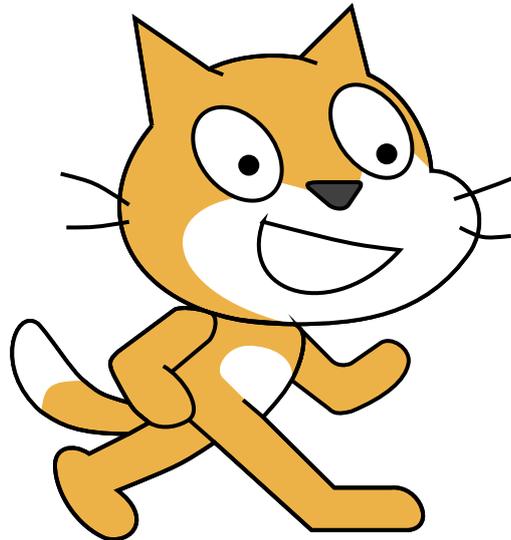
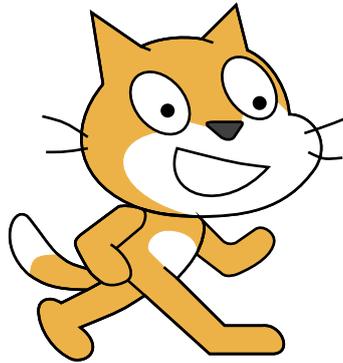


Guide d'utilisation du paquet [ScratchX.sty](#)

Thibault Ralet

mars 2017 ▼



## 1 Introduction

Le paquet `ScratchX.sty` permet d'écrire n'importe quel programme Scratch en L<sup>A</sup>T<sub>E</sub>X.

Le paquet `ScratchX.sty` doit être mis dans le même dossier que celui dans lequel le fichier `.tex` est créé, ou dans le dossier dédié aux packages de l'éditeur de fichiers `.tex`.

Le paquet `ScratchX.sty` doit être appelé dans le préambule du fichier `.tex` par :

```
\usepackage{ScratchX}
```

Le paquet `ScratchX.sty` utilise les paquets suivants :

```
\usepackage[nomessages]{fp}  
\usepackage{calc}  
\usepackage{xstring}  
\usepackage[alpine]{ifsym}%pour avoir VarFlag comme  
%drapeau de départ de Scratch  
\usepackage{ifthen}  
\usepackage{multido}  
\usepackage{xargs}
```

Dans le document `.tex` créé, il faut charger :

```
\usepackage{tikz}  
\usetikzlibrary{calc}
```

et aussi :

```
\usepackage{amssymb}
```

La compilation peut s'effectuer en XeLaTeX ou en pdfLaTeX.

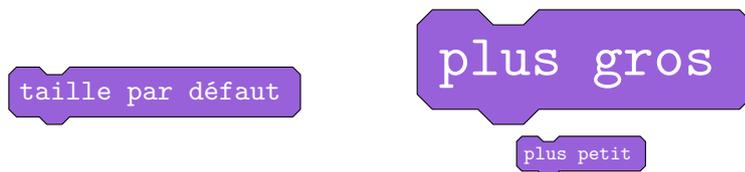
## 2 Description générale

On écrit un programme (ou simplement une commande) Scratch à l'aide de l'environnement :

```
\begin{Scratch}  
%commandes LaTeX pour créer les commandes Scratch  
\end{Scratch}
```

La commande `\begin{Scratch}` admet en option l'échelle (1 par défaut).

Ainsi, `\begin{Scratch}[2]` double la taille du programme, alors que `\begin{Scratch}[0.7]` réduit de 70% la taille du programme créé.



### Les couleurs

Les dix couleurs spécifiques à Scratch sont définies ainsi :

mvt :	Mouvement	evt :	Événements
app :	Apparence	ctrl :	Contrôle
son :	Sons	capt :	Capteurs
stylo :	Stylo	ope :	Opérateurs
data :	Données	bloc :	Ajouter blocs

### 3 Liste des commandes

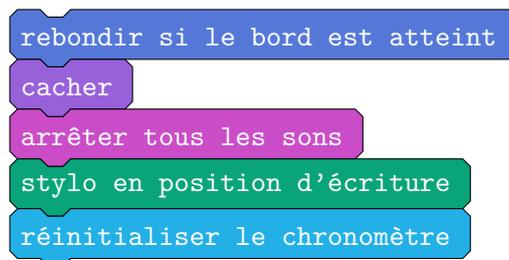
#### 3.1 Commandes Scratch simples

On les obtient avec `\sbox{<texte>}{<couleur>}`.

