

Algoid
www.algoid.net



Présenté par Yann Caron
aka CyaNn

Un langage de programmation, kezako ?

C'est comme une langue (le français), mais
pour les ordinateurs !

Comme en français, c'est impératif :

Fait ceci !

Fait cela !

Fait pas ci, fait pas ça ! la la la !

Une suite d'instructions (des ordres)

```
print ("Je calcule 4 x 3")  
set resultat = 4 * 3  
print ("le résultat de 4 x 3 est " .. resultat)
```

- La machine est obéissante, elle s'exécute :
- elle exécute le programme !

Qui est Algo ?

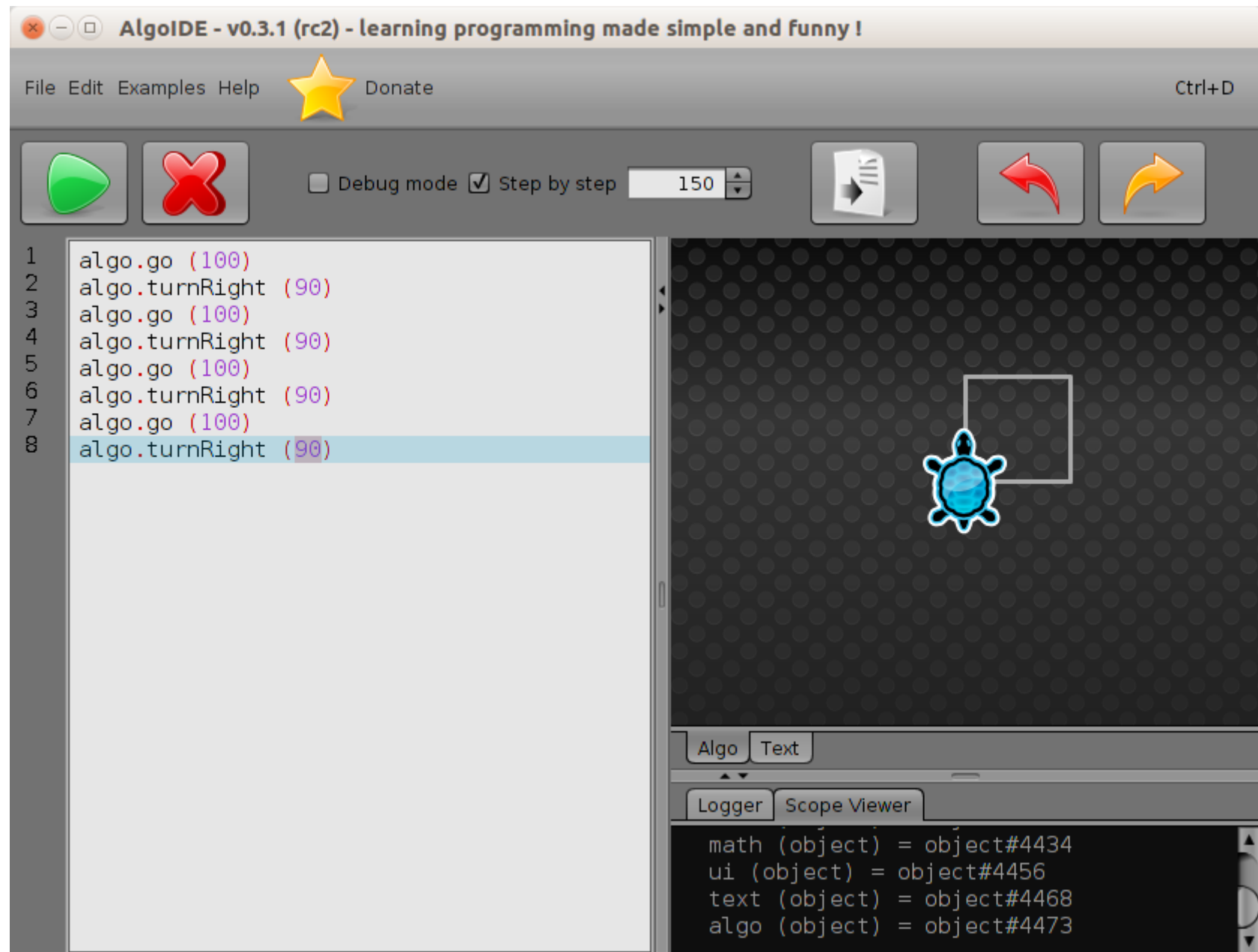
- Bonjour Algo la petite tortue !
- Algo est un objet
- Et on dit à l'objet ce qu'il doit faire

```
algo.go (100)  
algo.turnRight (90)
```

- A nous de jouer !
- Un carré ?

