

Studio Scratch de départ

PROJET

FICHES SCRATCH - HORLOGES

LE TEMPS QUI PASSE



DATE

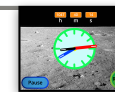
NOVEMBRE 2025

CIBLE

ATELIER CANOPÉ 88



Samuel Chalifour

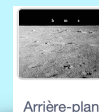


**Créez les variables *secondes*, *minutes* et *heures*.
Faites en sorte que le *Drapeau vert* initialise ces variables à 0. Puis
les *secondes* doivent défiler très vite indéfiniment.**

Événements

Contrôle

Variables



mettre

heures ▼

à



répéter indéfiniment



mettre

secondes ▼

à



mettre

minutes ▼

à



ajouter



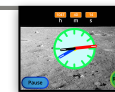
à

secondes ▼

quand



est cliqué

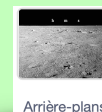


**L'ordre des blocs d'initialisation n'a pas d'importance.
Placez bien les étiquettes des variables au-dessus de
leur nom sur la scène.**

Événements

Contrôle

Variables

quand  est cliqué

mettre secondes ▼ à 0

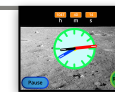
mettre minutes ▼ à 0

mettre heures ▼ à 0

répéter indéfiniment

ajouter 1 à secondes ▼



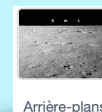


Pour faire avancer les *minutes*, on fait un test *si...alors*. Et on remet les *secondes* à 0. Ce script doit être ajouté au bon endroit dans le script précédent. Maintenant, au *Drapeau vert*, les minutes défilent, moins vite que les secondes.

Contrôle

Opérateurs

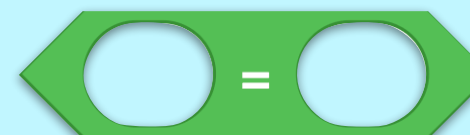
Variables



secondes

si

alors



attendre 1 secondes

mettre

secondes ▼

à

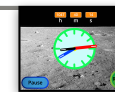


ajouter

1

à

minutes ▼

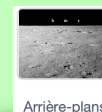
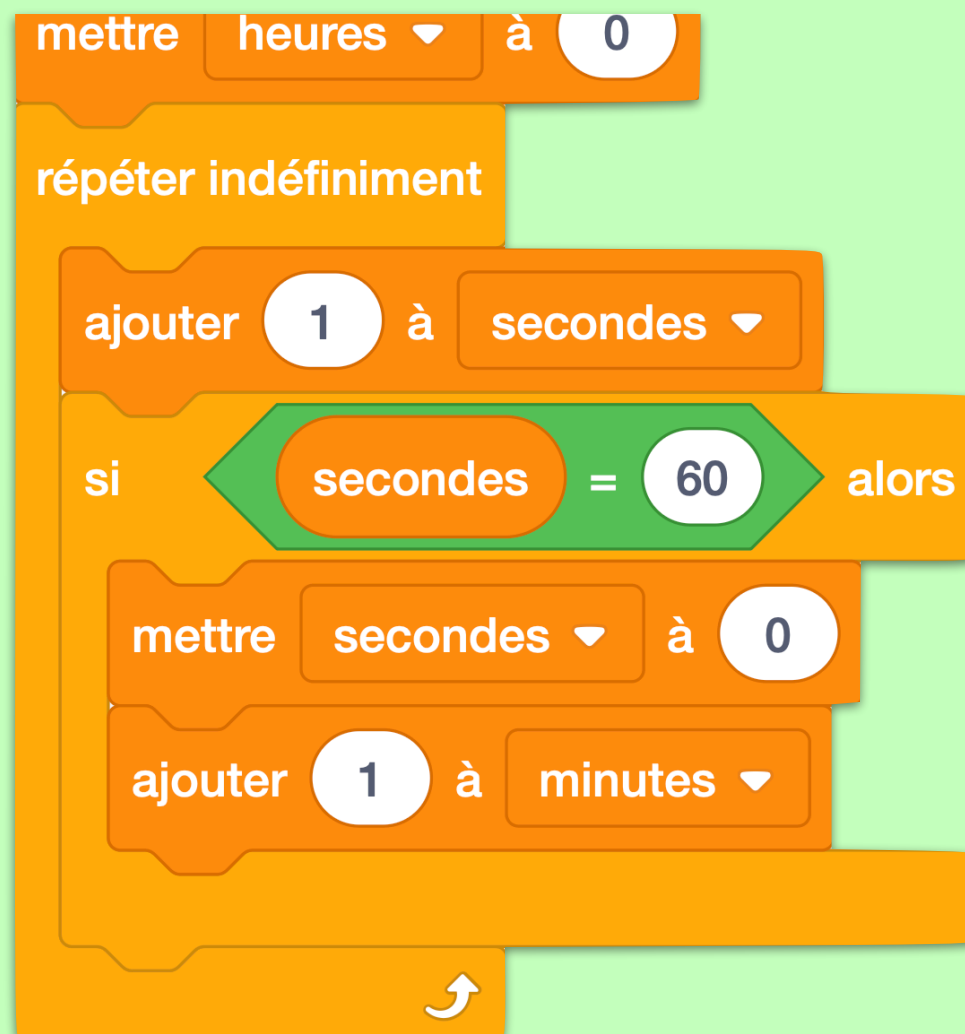


L'important est de faire attention à la condition du *si... alors*, ici (*secondes=60*), et à l'ordre des blocs. Il existe plusieurs solutions, à vous de juger des différences.