En tapant :

```
\begin{Scratch}
\sbox{rebondir si le bord est atteint}{mvt}
\sbox{cacher}{app}
\sbox{arrêter tous les sons}{son}
\sbox{stylo en position d'écriture}{stylo}
\sbox{réinitialiser le chronomètre}{capt}
\end{Scratch}
```

on obtient :



#### 3.2 Commandes Scratch spéciales

##### 3.2.1 Événement

On les obtient avec `\beginbox{<texte>}`.

`\beginbox{}` produit

`\beginbox{clone}` produit

`\beginbox{quand ce lutin est cliqué}` produit

##### 3.2.2 Tourner

On les obtient avec `\turnbox{<orientation>}{<angle>}`.

`\turnbox{}{-145}` produit 

`\turnbox{gauche}{30}` produit 

On peut également écrire `\turnbox{g}{30}` ou `\turnbox{G}{30}`

### 3.2.3 Boucles

On les obtient avec

`\boucle{<texte>}{<nombre de blocs à l'intérieur>}{<type>}`,  
où `<type>` est un entier égal à 1 (*répéter  $x$  fois* ou *répéter jusqu'à*) ou  
-1 (*répéter indéfiniment*).

`\boucle{répéter  $\$x$  fois}{2}{1}`  
`\sbox{deux blocs}{red}`  
`\sbox{dans la boucle}{gray}`

produit



`\boucle{répéter indéfiniment}{3}{-1}`  
`\sbox{cette fois-ci}{pink}`  
`\sbox{trois blocs}{blue}`  
`\sbox{dans la boucle}{purple}`



*Attention, lorsque l'on souhaite placer des boucles dans des boucles, il faut repenser le nombre de blocs de la boucle principale ! En effet, une boucle compte pour deux blocs (sans les blocs contenus à l'intérieur).*

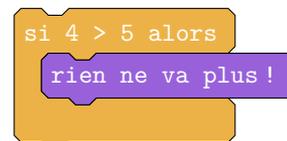
### 3.2.4 Si ... Alors

*Attention, on obtient cette commande Scratch avec la même syntaxe que précédemment, à savoir :*

`\boucle{<texte>}{<nombre de blocs à l'intérieur>}{2}`.

`\boucle{si 4 > 5 alors}{1}{2}`  
`\sbox{rien ne va plus !}{app}`

produit



Remarque : pour connaître comment bien écrire le test du si, voir *les petites boîtes* à la section 3.3.

### 3.2.5 Si ... Alors ... Sinon

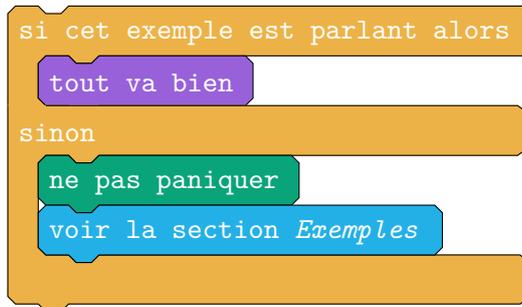
On les obtient avec

```
\sailors{<text>}{<nombre de blocs à l'intérieur>}
et
\simenon{<nombre de blocs à l'intérieur>}
```

En tapant :

```
\begin{Scratch}
\sailors{si cet exemple est parlant alors}{1}
\sabox{tout va bien}{app}
\simenon{2}
\sabox{ne pas paniquer}{stylo}
\sabox{voir la section \emph{Exemples}}{capt}
\end{Scratch}
```

on obtient :



### 3.2.6 Ajouter blocs

On les obtient avec `\bloibox{<text>}`.

`\bloibox{triangle}` produit 

### 3.2.7 Contrôle spécial

Cela concerne   et  .  
Ces commandes s'obtiennent avec `\kbox{<text>}`.

## 3.3 À l'intérieur des commandes Scratch : les petites boîtes

Comment obtenir certaines commandes plus évoluées, comme par exemple :

  ?

Comment écrire  ?

ou  ? ou encore

 ?

### 3.3.1 Les petites boîtes rectangulaires

- Dans les `\scbox` :  
on les obtient avec  
`\rb[<couleur>]{<texte>}`
- dans les `\beginbox` :  
on les obtient avec  
`\rbb[<couleur>]{<texte>}`

Dans les deux cas, `<couleur>` a par défaut la couleur de la boîte dans laquelle il est inscrit. Pour obtenir une boîte rectangulaire blanche, il suffit de mettre `<couleur>` à `white` ou `w`.

En tapant :

```
\begin{Scratch}
\scbox{jouer le son \rb{miaou}}{son}
\scbox{penser à \rb[white]{Hmm\dots}}{app}
\scbox{demander \rb[w]{What's your name?} et attendre}{capt}
\end{Scratch}
```

on obtient :



### 3.3.2 Les petites boîtes circulaires

- Elles sont en creux ou en relief.
- On les obtient avec  
`\cb[<couleur>]{<texte>}`

Par défaut, `<couleur>` est de la couleur de la boîte dans laquelle la boîte circulaire est insérée. Si `<couleur>` est `white` ou `w`, la boîte circulaire est en creux.