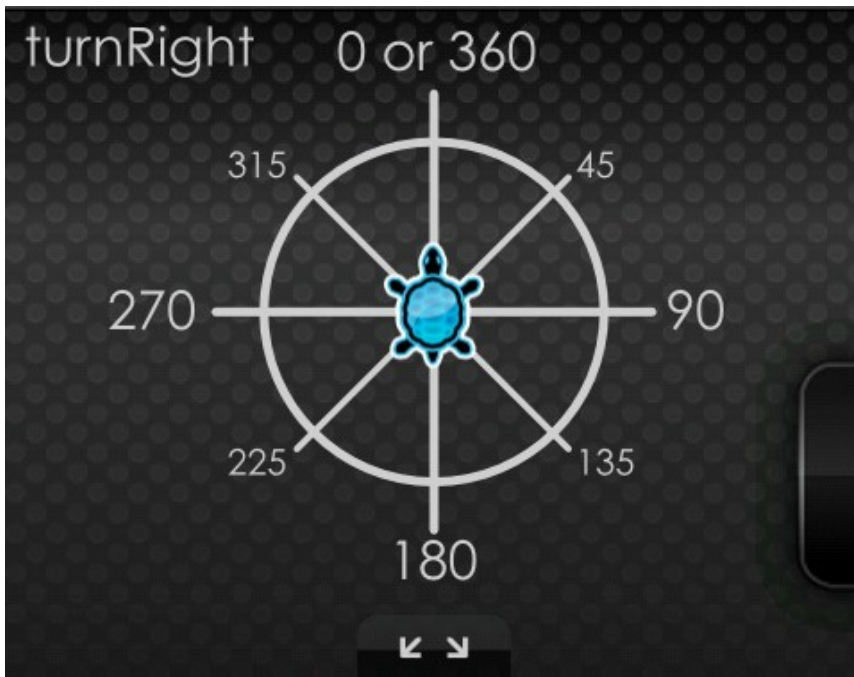


Demo 1



Eurêka ! Tout est relatif !

```
algo.turnRight (90)
```



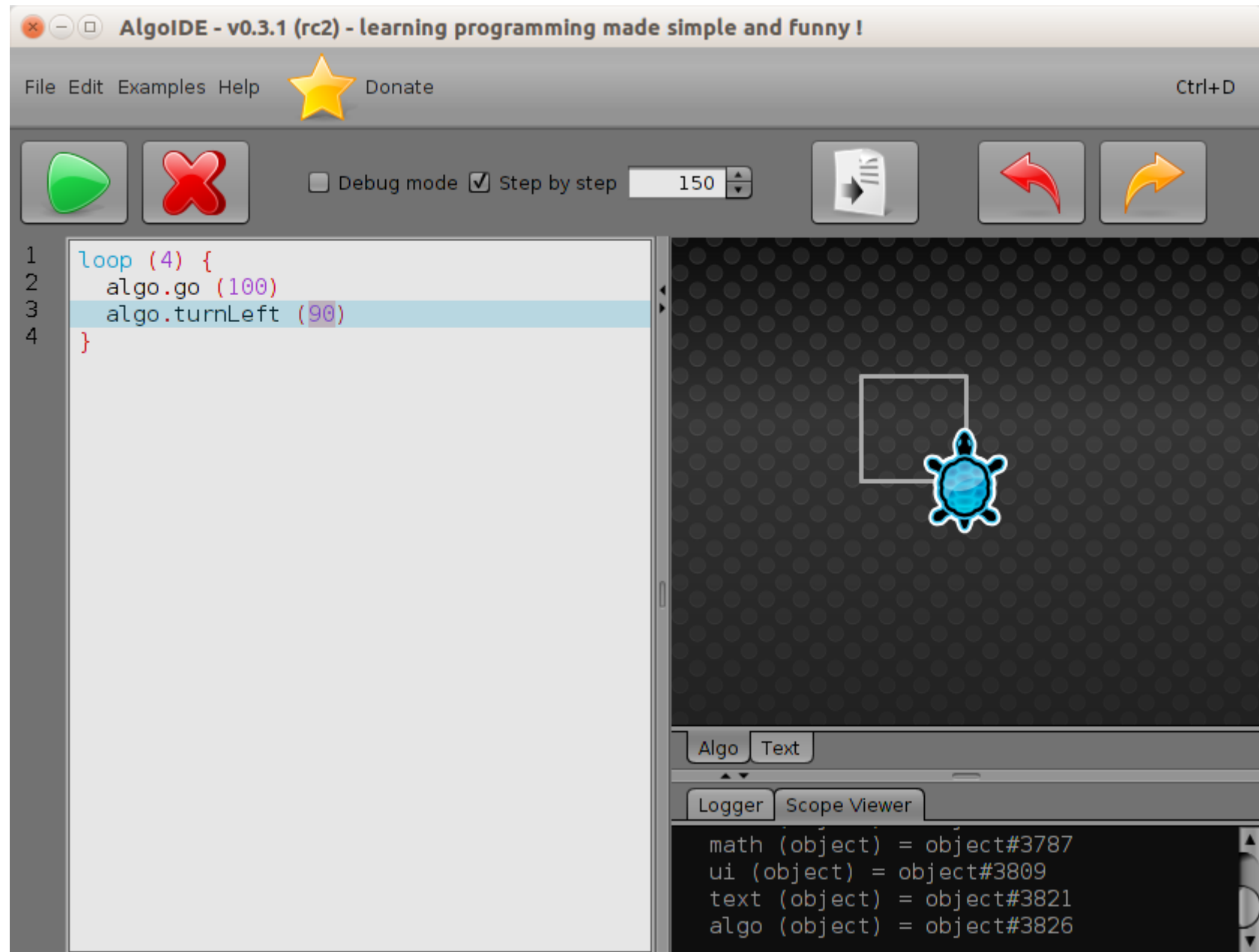
Et la boucle est bouclée

```
algo.go (100)  
algo.turnRight (90)  
algo.go (100)  
algo.turnRight (90)  
algo.go (100)  
algo.turnRight (90)  
algo.go (100)  
algo.turnRight (90)
```

```
loop (4) {  
    algo.go (100)  
    algo.turnLeft (90)  
}
```



