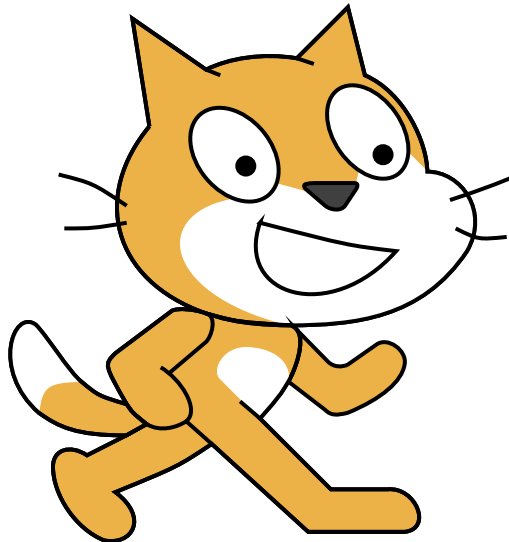
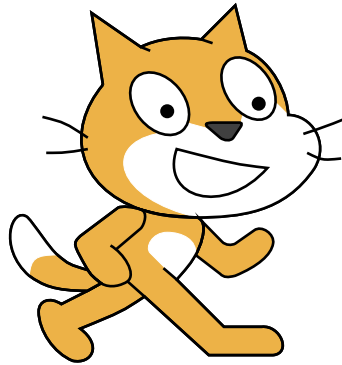


User guide: package [ScratchX.sty](#)

Thibault Ralet

March 2017 ▼



1 Introduction

The **ScratchX.sty** package allows you to write any kind of Scratch program in *L^AT_EX*.

The **ScratchX.sty** package has to be put at least in the same file where the document **.tex** is created, or in the dedicated place in your computer.

The **ScratchX.sty** package must be called in the preamble of the document **.tex** with:

```
\usepackage{ScratchX}
```

The **ScratchX.sty** package uses the following packages:

```
\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}
```

In the document **.tex** that you want to create, you need to load:

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

and also:

```
\usepackage{amssymb}
```

You can compile with XeLaTeX or pdfLaTeX.

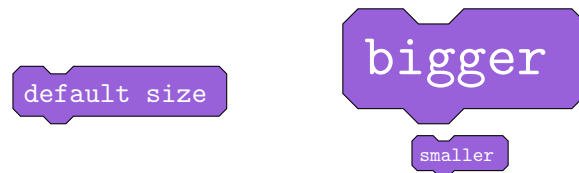
2 General description

We write a program (or simply a command) Scratch with the environment:

```
\begin{Scratch}
%\LaTeX commands in order to create the Scratch commands
\end{Scratch}
```

The command `\begin{Scratch}` has the scale in option (1 by default).

Thus, `\begin{Scratch}[2]` doubles the program size, whereas `\begin{Scratch}[0.7]` reduces by 70% the size of the created program.



The colors

The ten specific Scratch colors are defined like this:

mvt :	Motion	evt :	Events
app :	Looks	ctrl :	Control
son :	Sound	capt :	Sensing
stylo :	Pen	ope :	Operators
data :	Data	bloc :	More blocks

3 Command list

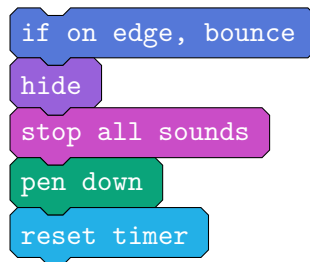
3.1 Simples

Obtained with `\scbox{<text>}{<color>}`.

Typing:

```
\begin{Scratch}
\scbox{if on edge, bounce}{mvt}
\scbox{hide}{app}
\scbox{stop all sounds}{son}
\scbox{pen down}{stylo}
\scbox{reset timer}{capt}
\end{Scratch}
```

you get:



3.2 Specials

3.2.1 Events

Obtained with `\beginbox{<text>}`.


`\beginbox{}` gives

`\beginbox{clone}` gives

`\beginbox{when this sprite clicked}` gives

3.2.2 Turn

Obtained with `\turnbox{<orientation>}{<angle>}`.

`\turnbox{}{-145}` gives 

`\turnbox{gauche}{30}` gives 

You can also write `\turnbox{g}{30}` or `\turnbox{G}{30}`.