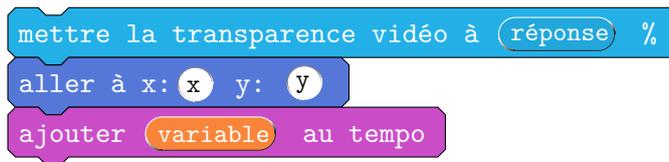
En tapant :

```

\begin{Scratch}
\sbox{mettre la transparence vidéo à \cb{réponse} \%}{capt}
\sbox{aller à x:\cb[w]{x} y:\cb[white]{y}}{mvt}
\sbox{ajouter \cb[data]{variable} au tempo}{son}
\end{Scratch}

```

on obtient :



### 3.3.3 Les petites boîtes hexagonales

Elles ne concernent que les commandes *Capteurs* et *Opérateurs*.

On les obtient avec

```
\hb[<couleur>]{<texte>}
```

Par défaut, <couleur> est ope.

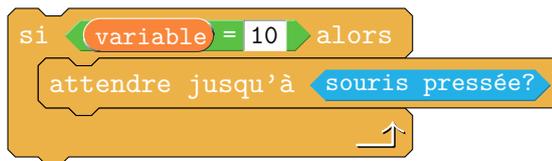
En tapant :

```

\begin{Scratch}
\boucle{si \hb{\cb[data]{variable}=\rb[w]{10}} alors}{1}{1}
\sbox{attendre jusqu'à \hb[capt]{souris pressée?}}{ctrl}
\end{Scratch}

```

on obtient :



### 3.3.4 Les petites boîtes carrées

Elles ne concernent que les carrés de couleur.

On les obtient avec

```
\sqb{<couleur>}
```

En tapant :

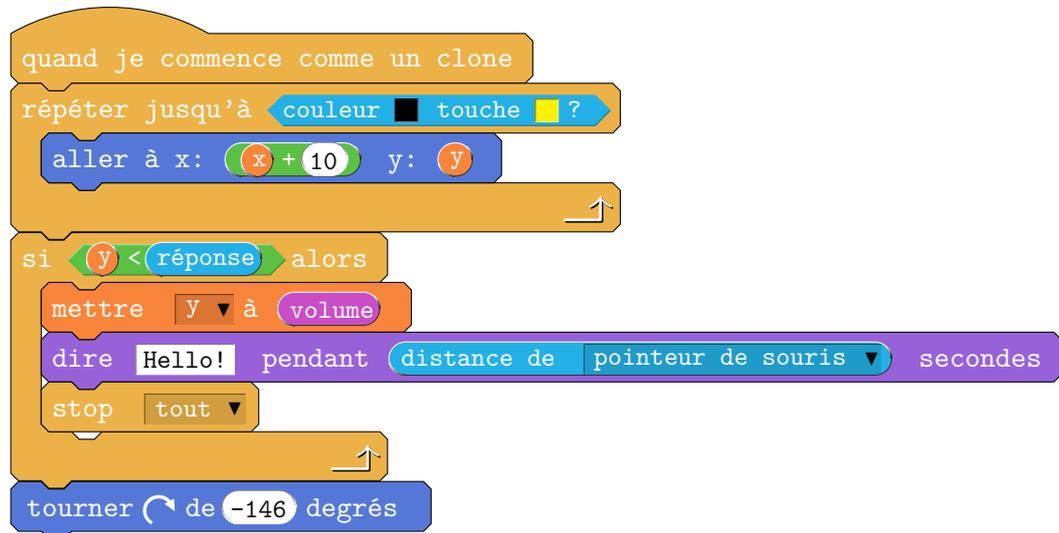
```

\begin{Scratch}
\sbox{mettre la couleur du stylo à \sqb{brown}}{stylo}
\end{Scratch}

```

on obtient : 

### 3.4 Des commandes plus riches



Obtenu avec :

```
\begin{Scratch}
\beginbox{clone}
\boucle{répéter jusqu'à \hb[cap]{couleur \sqb{black}
touche \sqb{yellow}? }}{1}{1}
\scbox{aller à x: \cb[ope]{\cb[data]{x}+\cb[w]{10}} y: \cb[data]{y}}{mvt}
\boucle{si \hb{\cb[data]{y}<\cb[cap]{réponse}} alors}{3}{1}
\scbox{mettre \rb{y} à \cb[son]{volume}}{data}
\scbox{dire \rb[w]{Hello!} pendant \cb[cap]{distance de
\rb{pointeur de souris}} secondes}{app}
\kbox{stop \rb{tout}}
\turnbox{2}{-146}
\end{Scratch}
```

### 3.5 Autre type de commandes

#### 3.5.1 Dans les boucles

La commande `\blank` ne s'emploie (artificiellement) que lorsque dans le programme Scratch, deux boucles se terminent l'une après l'autre.