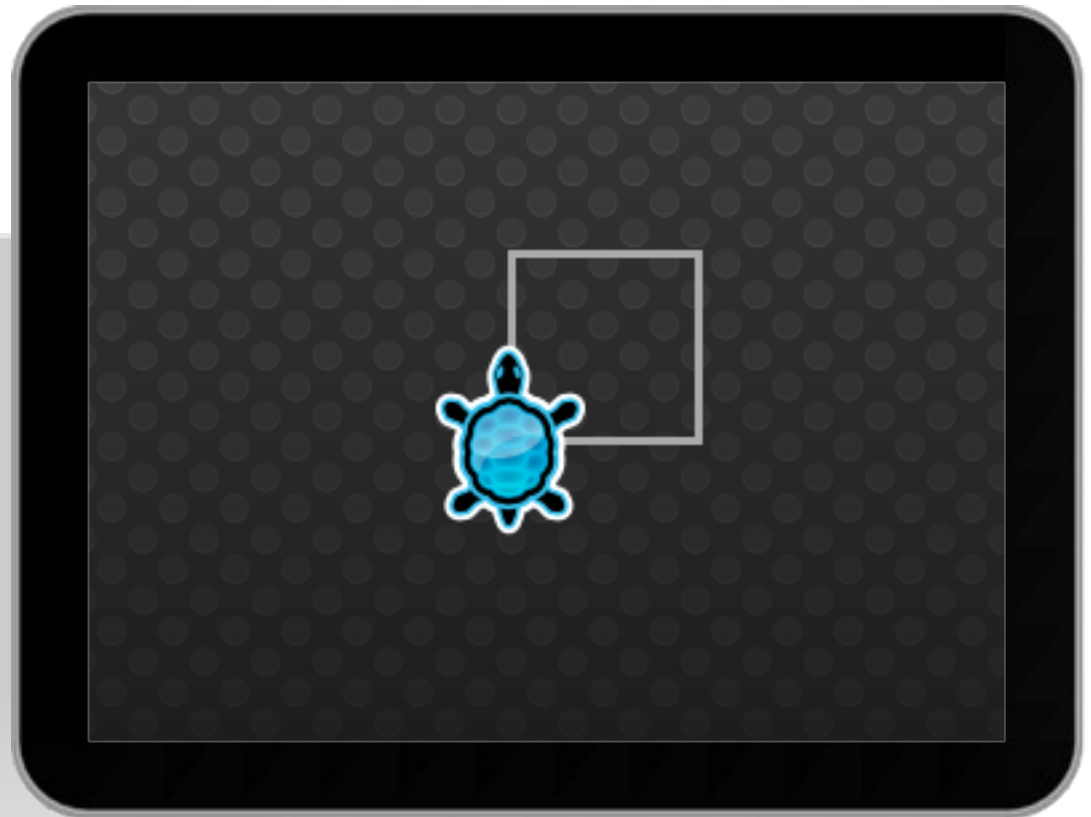
Demo 2



Les variables


```
set nbCote = 4
set angle =
360 / nbCote // 90°



loop (nbCote) {
  algo.go (100)
  algo.turnRight (angle)
}
```




Demo 3

AlgoIDE - v0.3.1 (rc2) - learning programming made simple and funny !

File Edit Examples Help  Donate Ctrl+D

  ☒ Debug mode ☒ Step by step 150

```
1
2  set nbCote = 4
3
4  set angle = 360 / nbCote // 90°
5
6  loop (nbCote) {
7
8      algo.go (100)
9
10     algo.turnRight (angle)
11
12 }
```



Algo Text

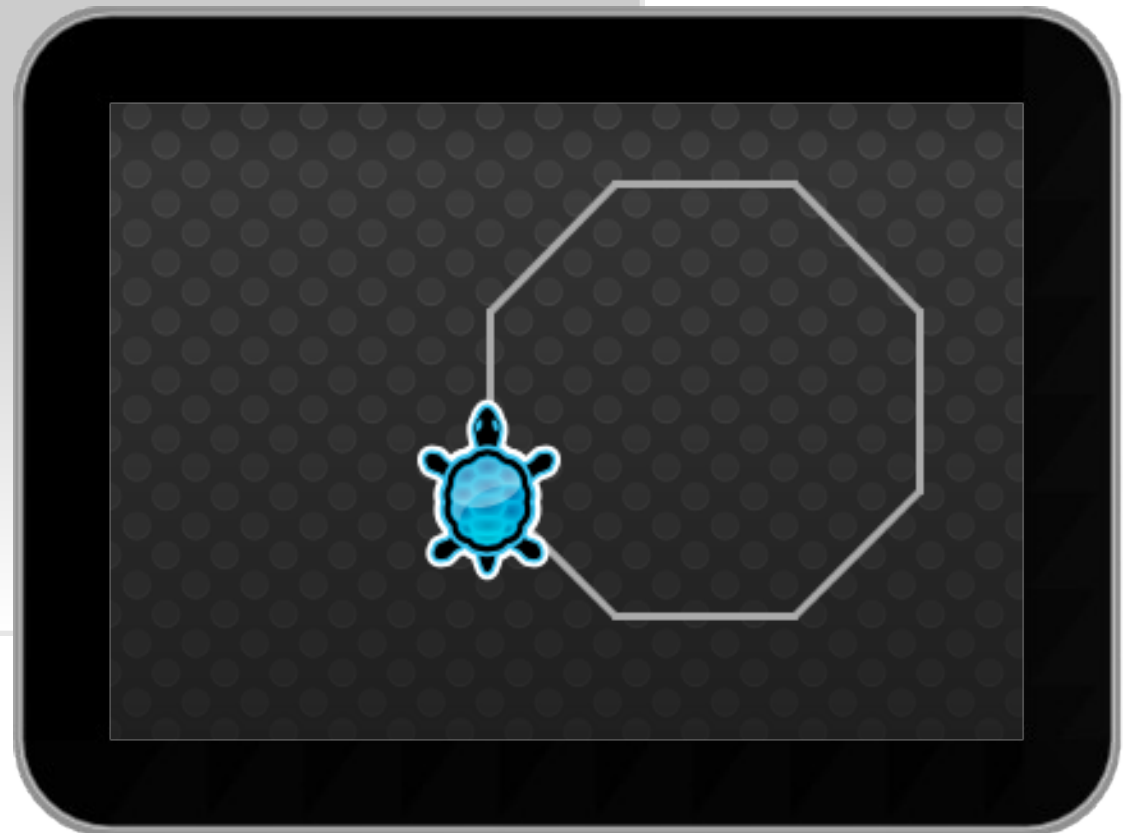
Logger Scope Viewer

```
algo (object) = object#8355
nbCote (number) = 4
angle (number) = 90

loop
```