3.2.3 Loops

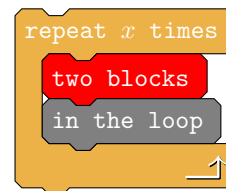
Obtained with

`\boucle{<text>}{<number of blocks inside>}{<type>}`,

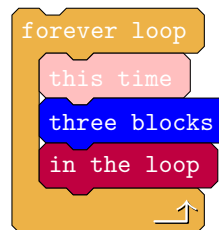
where `<type>` is an integer which can be equal to 1 (*repeat x times* or *repeat until*) or -1 (*forever loop*).

`\boucle{repeat x times}{2}{1}`
`\sbox{two blocks}{red}`
`\sbox{in the loop}{gray}`

makes



`\boucle{forever loop}{3}{-1}`
`\sbox{this time}{pink}`
`\sbox{three blocks}{blue}`
`\sbox{in the loop}{purple}`



Notice that if you want to put a loop into another loop, you have to count the correct number of blocks inside the main loop. Actually, a loop counts for two blocks (without the inside blocks).

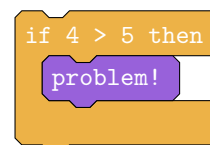
3.2.4 If ... Then

Notice that you get this command with the same syntax seen above. Actually:

`\boucle{<text>}{<number of blocks inside>}{2}`.

`\boucle{if 4 > 5 then}{1}{2}`
`\sbox{problem!}{app}`

produit



Remark: in order to properly type the *if ... then* test, see the *small boxes* in section 3.3.

3.2.5 If ... Then ... Else

Obtained with

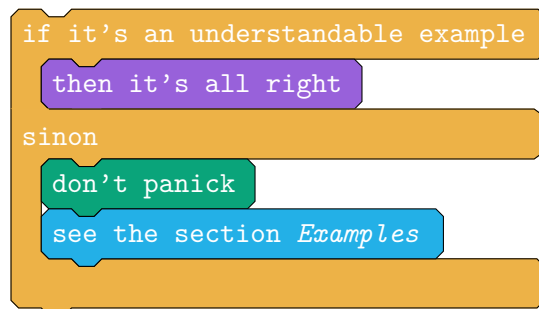
`\sailors{<text>}{<number of blocks inside>}`

et
`\simenon{<number of blocks inside>}`.

Typing:

```
\begin{Scratch}
\sailors{if it's an understandable example}{1}
\sbox{then it's all right}{app}
\simenon{2}
\sbox{don't panick}{stylo}
\sbox{see the section \emph{Examples}}{capt}
\end{Scratch}
```

you get:



3.2.6 More blocks

You get them with `\blocbox{<text>}`.

`\blocbox{triangle}` gives



3.2.7 Spécial control

It's for `stop` `all ▼` and `delete this clone`.

These commands are obtained with `\kbox{<text>}`.

3.3 Inside the Scratch commands: the little boxes

How to get some specific commands, like: `stop` `all ▼` ?

How to type `wait until` `color` `is touching` ?

or `point towards` `mouse-pointer ▼` ? or even


 A Scratch block with a purple header "change volume by" and a green body containing "pick random 1 to 10".

3.3.1 The little rectangular boxes

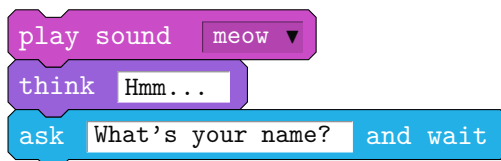
- In the `\scbox`:
obtained with
`\rb[<color>]{<text>}`
- dans les `\beginbox` :
you get them with
`\rbb[<color>]{<text>}`

In both cases, `<color>` has by default the color of the box it is inside. In order to get a white rectangular box, you just have to put `<color>` at white or w.

Typing:

```
\begin{Scratch}
\scbox{play sound \rb{meow}}{son}
\scbox{think \rb[white]{Hmm\dots}}{app}
\scbox{ask \rb[w]{What's your name?} and wait}{capt}
\end{Scratch}
```

you get:



3.3.2 The small round boxes

They are hollowed or embossed.

