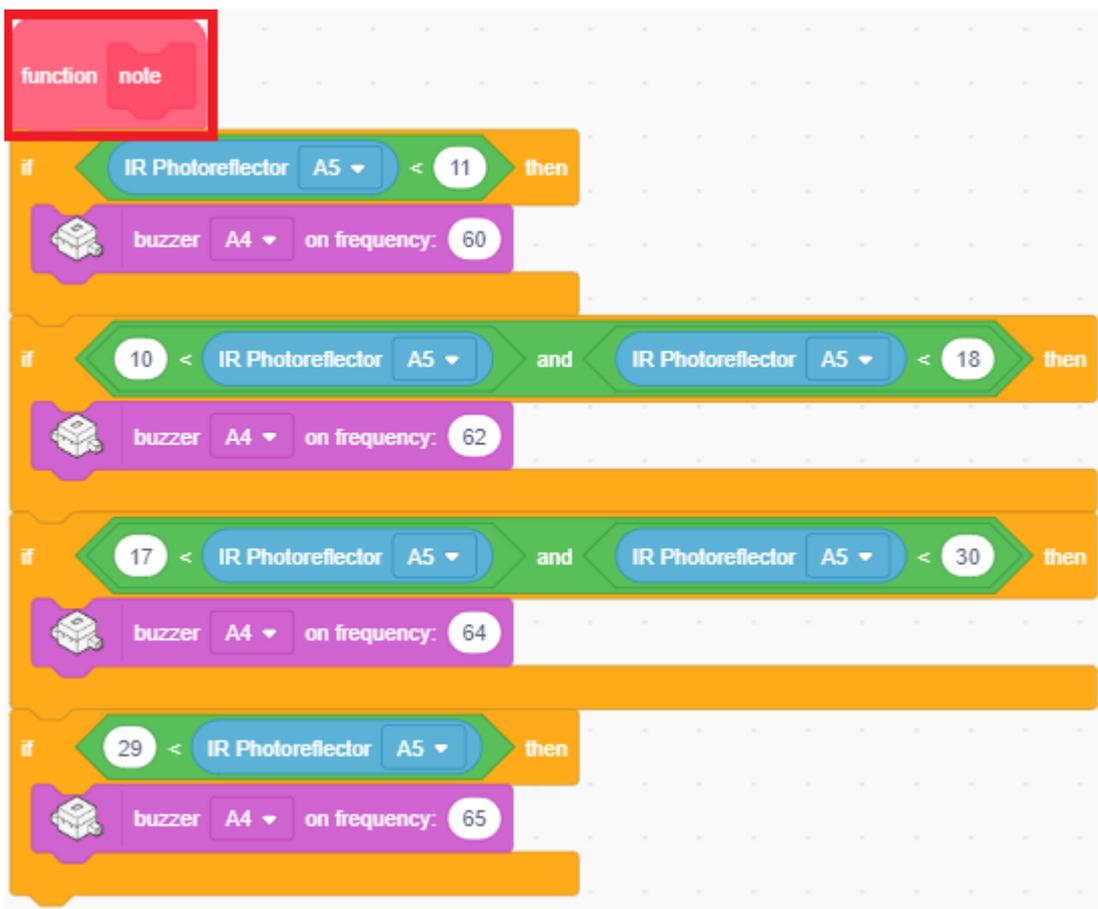


4 Fais en sorte que ton instrument reste silencieux quand les boutons du Studuino sont relâchés. Utilise un bloc  pour créer la condition.



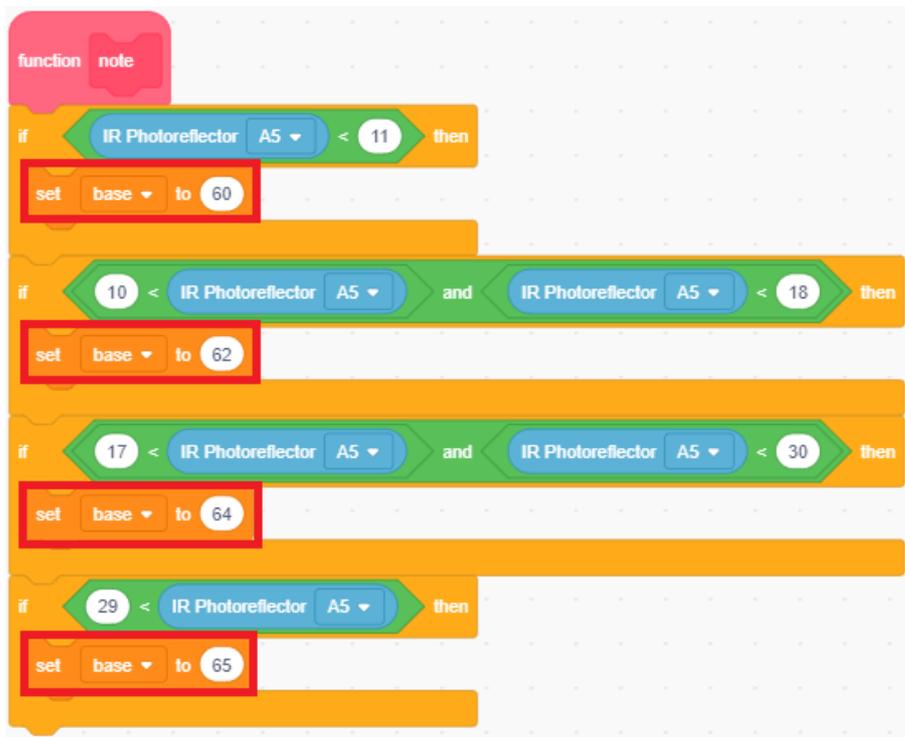
```
when clicked
  forever
    if (Button A0 = 0 or Button A1 = 0 or Button A2 = 0 or Button A3 = 0) then
      buzzer A4 on frequency: base + intervalle
    if (Button A0 = 1 and Button A1 = 1 and Button A2 = 1 and Button A3 = 1) then
      buzzer A4 off
```