On change la donne

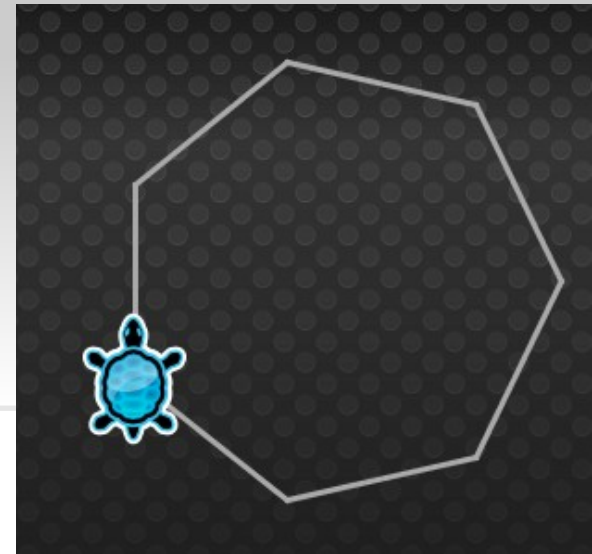
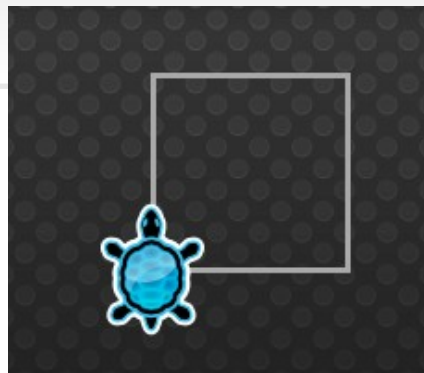
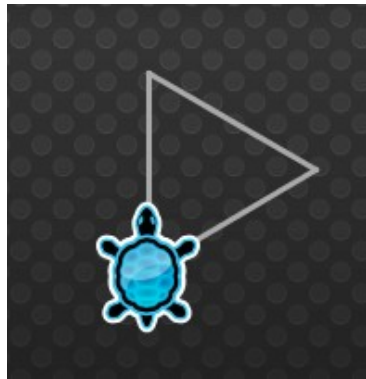
```
set nbCote = 8  
set angle = 360 / nbCote // 45°  
  
loop (nbCote) {  
  algo.go (100)  
  algo.turnRight  
    (angle)  
}
```



Jeu de hasard

```
set nbCote = 3 + math.random(14)
set angle = 360 / nbCote // ??°

loop (nbCote) {
  algo.go (150)
  algo.turnRight (angle)
}
```



Créer / sauver

- Créons maintenant un nouveau programme
- Appelons le Kids2015
- N'oublions pas de le sauver régulièrement



exercice 1: Le carré

- Difficulté : très facile
- Dessinons un carré à l'aide de `algo.go` et `algo.turnLeft` (ou `turnRight`)
- Puis avec seulement deux lignes plus l'instruction `loop`
- Comment faire un rectangle ?

exercice 2 : le triangle

- Difficulté : moyen
- Dessinons un triangle
- Puis à l'aide de l'instruction `algo.circle`, dessinons les cercles autour des points.
- Faisons disparaître les traits du triangles avec l'instruction `algo.jump`

exercices 3 : l'escalier

- Difficulté : moyen
- Dessinons un escalier
- Demandons nous comment réduire le nombre de ligne de programme ?
- A l'aide de ce que nous avons fait, dessinons une pyramide

exercice 4 : la cible

- Difficulté : difficile
- A l'aide de l'instruction disc, dessinons une cible
- Demandons nous comment réduire le nombre de ligne de programme ? (instruction for ou utilisation d'une variable)
- Ecrivons un programme qui permet de définir toutes les cibles que l'on veut à l'aide des variables

exercice 5 : la spirale

- Difficulté : moyen +
- Dessinons une spirale à l'aide d'une instruction loop
- A l'aide des variables, donnons des paramètres à notre spirale