You get them with
`\cb[<color>]{<text>}`

By default, `<color>` has the same color of the circular box it is inside. If `<color>` is white or w, the circular box is hollowed.

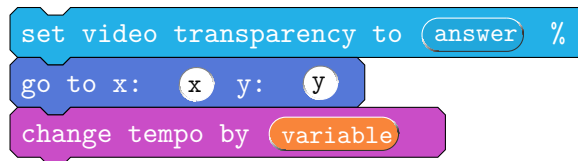
Typing:

```

\begin{Scratch}
\scbox{set video transparency to \cb{answer} \%}{capt}
\scbox{go to x:\cb[w]{x} y:\cb[white]{y}}{mvt}
\scbox{change tempo by \cb[data]{variable} }{son}
\end{Scratch}

```

you get:



3.3.3 The small hexagonal boxes

Only for *Sensing* et *Operators* commands.

You get them with

```
\hb[<color>]{<text>}
```

By default, <color> is ope.

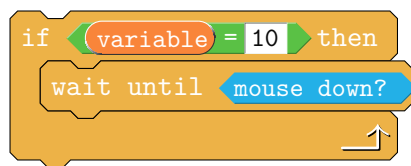
Typing:

```

\begin{Scratch}
\boucle{if \hb{\cb[data]{variable}=\rb[w]{10}} then}{1}{1}
\scbox{wait until \hb[capt]{mouse down?}}{ctrl}
\end{Scratch}

```

you get:



3.3.4 The small squared boxes

Only for the colored squares.

You get them with

```
\sqb{<color>}
```

Typing:

```

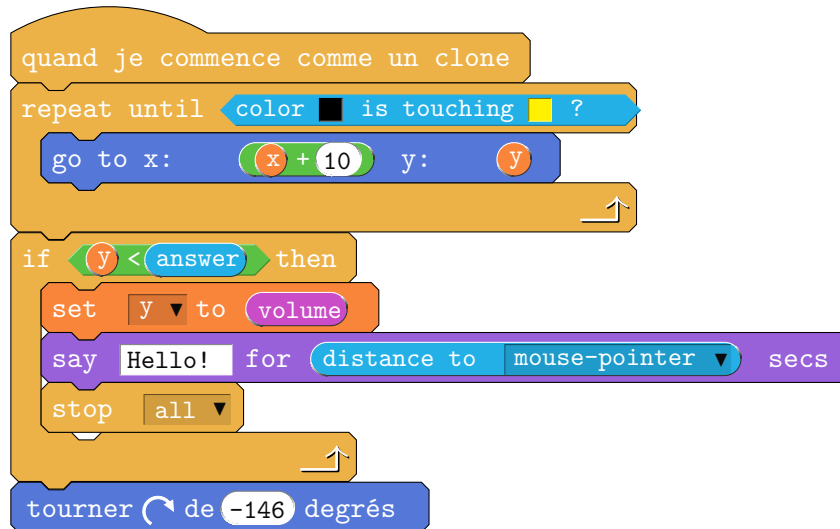
\begin{Scratch}
\scbox{set pen color to \sqb{brown}}{stylo}
\end{Scratch}

```


you get:



3.4 Intricate commands



Got with:

```
\begin{Scratch}
\beginbox{clone}
\boucle{repeat until \hb[capt]{color \sqb{black}
  is touching \sqb{yellow} ? }}{1}{1}
\scbox{go to x: \cb[oep]{\cb[data]{x}+\cb[w]{10}} y: \cb[data]{y}}{mvt}
\boucle{if \hb{\cb[data]{y}<\cb[capt]{answer}} then}{3}{1}
\scbox{set \rb{y} to \cb[son]{volume}}{data}
\scbox{say \rb[w]{Hello!} for \cb[capt]{distance to
  \rb{mouse-pointer} } secs}{app}
\kbox{stop \rb{all}}
\turnbox{2}{-146}
\end{Scratch}
```

3.5 Other sort of commands

3.5.1 In the loops

You need to use the command `\blank` when in the Scratch program, two loops ends at the same time.