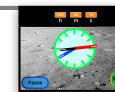
Contrôle

Opérateurs

Variables



Arrière-plans

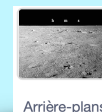


De la même façon, pour faire avancer les *heures*, on fait le même test. Et on remet à zéro les *minutes*. Ce morceau de script doit être ajouté au bon endroit dans le script précédent. On ajoute une instruction d'attente pour être plus au rythme des vraies secondes. On ne doit jamais voir le chiffre 60 s'afficher.

Contrôle

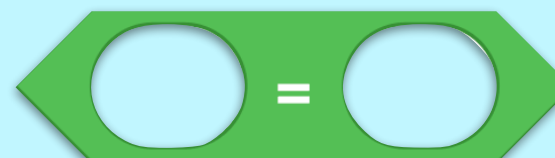
Opérateurs

Variables



si

alors



mettre

minutes ▼

à



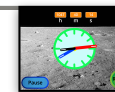
minutes

ajouter



à

heures ▼

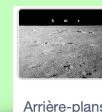
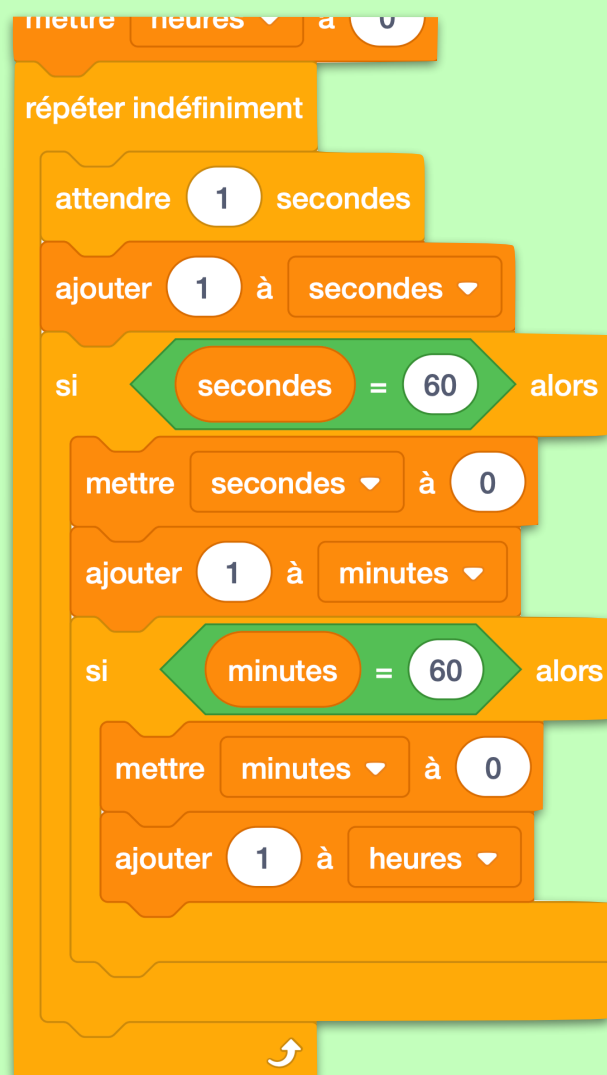


Il s'agit surtout de mettre l'instruction d'attente au bon endroit. Vous devrez adapter la valeur de cette attente pour coller au plus près du bon rythme d'une horloge.

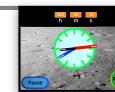
Contrôle

Opérateurs

Variables



Arrière-plans



Faites apparaître le bouton *Reset* qui remet tout à zéro quand on clique dessus. On agit sur son apparence (changement de costume) pour faire croire que le bouton est vraiment appuyé. Puis on le remet en normal une fois qu'il a fait son effet.

Apparence

Événements

Contrôle

Variables

costume suivant

attendre 0.1 secondes



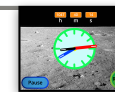
mettre heures ▼ à 0

quand ce sprite est cliqué

costume suivant

mettre secondes ▼ à 0

mettre minutes ▼ à 0



Rien de particulier ici, veuillez à bien placer l'attente entre les deux changements de costume, sinon les calculs des autres blocs sont trop rapides pour être vus à l'oeil humain, l'ordre n'a donc pas d'importance.