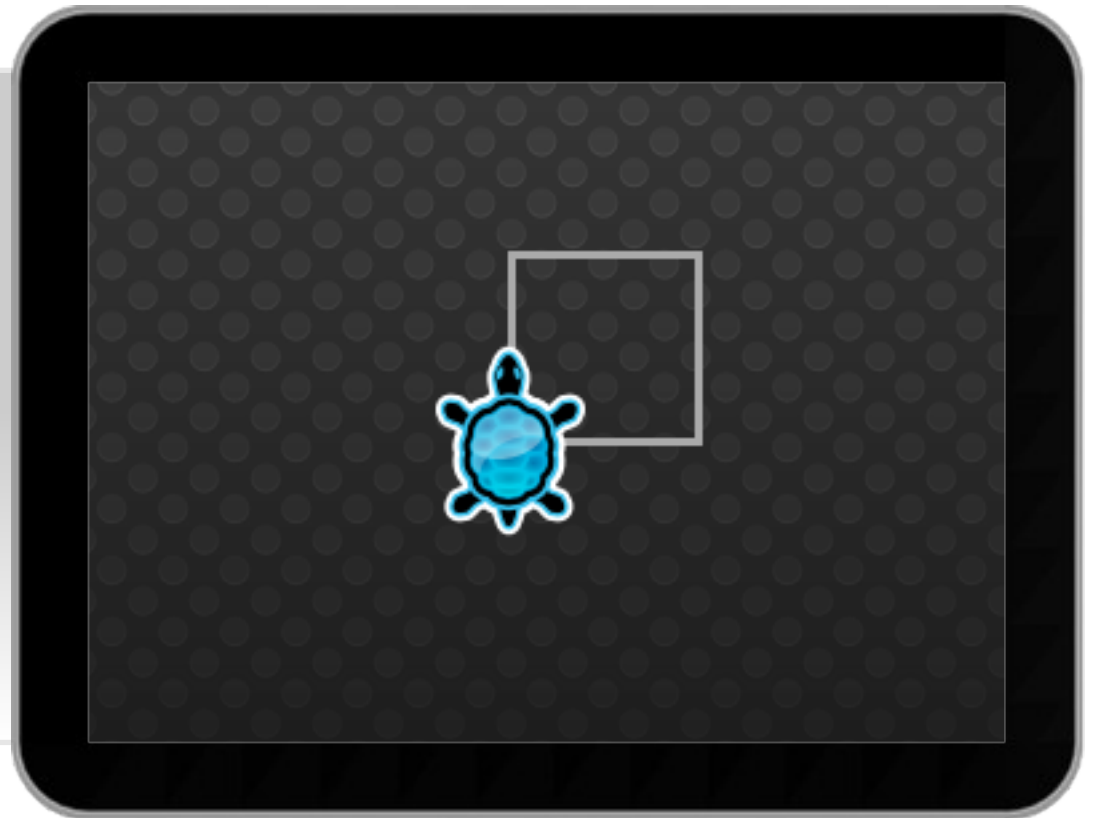
Un pause ?



Les fonctions

- Kezako ?

```
set carre = function () {  
  loop (4) {  
    algo.go (100)  
    algo.turnRight (90)  
  }  
}  
  
carre ()
```