Typing:

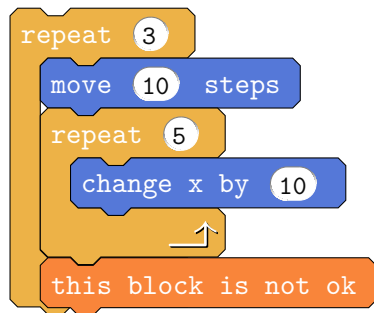
```
\begin{Scratch}
\boucle{repeat \cb[w]{3}}{4}{1}
```

```

\scbox{move \cb[w]{10} steps}{mvt}
\boucle{repeat \cb[w]{5}}{1}{1}
\scbox{change x by \cb[w]{10}}{mvt}
\scbox{this block is not ok}{data}
\end{Scratch}

```

you get:



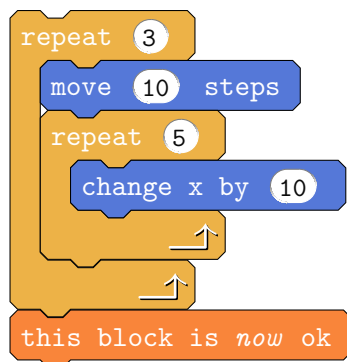
Whereas typing:

```

\begin{Scratch}
\boucle{repeat \cb[w]{3}}{4}{1}
\scbox{move \cb[w]{10} steps}{mvt}
\boucle{repeat \cb[w]{5}}{1}{1}
\scbox{change x by \cb[w]{10}}{mvt}
\blank
\scbox{this block is \emph{now} ok}{data}
\end{Scratch}

```

you get:



If you type:

```

\begin{Scratch}
\boucle{repeat \cb[w]{3}}{5}{1}

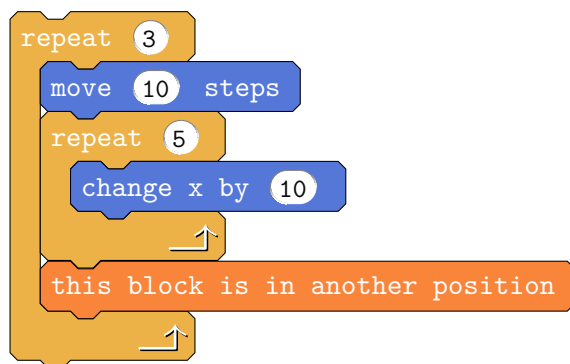
```

```

\scbox{move \cb[w]{10} steps}{mvt}
\boucle{repeat \cb[w]{5}}{1}{1}
\scbox{change x by \cb[w]{10}}{mvt}
\scbox{this block is in another position}{data}
\end{Scratch}

```

you get:



3.5.2 How to draw the cat

You can get the cover cat with:

```
\Scratchy[<scale>][<lines width>]
```

By default, the scale is 0.25 and the lines width is set at 0.25 pt.

Here is the code for the cover:

```
\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 Known problems and solutions

1. The black little triangle doesn't exist in the command *point in direction* (motion). You have to write it down.