En tapant :

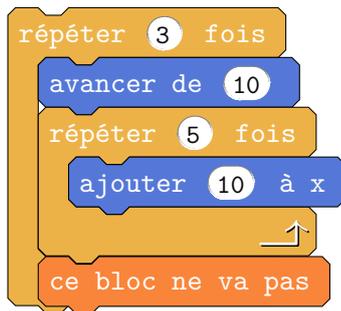
```
\begin{Scratch}
```

```

\boucle{répéter \cb[w]{3} fois}{4}{1}
\sbox{avancer de \cb[w]{10}}{mvt}
\boucle{répéter \cb[w]{5} fois}{1}{1}
\sbox{ajouter \cb[w]{10} à x}{mvt}
\sbox{ce bloc ne va pas}{data}
\end{Scratch}

```

on obtient :



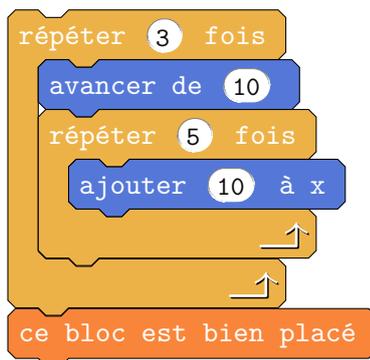
Alors qu'en tapant :

```

\begin{Scratch}
\boucle{répéter \cb[w]{3} fois}{4}{1}
\sbox{avancer de \cb[w]{10}}{mvt}
\boucle{répéter \cb[w]{5} fois}{1}{1}
\sbox{ajouter \cb[w]{10} à x}{mvt}
\blank
\sbox{ce bloc est bien placé}{data}
\end{Scratch}

```

on obtient :



Noter que si l'on tape :

```

\begin{Scratch}

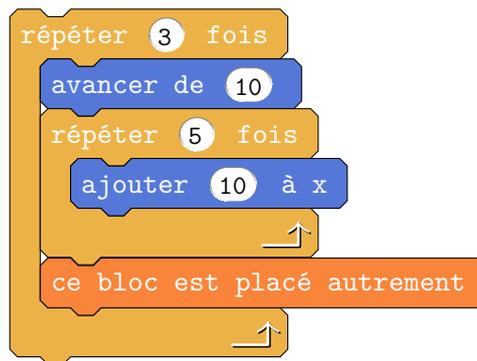
```

```

\boucle{répéter \cb[w]{3} fois}{5}{1}
\sbox{avancer de \cb[w]{10}}{mvt}
\boucle{répéter \cb[w]{5} fois}{1}{1}
\sbox{ajouter \cb[w]{10} à x}{mvt}
\sbox{ce bloc est placé autrement}{data}
\end{Scratch}

```

on obtient :



### 3.5.2 Dessiner le chat

Le chat de la couverture est obtenu en appelant la commande :

```
\Scratchy[<échelle>][<largeur des lignes>]
```

Par défaut, l'échelle est 0.25 et la largeur des lignes est fixée à 0.25 pt.

Voici le code écrit pour la couverture :

```
\parbox{1ex}{\Scratchy[0.1][0.1]}\
```

```
\hspace{1cm}\parbox{1ex}{\Scratchy[0.2][0.2]}\
```

```
\hspace{2cm}\parbox{1ex}{\Scratchy[0.5][1]}\
```

```
\hspace{5cm}\parbox{1ex}{\Scratchy[0.75][1.5]}\
```

## 4 Problèmes connus et solutions

1. On n'a pas le petit triangle dans la commande *s'orienter à* (mouvement). Il faut le placer à la main.

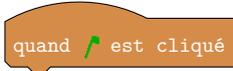
```
\sbox{s'orienter à \cb[w]{90 \scriptsize$\blacktriangledown$}}{mvt}
```



2. La hauteur des boîtes est fixée. Ainsi, on ne peut pas imbriquer beaucoup de sous-commandes dans des commandes Scratch.
3. Lorsque l'on insère une seule commande Scratch dans du texte, elle n'est pas centrée verticalement. On peut la réajuster avec un `\raisebox{-3mm}`.
4. Il y a un également un décalage horizontal. À la fin d'un environnement Scratch, il faut souvent ajouter un `\hspace{-1cm}`.
5. Le temps de compilation est parfois assez long.