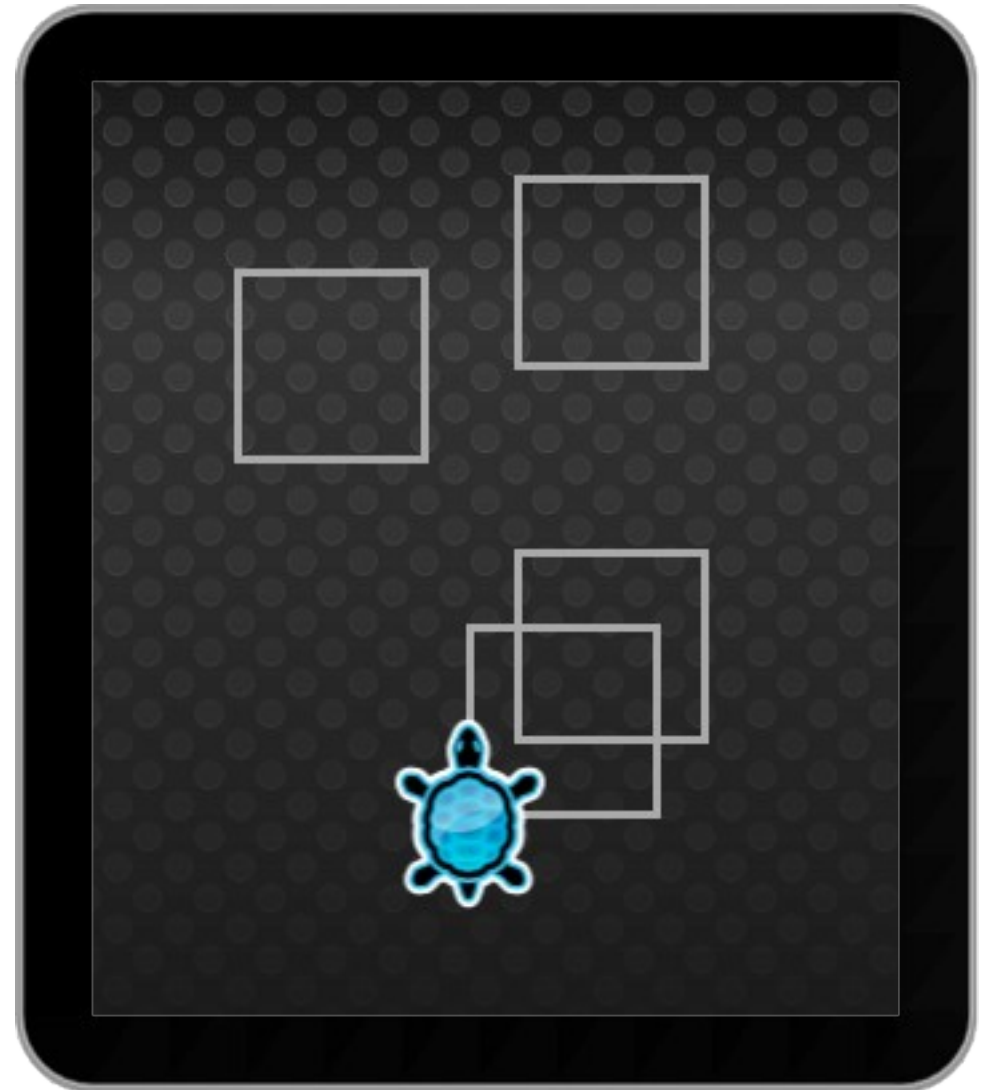


- Encore un carré ? Oui mais !

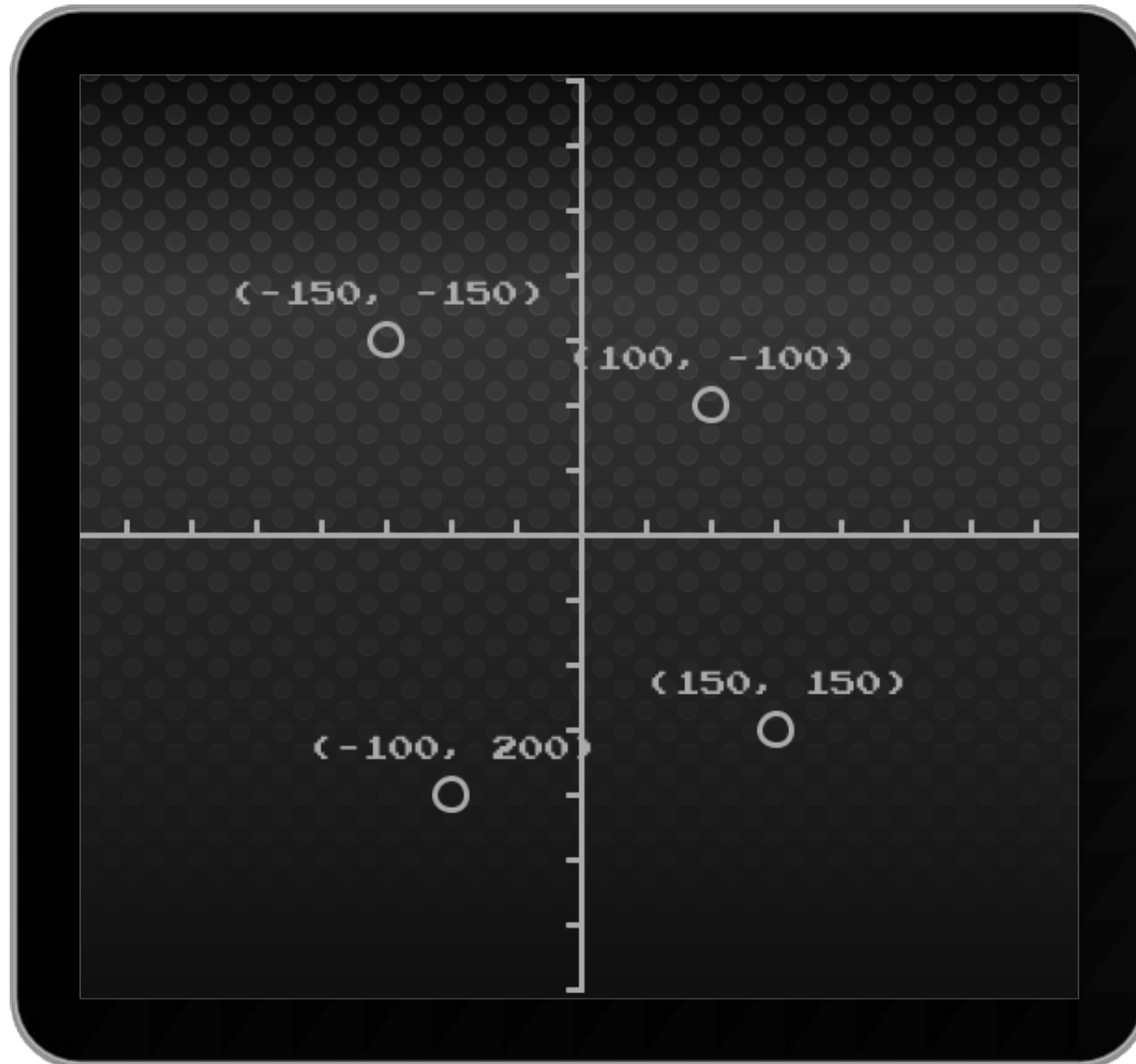
Utilisons la fonction

- On ré-utilise

```
algo.goTo(50, 50)  
carre ()  
algo.goTo(-100, -100)  
carre ()  
algo.goTo(50, -150)  
carre ()  
algo.goTo(25, 90)  
carre ()
```