5 Crée une fonction appelée « **note** » qui décide quelles valeurs stocker dans la variable « **base** ». Cette fonction est en fait le programme que tu as fait au chapitre 3 pour **jouer une note lorsqu'on presse le bouton A0**.



```
function note
  if (IR Photoreflexor A5 < 11) then
    buzzer A4 on frequency: 60
  if (10 < IR Photoreflexor A5 and IR Photoreflexor A5 < 18) then
    buzzer A4 on frequency: 62
  if (17 < IR Photoreflexor A5 and IR Photoreflexor A5 < 30) then
    buzzer A4 on frequency: 64
  if (29 < IR Photoreflexor A5) then
    buzzer A4 on frequency: 65
```

6 Remplace les blocs  par des blocs  pour que ton programme enregistre les valeurs des notes quand tu appuies sur A0.



```
function note
if IR Photorelector A5 < 11 then
set base to 60
if 10 < IR Photorelector A5 and IR Photorelector A5 < 18 then
set base to 62
if 17 < IR Photorelector A5 and IR Photorelector A5 < 30 then
set base to 64
if 29 < IR Photorelector A5 then
set base to 65
```

7 Crée une fonction appelée « hauteur » qui décide quelles valeurs stocker dans la variable « intervalle ». Cette fonction change la variable « intervalle » en valeurs que tu as trouvées à la page 36 quelque soit le bouton que tu presses. **Puisque tu n'as pas besoin d'ajouter quoi que ce soit quand tu appuies sur A0, continue en réglant la valeur d'intervalle à 0 !**



```
function hauteur
if Button A0 = 0 then
set intervalle to 0
if Button A1 = 0 then
set intervalle to 7
if Button A2 = 0 then
set intervalle to 12
if Button A3 = 0 then
set intervalle to 19
```

8 Pour jouer la bonne note pour la valeur dans la variable, mets les blocs qui appellent les fonctions **note** et **hauteur** avant le bloc :

9 Transfère ton programme et vois si ça fonctionne !

4 Un petit concert

Utilise ta guitare pour jouer Brille, brille, petite étoile. Si tu as des difficultés à jouer les bonnes notes, essaie de les résoudre en ajustant tes notes et les valeurs de ton photoréflexeur IR !

Do Do Sol Sol La La Sol Fa Fa Mi Mi Ré Ré Do

Sol Sol Fa Fa Mi Mi Ré Sol Sol Fa Fa Mi Mi Ré

Do Do Sol Sol La La Sol Fa Fa Mi Mi Ré Ré Do

Jouer la 1^{ère} mesure

Note	Position	Bouton
Do (60)		
Do (60)		
Sol (67)		
Sol (67)		
La (69)		
La (69)		
Sol (67)		

Apprendre à programmer des robots pour comprendre le monde d'aujourd'hui et de demain.

Les machines programmées, de plus en plus intelligentes, font partie intégrante de notre vie de tous les jours. Elles nous accompagnent, nous entourent et ont envahi tous les domaines de notre vie quotidienne. Maîtriser le monde, ce n'est pas les utiliser, mais avant tout comprendre comment elles fonctionnent.

Comment fonctionnent-elles ?

Selon quelle logique ? Selon quels algorithmes ?

Comment sont conçus les programmes qui leur dictent leurs actions et réactions ?

C'est ce que vous apprendrez tout au long de ces livrets d'apprentissage. Et pas seulement "en théorie" : vous allez vous-même concevoir et programmer vos propres robots : des actions simples aux plus complexes, vous apprendrez à programmer des robots amusants et originaux que vous aurez conçus vous-même. Une seule limite : votre créativité !

L'école Algora permet à tous de s'initier à la programmation en s'amusant, un enjeu majeur, aujourd'hui et demain.



Pour en savoir plus : www.ecolerobots.com