Apparence

Événements

Contrôle

Variables

quand ce sprite est cliqué

costume suivant

mettre secondes ▼ à 0

mettre minutes ▼ à 0

mettre heures ▼ à 0

attendre 0.1 secondes

costume suivant





Faites apparaître le bouton *Start*. Au *Drapeau vert*, il passe en costume *Pause*. Quand on clique dessus, s'il est sur *Pause*, il repasse sur le costume *Start* et arrête tout, sinon, il repasse en *Pause* et envoie le message *start*. Le script des *Arrières-plans* doit être séparé en 2 scripts et les nouveaux blocs doivent leur être associés.

Apparence Événements Contrôle Opérateurs

costume suivant costume suivant

envoyer à tous start ▼

si alors

sinon stop tout ▼

Pause

quand ce sprite est cliqué

nom ▼ du costume

Start

basculer sur le costume Pause ▼

quand drapeau vert est cliqué

quand je reçois start ▼

envoyer à tous start ▼

quand drapeau vert est cliqué

mettre secondes à 0

mettre minutes à 0

mettre heures à 0

répéter indéfiniment

ajouter 1 à secondes ▼

si secondes = 60 alors

mettre secondes à 0

ajouter 1 à minutes ▼

si minutes = 60 alors

mettre minutes à 0

ajouter 1 à heures ▼

Arrière-plans



Le bloc *stop tout*, arrête vraiment l'exécution de tous les scripts.
Saurez-vous refaire ce script de *Start* en mieux ? Vous pouvez enlever un bloc *Costume suivant* et utiliser un *si...alors*, plutôt qu'un *si...alors...sinon*.

Apparence Événements Contrôle Opérateurs

Script 1 (Apparence) :

- quand ce sprite est cliqué
- si $\text{nom du costume} = \text{Pause}$ alors
- costume suivant
- stop tout
- sinon
- costume suivant
- envoyer à tous start

Script 2 (Opérateurs) :

- quand est cliqué
- mettre secondes à 0
- mettre minutes à 0
- mettre heures à 0
- envoyer à tous start

Script 3 (Opérateurs) :

- quand je reçois start
- répéter indéfiniment
- ajouter 1 à secondes
- si secondes = 60 alors
- mettre secondes à 0
- ajouter 1 à minutes
- si minutes = 60 alors
- mettre minutes à 0
- ajouter 1 à heures

Script 4 (Apparence) :

- quand est cliqué
- basculer sur le costume Pause



Activez le deuxième arrière-plan de la Scène. Montrez le lutin *Aiguille secondes*. Au *Drapeau vert*, celle-ci doit tourner de 6° toutes les secondes (un tour complet en 360°). À chaque fois qu'elle revient à son point de départ, elle envoie un message pour prévenir qu'on doit ajouter une minute.

Mouvement

Événements

Contrôle

Opérateurs

quand  est cliqué

aller à x: 0 y: 0

s'orienter à 90

attendre 1 secondes

= 90

quand je reçois start ▼

quand je reçois reset ▼

s'orienter à 90

si alors

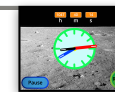
envoyer à tous newMinute ▼

répéter indéfiniment

tourner de 6 degrés

direction





Les boutons Reset et Start doivent aussi fonctionner pour cette aiguille. C'est elle qui va rythmer le déplacement des deux autres aiguilles. Au lieu de calculer soi-même la rotation de 6°, on aurait pu faire le calcul par des blocs : 360 (un tour complet) divisé par 60 (nombre de secondes dans une minutes) :

360 / 6

Mouvement

Événements

Contrôle

Opérateurs






Faites la même chose pour l'*Aiguille minutes*, mais elle n'avance qu'en recevant le message *newMinute*. Quand elle fait un tour complet, elle envoie le message *newHeure*. Trouvez, avec les mêmes blocs, comment doit être programmée l'*Aiguille heures*.


Mouvement Événements Contrôle Opérateurs

Aiguille ... Aiguille h...

quand  est cliqué

aller à x: 0 y: 0

s'orienter à 90

tourner  de 6 degrés

envoyer à tous newheure ▼


quand je reçois reset ▼

s'orienter à 90

quand je reçois newminute ▼

direction

si  alors

 = 90



Pour l'*Aiguille heures*, c'est donc bien plus simple, sauf si vous voulez rajouter le compteur de jours, voire d'années... mais cela deviendra plus complexe avec les années bissextiles du fait que les rotations de la Terre durent en fait 23h56min4sec soit un peu de moins de 24h.

Trouvez d'abord un moyen de mieux rythmer l'horloge puisqu'il y a en fait un décalage dû au fait qu'il y a plus d'une seconde entre chaque attente d'une seconde. Cherchez pas exemple dans les blocs « Capteurs ».

Mouvement

Événements

Contrôle

Opérateurs



Aiguille ...

quand je reçois newminute ▼

tourner ↻ de 6 degrés

si direction = 90 alors

envoyer à tous newheure ▼



Aiguille h...

quand je reçois newheure ▼

tourner ↻ de 6 degrés