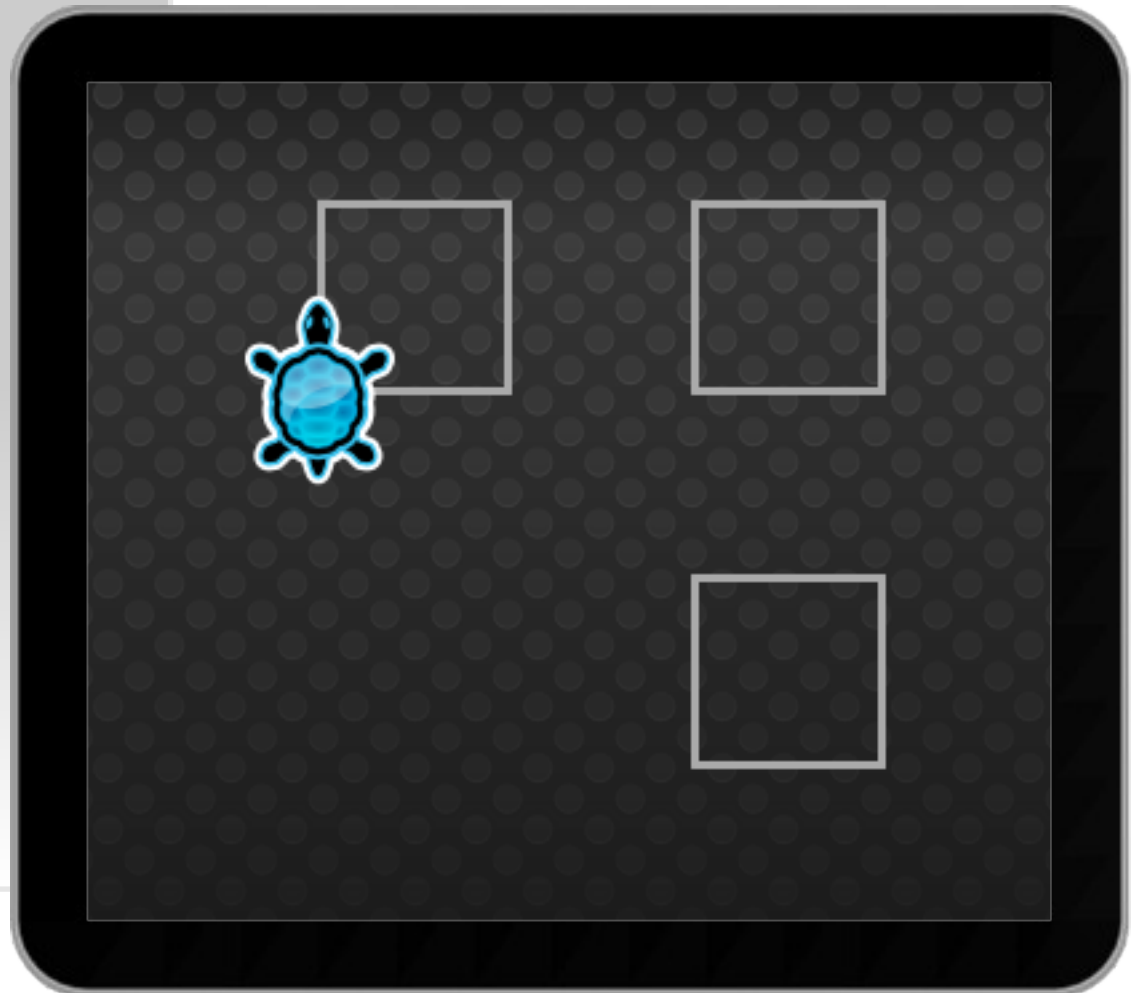


Absolument !