## 5 Résumé des commandes

`\beginbox{}`



`\beginbox{<texte>}`  
(quand ce lutin est cliqué)



`\beginbox{clone}`



`\blocbox{<texte>}`



`\turnbox{}{90}`



`\turnbox{g}{-270}`  
(ou <gauche> ou <G> ou <g>)



`\scbox{<texte>}{<couleur>}`



`\boucle{répéter}{2}{1}`  
({<texte>}{<nbr blocs>}{<type>})



`\boucle{répéter indéfiniment}{1}{-1}`  
({<texte>}{<nbr blocs>}{<type>})



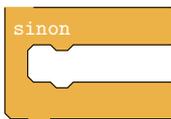
`\boucle{si ...alors}{1}{2}`  
({<texte>}{<nbr blocs>}{<type>})



`\sailors{si...alors...}{2}`  
({<texte>}{<nbr blocs>})

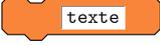


`\simenon{<nbre blocs>}`  
(ici, `\simenon{1}`)



`\kbox{<texte>}`

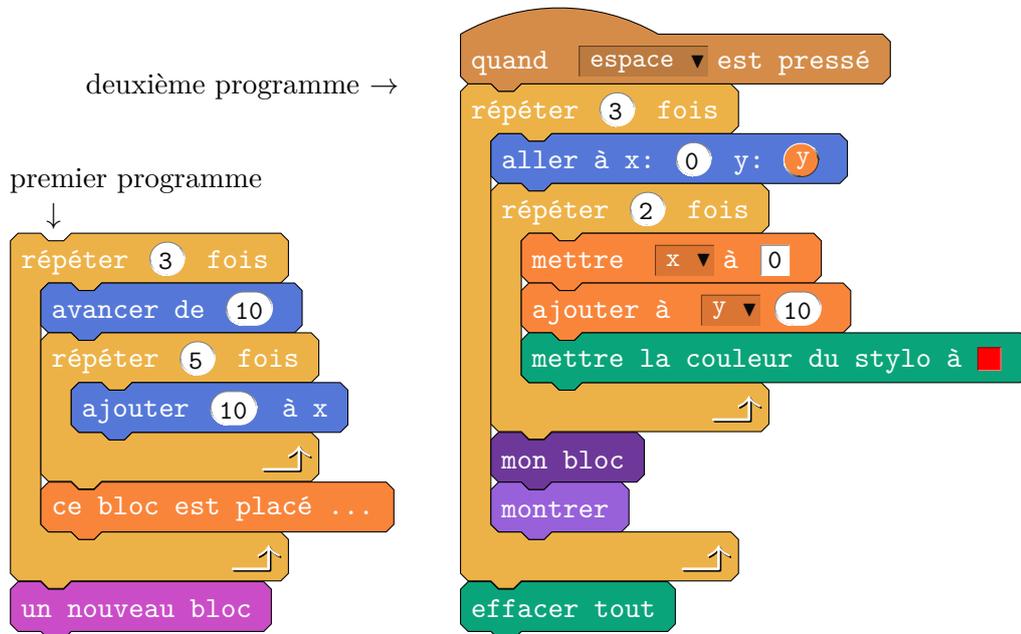


<code>\rb{&lt;texte&gt;}</code> ( <code>\rb{variable}</code> )	
<code>\rb[w]{&lt;texte&gt;}</code> (ou <code>\rb[white]{&lt;texte&gt;}</code> )	
<code>\rbb{&lt;texte&gt;}</code> (uniquement pour <code>\beginbox</code> ) <code>\beginbox{quand je reçois \rbb{message1}}</code>	
<code>\cb{&lt;texte&gt;}</code> (transparent)	
<code>\cb[w]{&lt;texte&gt;}</code> (ou <code>\cb[white]{&lt;texte&gt;}</code> )	
<code>\cb[&lt;couleur&gt;]{&lt;texte&gt;}</code> <code>\cb[ope]{réponse}</code>	
<code>\hb{&lt;texte&gt;}</code>	
<code>\hb[capit]{&lt;texte&gt;}</code>	
<code>\sqb{&lt;couleur&gt;}</code>	

## 6 Exemples de programme

```
\begin{Scratch}
\boucle{répéter \cb[w]{3} fois}{5}{1}
\sbox{avancer de \cb[w]{10}}{mvt}
\boucle{répéter \cb[w]{5} fois}{1}{1}
\sbox{ajouter \cb[w]{10} à x}{mvt}
\sbox{ce bloc est placé autrement}{data}
\sbox{un nouveau bloc}{son}
\end{Scratch}
```

```
\begin{Scratch}
\beginbox{quand \rbb{espace} est pressé}
\boucle{répéter \cb[w]{3} fois}{8}{1}
\sbox{aller à x: \cb[w]{0} y: \cb[data]{y}}{mvt}
\boucle{répéter \cb[w]{2} fois}{3}{1}
\sbox{mettre \rb{x} à \rb[w]{0}}{data}
\sbox{ajouter à \rb{y} \cb[w]{10}}{data}
\sbox{mettre la couleur du stylo à \sqb{red}}{stylo}
\sbox{mon bloc}{bloc}
\sbox{montrer}{app}
\sbox{effacer tout}{stylo}
\end{Scratch}
```



