

Afficher l'heure avec des aiguilles  
Ou  
Minute ! On n'est pas à la seconde !

Activité réalisée en classe de 4<sup>e</sup>.

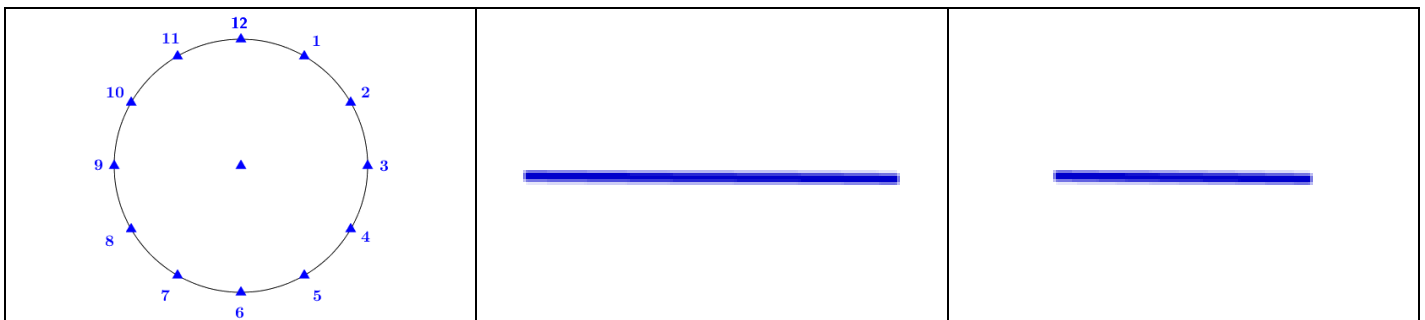
**1<sup>ère</sup> séance** : en salle informatique

L'activité demandée est présentée aux élèves par le biais de la vidéo 'horloge' qui est montrée 2 fois et commentée. Il est indiqué à la classe que l'objectif minimum à atteindre peut être beaucoup plus modeste !

Les élèves ont à leur disposition sur le cahier de texte en ligne 3 images à importer sur Scratch :

- Un cadran
- Une petite aiguille
- Une grande aiguille

Ce sont des images avec transparence au format png pour une importation efficace.



La classe réagit très positivement à ce qui est demandé : le visuel de la vidéo leur convient visiblement, ils trouvent cela sympathique.

Les élèves se lancent alors 'sans réfléchir' : ouverture de Scratch, importation des images pour ceux qui ont leur code pour se connecter à l'ENT, création de 3 lutins un peu plus artisanaux pour les autres.

Une certaine euphorie s'installe au démarrage et retombe assez vite dès lors qu'il faut commencer à déplacer en emboîter des blocs pour programmer. Les élèves essaient d'abord seuls, puis demandent à leurs voisins comment ils voient les choses mais personne n'aboutit.

Tout cela a pris environ 25 à 30 minutes. Un temps de bilan collectif est alors demandé.

Peu à peu, il ressort que sous un aspect simple (la consigne : « afficher l'heure » est bien comprise par tous), il y a en réalité une vraie complexité dans la programmation demandée. Cette programmation ne peut donc pas être traitée *frontalement*.

Là encore, le dialogue met en évidence des « passages obligés », qui deviendront des sous-programmes :

- Bien positionner le cadran, bien positionner chacune des aiguilles : l'initialisation
- Etre capable de faire tourner juste une aiguille (minute ou heure)
- Ceci induit de ne demander à l'utilisateur qu'une seule information (minute ou heure)
- Puis recommencer avec l'autre aiguille
- Ceci amène une temporalité : un message doit définir le moment de la deuxième question.

On se met d'accord sur un résultat minimum qui correspondra à un 'programme réussi'.

Les élèves retournent programmer et certains parviennent à démarrer quelque chose juste avant la fin de cette première séance.