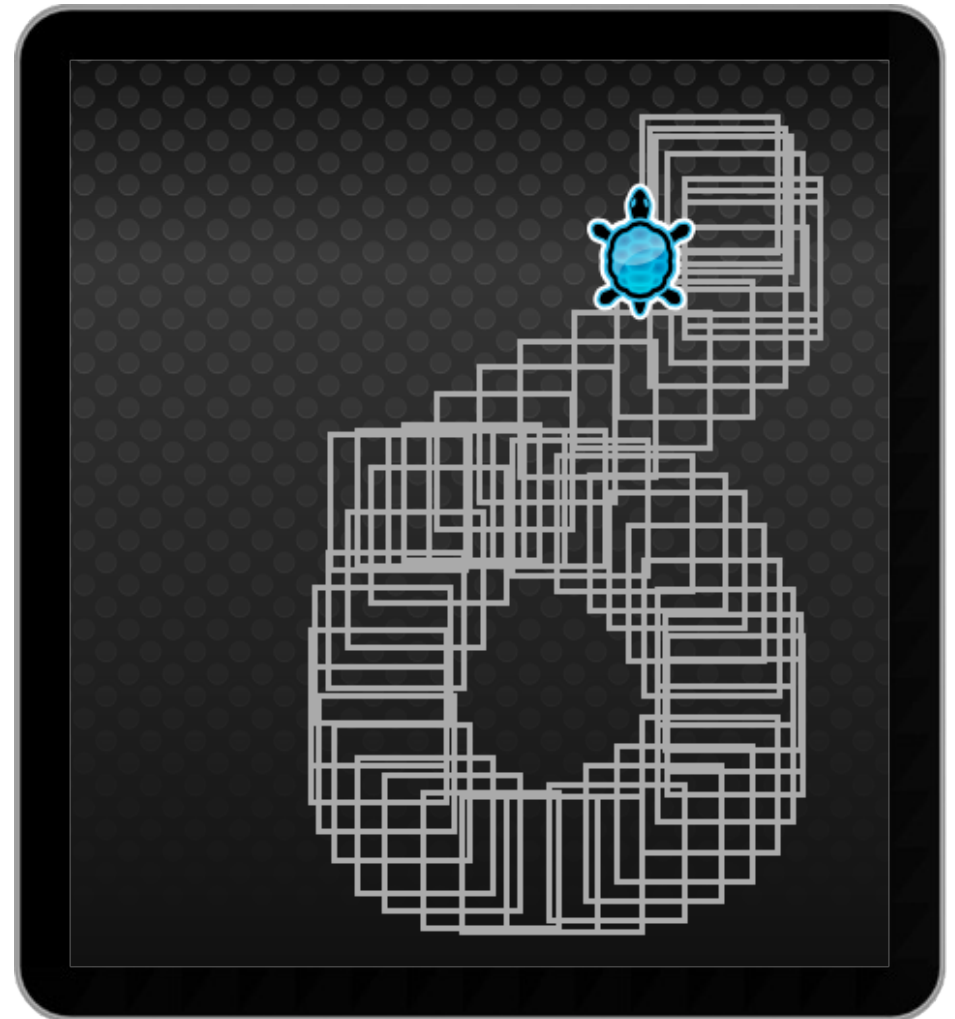
Les paramètres

```
set carre = function (x, y) {  
  algo.goTo (x, y);  
  loop (4) {  
    algo.go (100);  
    algo.turnRight (90);  
  }  
};  
  
carre (100, 100);  
carre (100, -100);  
carre (-100, -100);
```



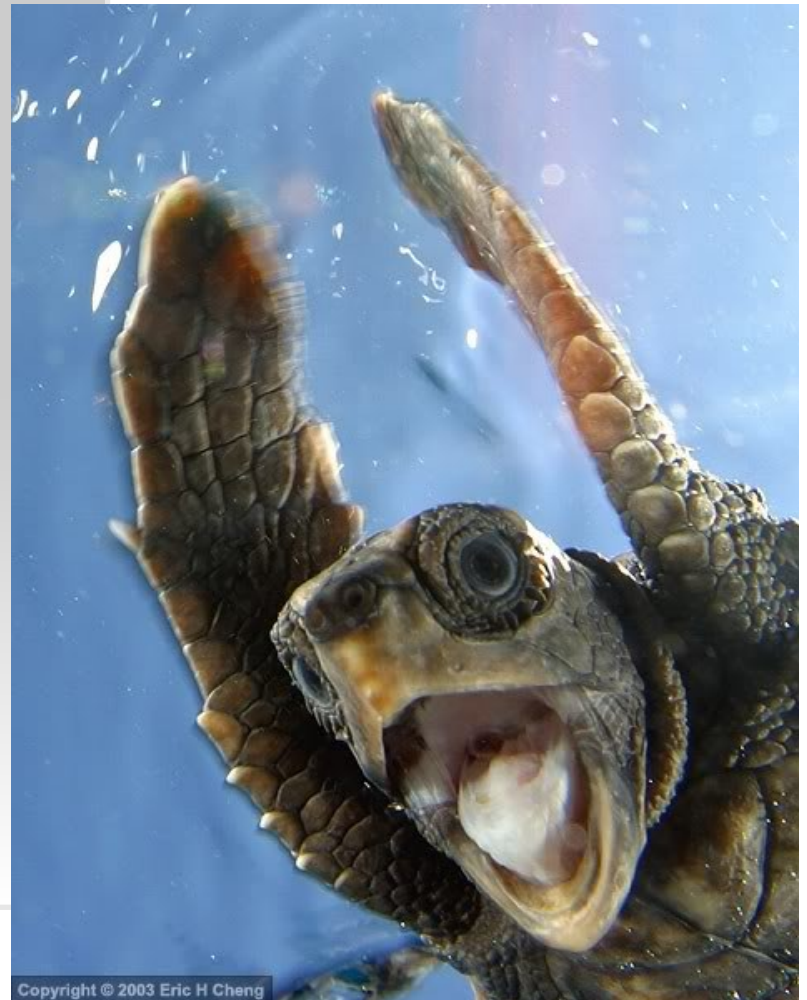
Les événements

```
set carre = function (x, y) {  
  algo.goTo (x, y);  
  loop (4) {  
    algo.go (100);  
    algo.turnRight (90);  
  }  
};  
  
algo.onTap (carre);  
  
// ou  
  
algo.onTouch (carre);
```



Il est vivant !

```
set carre = function (x, y) {  
  algo.clear ();  
  algo.goTo (x, y);  
  loop (4) {  
    algo.go (100);  
    algo.turnRight (90);  
  }  
};  
  
algo.onTouch (carre);
```



exercice commun

- Un logiciel de dessin !
- Créons une fonction qui dessine un disque `(algo.disc(10))`
- A la position donnée par `x` et `y` `(algo.goTo(x, y))`
- Branchons cette fonction à l'évènement `algo.onTouch()`
- Et la couleur par hasard ?
`(algo.setColor() / math.random() / 16 couleurs)`