## 6.1 Des boucles imbriquées

```
\begin{Scratch}
\beginbox{quand \rb{chronomètre} > \cb[w]{10}}

\sailors{si \hb[cap]{touche \rb{espace} pressée?} alors}{12}

\sailors{si \hb[cap]{souris pressée?} alors}{1}
\sabox{aller à x: \cb[w]{0} y: \cb[w]{0}}{mvt}

\simenon{2}
\sabox{donner la valeur \cb[w]{0} à x}{mvt}
\sabox{ajouter \cb[w]{10} à y}{mvt}

\sailors{si couleur \sqb{son} touchée?}{1}
\sabox{aller à x: \cb[w]{0} y: \cb[w]{0}}{mvt}

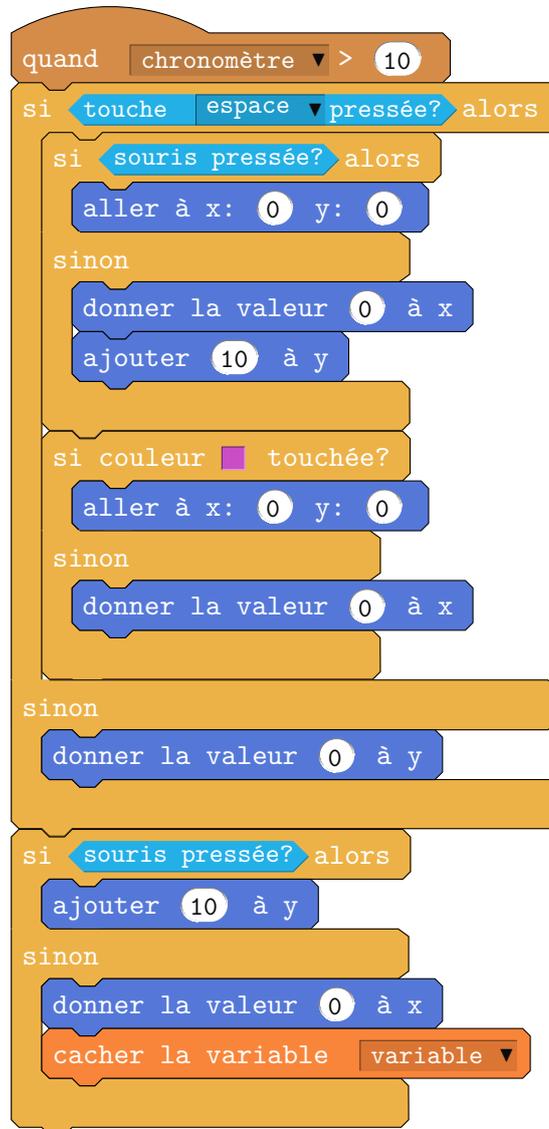
\simenon{1}
\sabox{donner la valeur \cb[w]{0} à x}{mvt}

\simenon{1}
\sabox{donner la valeur \cb[w]{0} à y}{mvt}

\sailors{si \hb[cap]{souris pressée?} alors}{1}
\sabox{ajouter \cb[w]{10} à y}{mvt}

\simenon{2}