**2<sup>ème</sup> séance** : en classe (15 minutes)

La feuille donnée en [annexe](#) est lue et analysée en classe avec les élèves. Elle fait le point *à froid* de la stratégie qui a été mise en place la veille. Les points en caractères gras sont présentés comme essentiels :

- **Analyser la difficulté et le degré de complexité de ce qui est demandé.**
- **Décomposer le problème en sous-problèmes afin de structurer le programme.**

Le dernier point marqué d'une flèche est présenté comme possible dès lors que les deux autres objectifs précédents sont atteints :

- Chercher des évolutions permettant d'améliorer le programme

Cette feuille est une trace écrite à destination du cahier de cours.

Le reste de la séance est un travail sur la rotation avec papier et crayon.

**3<sup>ème</sup> séance** : en salle informatique

L'objectif donné à tous est de réussir à décomposer le programme en sous-programme pour faire tourner convenablement les deux aiguilles.

Cela donne des programmes basiques comme [élève 1](#) (très minimaliste), [élève 2](#) ou [élève 3](#).

Cela donne aussi des programmes en voie d'amélioration, mais avec encore des petites erreurs :

[Elève 4](#) ou [élève 5](#) : tentative pour que la petite aiguille progresse entre deux valeurs chiffrées proportionnellement au nombre de minutes. La proportionnalité n'est pas encore convenablement gérée.

Quelques élèves, deux ou trois, ont abouti à un programme avec une amélioration (sur la position de l'aiguille des heures : [élève 6](#)).

## Réaliser une montre sur Scratch

### 1) Découverte de l'activité

L'activité consiste à utiliser 3 lutins : un lutin cadran, un lutin « grande aiguille », un lutin « petite aiguille ». Lorsque l'heure est indiquée par l'utilisateur, les aiguilles doivent tourner de façon visible pour se positionner à l'heure indiquée.

#### ➔ Analyser la difficulté et le degré de complexité de ce qui est demandé.

Cela présente plusieurs difficultés. Par exemple : comment récupérer l'heure ? Sous quel format ? Comment faire tourner les aiguilles ? Les aiguilles doivent-elles avoir des spécificités pour que la rotation soit correcte ? ...

### 2) Analyse des premières difficultés et début de la programmation

En étant confronté à l'ensemble du programme, on se rend compte qu'il est très délicat, voire très difficile, de le mener à bien de façon immédiate et simple.

#### ➔ Décomposer le problème en sous-problèmes afin de structurer le programme.

Cela permet :

- De démarrer l'activité
- D'avancer peu à peu en améliorant au fur et à mesure des étapes
- De comprendre en profondeur ce qui est demandé

Il est convenu de commencer par :

- Importer les 3 lutins et réaliser l'initialisation, c'est-à-dire programmer la position initiale de chaque lutin afin de pouvoir démarrer le programme
- Ne s'occuper que d'une aiguille à la fois dans un premier temps
- Demander l'heure du moment en séparant en deux temps : heures d'une part, minutes d'autre part.

### 3) Perspective

Une fois le programme commencé, élément par élément, on peut chercher des pistes d'amélioration.

#### ➔ Chercher des évolutions permettant d'améliorer le programme

Deux pistes d'amélioration sont envisagées :

- Obtenir un visuel dynamique qui donnera plus d'esthétique au rendu
- Anticiper sur des erreurs de manipulation de l'utilisateur pour que le programme fonctionne malgré tout.


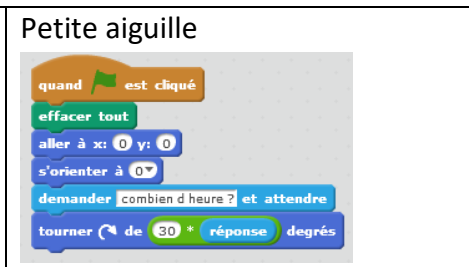
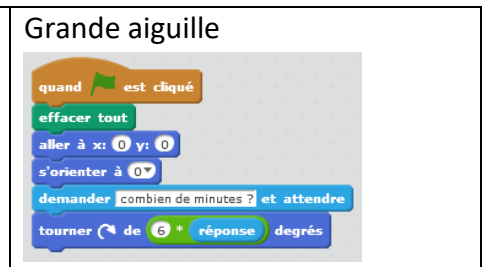
Elève\_1

Deux aiguilles simplement, et pas de cadran.