```
\scbox{point in direction \cb[w]{90 \scriptsize$\blacktriangledown$}}{mvt}
```



2. The height of the boxes is set. Therefor, you cannot put a lot of under-commands in a Scratch command.
3. When you need to put only one Scratch command into some text, it is not vertically centered. You can use a: `\raisebox{-3mm}`.
4. There is also a tiny horizontal gap. When a Scratch environment is over, you often need to add a `\hspace{-1cm}`.
5. The compile time is sometimes long!

5 Commands summary

`\beginbox{}`

quand  est cliqué

`\beginbox{<text>}`
(when this sprite clicked)

quand ce lutin est cliqué

`\beginbox{clone}`

quand je commence comme un clone

`\blocbox{<text>}`

définir  fonction

`\turnbox{}{90}`

tourner  de 90 degrés

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

tourner  de -270 degrés

`\sbox{<text>}{<color>}`

any color, any text

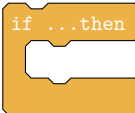
`\boucle{repeat}{2}{1}`
({<text>}{<nbr blocks>}{<type>})

repeat

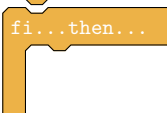

`\boucle{forever}{1}{-1}`
({<text>}{<nbr blocks>}{<type>})

forever

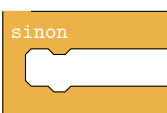

`\boucle{if ...then}{1}{2}`
({<text>}{<nbr blocks>}{<type>})

if ...then


`\sailors{if...then...}{2}`
({<text>}{<nbr blocks>})

fi...then...


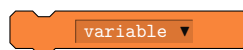
`\simenon{<nbre blocks>}`
(ici, `\simenon{1}`)

sinon


`\kbox{<text>}`

delete this clone

`\rb{<text>}`
(`\rb{variable}`)



`\rb[w]{<text>}`
(ou `\rb[white]{<text>}`)



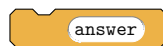
`\rbb{<text>}`
(only for `\beginbox`)
`\beginbox{when I receive \rbb{message1}}`



`\cb{<text>}`
(transparent)



`\cb[w]{<text>}`
(ou `\cb[white]{<text>}`)



`\cb[<color>]{<text>}`



`\cb[o]pe{answer}`

`\hb{<text>}`



`\hb[cap]{<text>}`



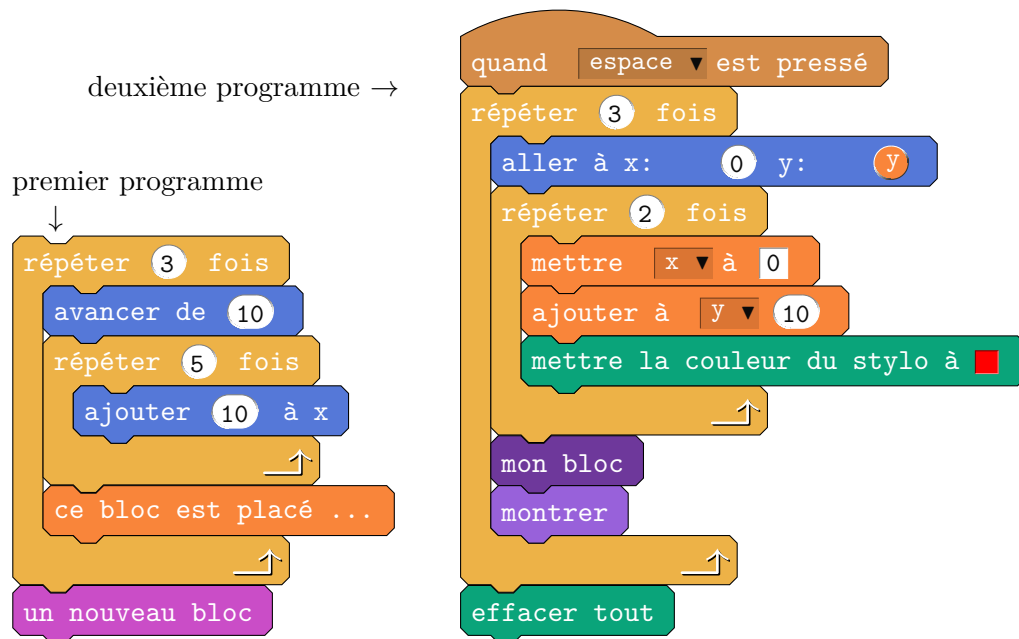
`\sqb{<color>}`



6 Examples of programs (in French):

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

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



6.1 Loops of loops

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