\sabox{donner la valeur \cb[w]{0} à x}{mvt}
\sabox{cacher la variable \rb{variable}}{data}

\end{Scratch}
```



```
\begin{Scratch}
\beginbox{}

\sailors{si couleur \sqb{stylo} touchée?}{12}

\sailors{si \hb[capt]{souris pressée?} alors}{6}
\scbox{aller à x: \cb[w]{0} y: \cb[w]{0}}{mvt}

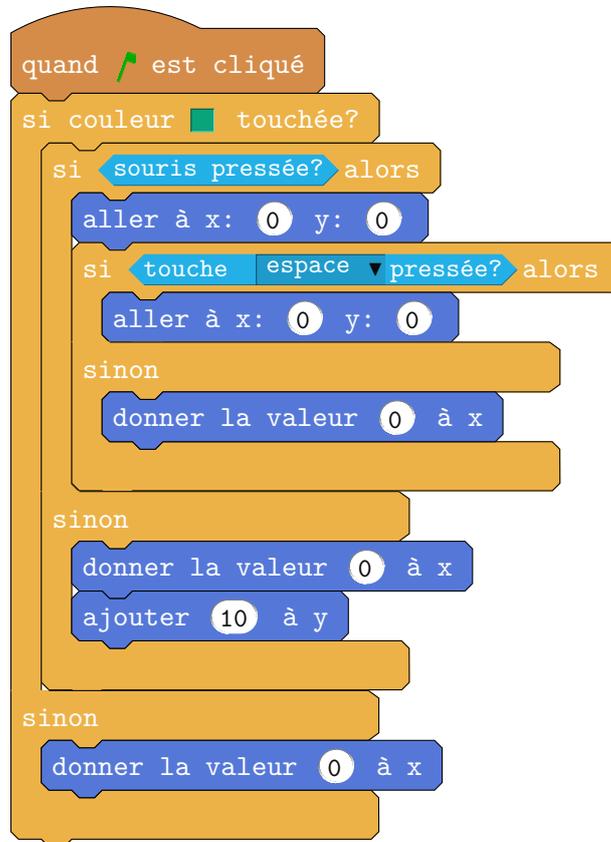
\sailors{si \hb[capt]{touche \rb{espace} pressée?} alors}{1}
\scbox{aller à x: \cb[w]{0} y: \cb[w]{0}}{mvt}

\simenon{1}
\scbox{donner la valeur \cb[w]{0} à x}{mvt}

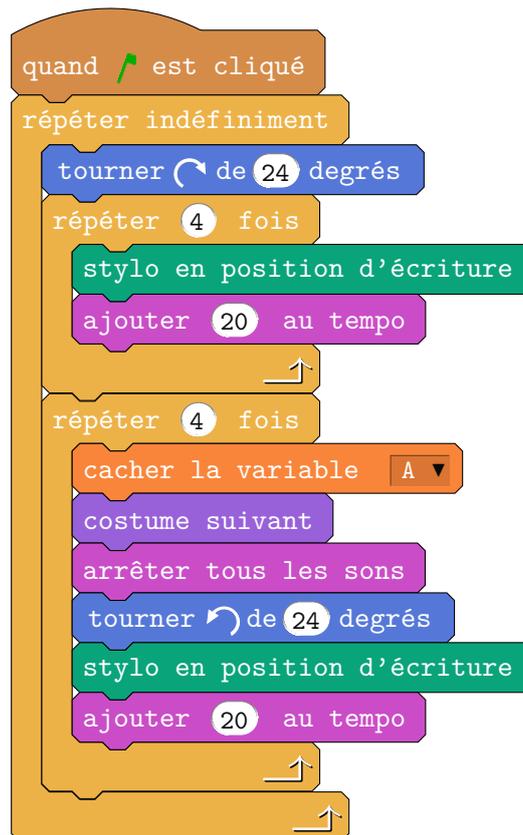
\simenon{2}
\scbox{donner la valeur \cb[w]{0} à x}{mvt}
\scbox{ajouter \cb[w]{10} à y}{mvt}

\simenon{1}
\scbox{donner la valeur \cb[w]{0} à x}{mvt}

\end{Scratch}
```