<p>Petite aiguille</p> 	<p>Grande aiguille</p> 
---	---

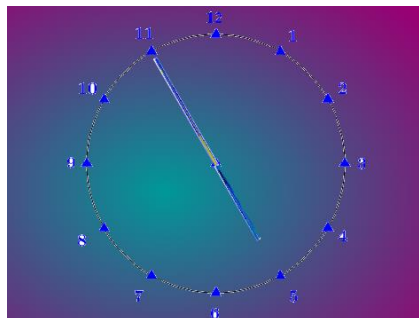
Elève\_2

Par rapport au programme précédent, il y a un cadran...dont la programmation est très simple.

<p>Cadran</p> 	<p>Petite aiguille</p> 	<p>Grande aiguille</p> 
--	---	--


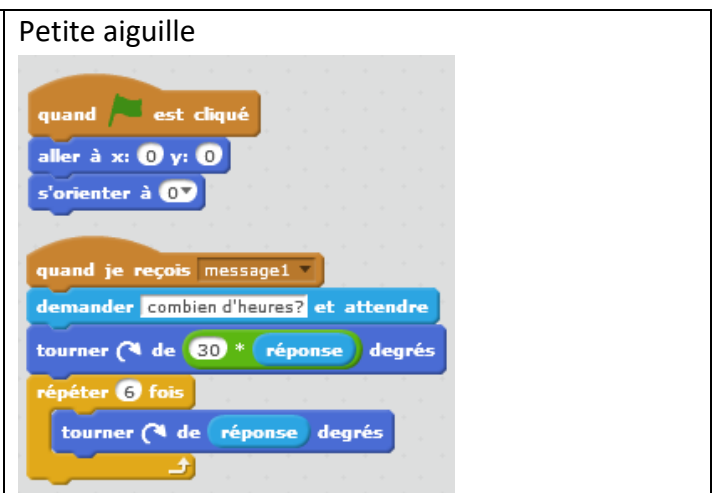
Elève\_3

On retrouve le même principe, avec un aspect 'décoratif' du fond de scène :



Elève\_4

Le cadran est positionné 'à la main' sans programmation. Les minutes sont ensuite demandées avec un message pour lancer la demande des heures.

<p>Grande aiguille</p> 	<p>Petite aiguille</p> 
---	---

## Elève\_5

La programmation est semblable à celle d'élève\_5. Il y a cependant une tentative d'évolution : faire en sorte que la petite aiguille se positionne différemment selon que l'on demande par exemple 15h05 et 15h50.

Cette tentative en revanche n'est pas aboutie totalement et présente une erreur.



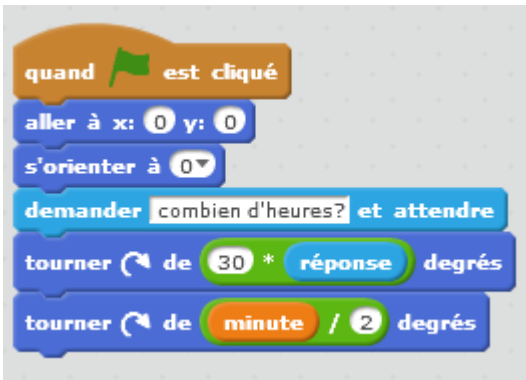
## Elève\_6

Le cadran est programmé de façon très simple :



Et le travail sur un positionnement optimisé de la petite aiguille est abouti

### Grande aiguille



### Petite aiguille



Il n'y a cependant pas d'envoi de message pour gérer l'ordre des questions : c'est le programme qui le fait...en demandant d'abord les minutes.