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

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

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

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

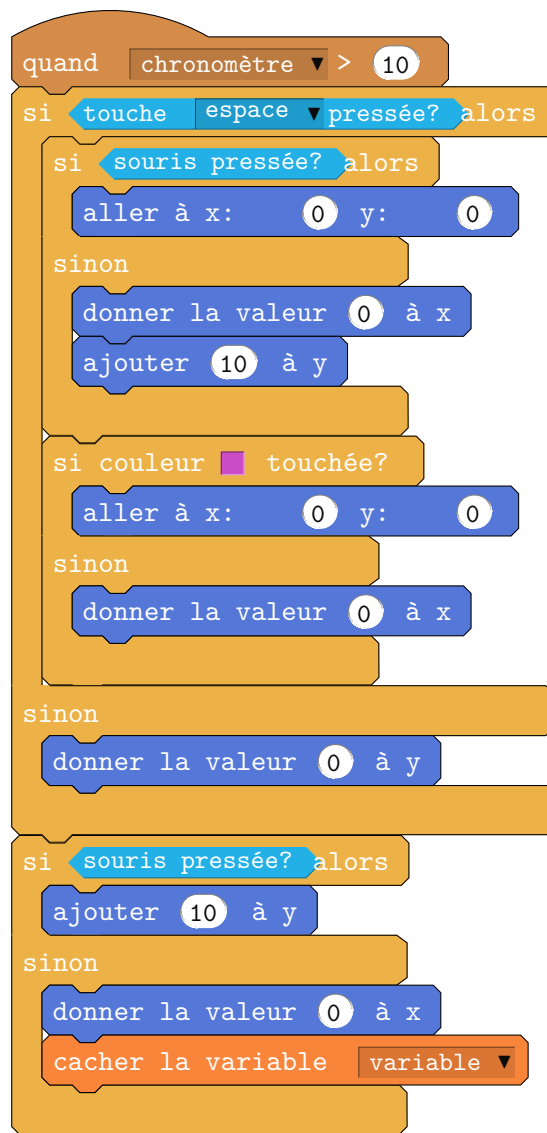
\simenon{1}
\scbox{donner la valeur \cb[w]{0} à x}{mvt}

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

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

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

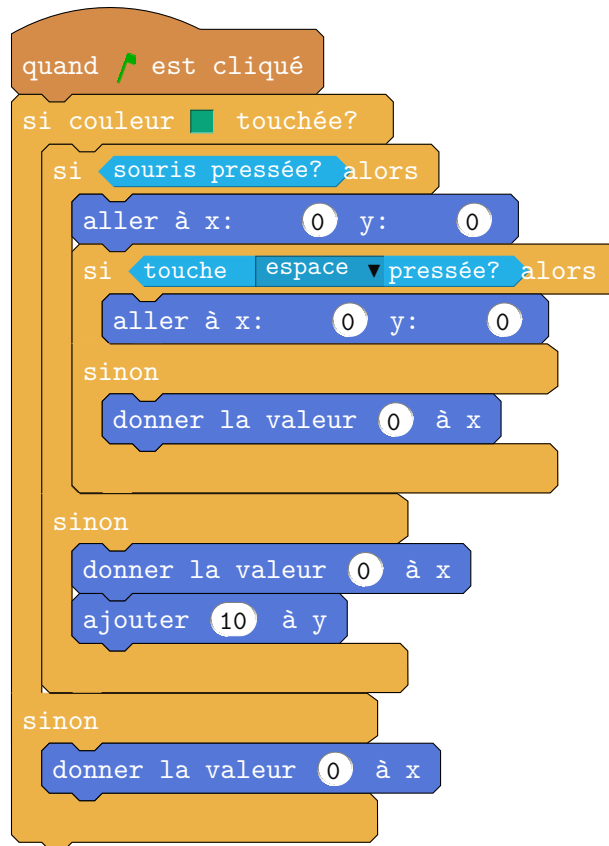
\sailors{si \hb[capt]{touche \rb{espace} pressée?} alors}{1}
\scbox{aller à x: \cb[w]{0} y: \cb[w]{0}}{mvt}

\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}

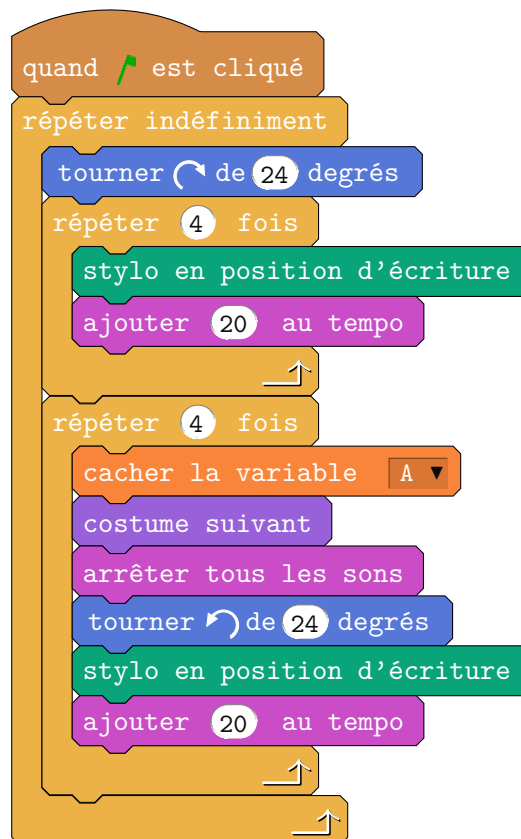
\end{Scratch}
```



```

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

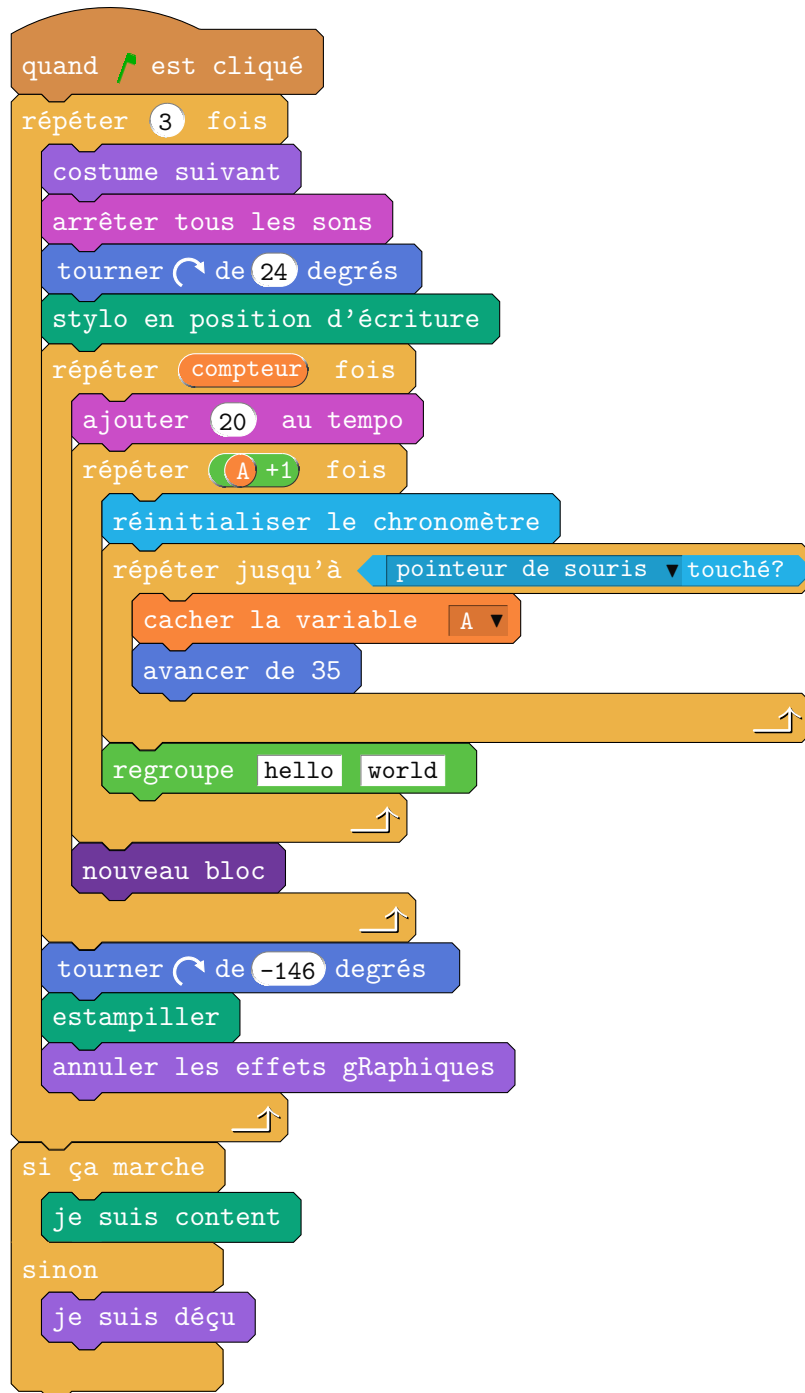
```



```
\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[oep]{\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

Please feel free to leave a comment:

Thibault.Ralet@ac-clermont.fr.

Thank you!