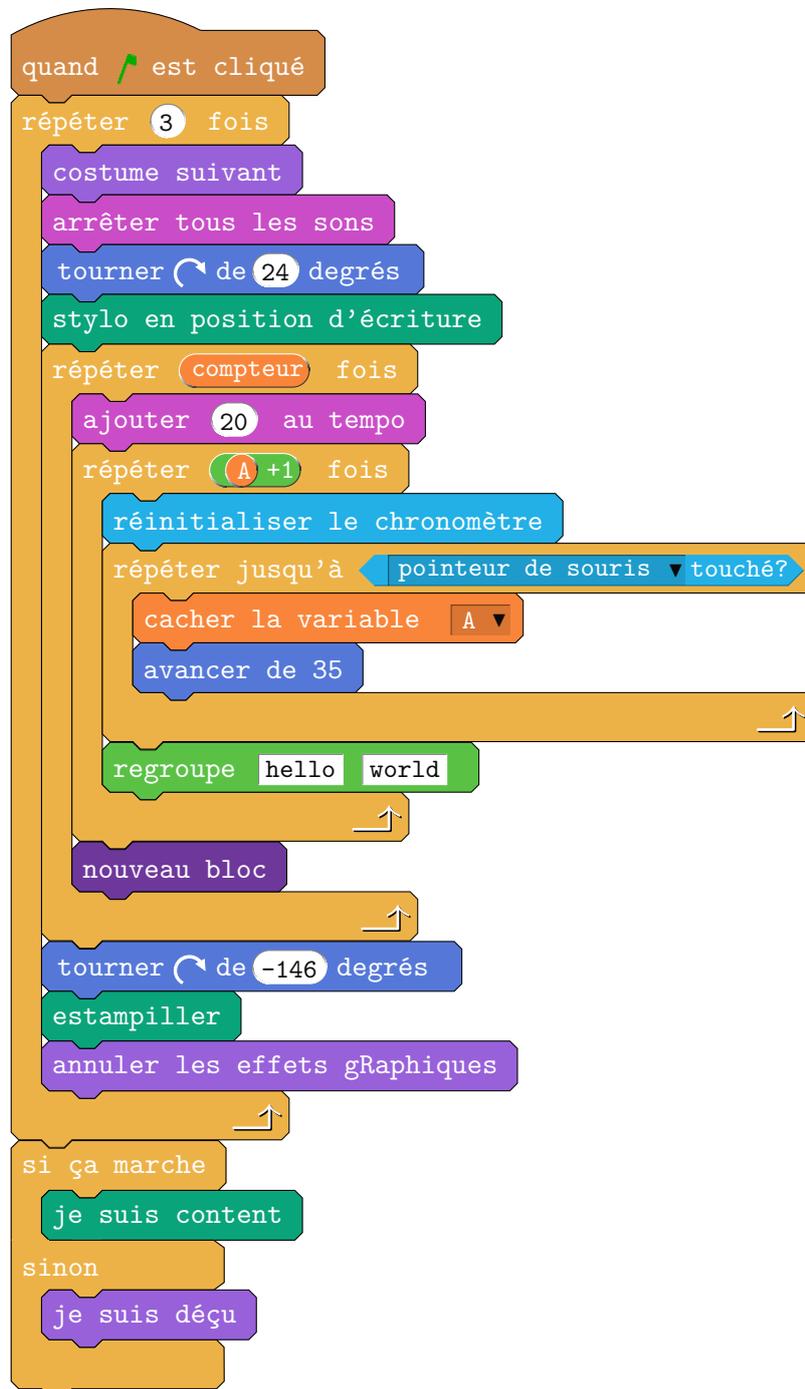
```
\begin{Scratch}  
\beginbox{}  
\boucle{répéter indéfiniment}{13}{-1}  
\turnbox{}{24}  
\boucle{répéter \cb[w]{4} fois}{2}{1}  
\sbox{stylo en position d'écriture}{stylo}  
\sbox{ajouter \cb[w]{20} au tempo}{son}  
\boucle{quand le lutin s'en va}{6}{1}  
\sbox{cacher la variable \rb{A}}{data}  
\sbox{costume suivant}{app}  
\sbox{arrêter tous les sons}{son}  
\turnbox{g}{24}  
\sbox{stylo en position d'écriture}{stylo}  
\sbox{ajouter \cb[w]{20} au tempo}{son}  
\blank  
\end{Scratch}
```



```
\begin{Scratch}  
\beginbox{quand on le veut}  
\boucle{répéter un certain nombre de fois}{8}{1}  
\sbox{aller à x: \cb[w]{0} y: \cb[w]{0}}{mvt}  
\boucle{le dernier??}{2}{1}  
\sbox{on peut faire}{gray}  
\sbox{ce que l'on veut}{black}  
\boucle{ne pas répéter}{1}{1}  
\sbox{faux bloc}{brown}  
\blank  
\sbox{dernier bloc}{pink}  
\end{Scratch}
```



```
\begin{Scratch}
\beginbox{}
\boucle{répéter \cb[w]{3} fois}{19}{1}
\scbox{costume suivant}{app}
\scbox{arrêter tous les sons}{son}
\turnbox{1}{24}
\scbox{stylo en position d'écriture}{stylo}
\boucle{répéter \cb[data]{compteur} fois}{10}{1}
\scbox{ajouter \cb[w]{20} au tempo}{son}
\boucle{répéter \cb[ope]{\cb[data]{A}+1} fois}{6}{1}
\scbox{réinitialiser le chronomètre}{capt}
\boucle{répéter jusqu'à \hb[capt]{\rb{pointeur de souris} touché?}}{2}{1}
\scbox{cacher la variable \rb{A}}{data}
\scbox{avancer de 35}{mvt}
\scbox{regroupe \rb[w]{hello}\rb[w]{world}}{ope}
\scbox{nouveau bloc}{bloc}
\turnbox{2}{-146}
\scbox{estampiller}{stylo}
\scbox{annuler les effets graphiques}{app}
\sailors{si ça marche}{1}
\scbox{je suis content}{stylo}
\simenon{1}
\scbox{je suis déçu}{app}
\end{Scratch}
```



## 7 Conclusion

Pour tout suggestion, remarque ou commentaire :  
Thibault.Ralet@ac-clermont.fr.