La boucle if

Je suis parti de l'observation de code, distribué sur feuille de papier et j'ai demandé aux élèves d'essayer de comprendre ce qu'il produisait comme effet.

Inducteur : analyse de code

```
$a = 5
$b = ask "Saisir un chiffre (b) : "

if ($b < $a) {
$c = $a - $b
} else {
$c = $a + $b
}

message $c
```

```
# ======================
$genre = ask "Saisir f pour fille ; g pour garçon"

if $genre == "f" {
message "Bonjour Madame"
} else {
message "Nonjour Monsieur"
}
```

Premier temps

Après 5 à 6 minutes de réflexion, phase de bilan, de mise au point. Globalement le second code est compris. Et les retours sont du style “On demande à l'ordinateur si l'on est une fille ou un garçon…”

Le mot SI est donc mis en évidence, on le note au tableau. Et on écris l'algorithme correspondant.

Par contre, le premier code, on tâtonne mais le résultat n'est pas réellement compris.

Second temps

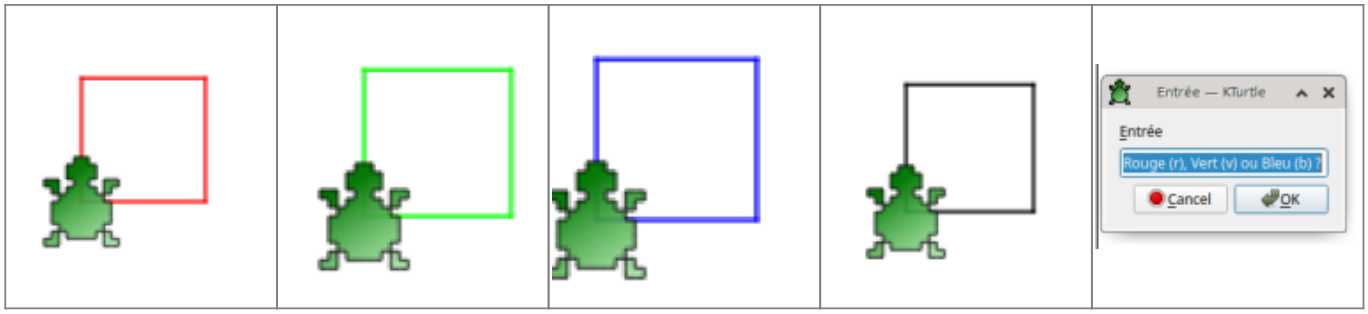
Donc pour le premier code, direction PC, les élèves le tapent et le testent plusieurs avec des valeurs différentes.

C'est plus simple pour eux pour déterminer la finalité de ce code.

On retourne à une phase collectif qui reprend le code ligne par ligne et on réalise ensemble un algorithme de fonctionnement.

Jouons avec les couleurs

** *Inducteur : Demander de poser la question “Quelle couleur pour le carré. Laisser un choix entre les 3 couleurs et en fonction de la réponse, appliquer la couleur choisie.***



L'élève va devoir réinvestir la condition **if** mais non plus une fois mais sur plusieurs tests :

- Tester si la valeur vaut “r” –> on applique le rouge
- Tester si la valeur vaut “b” –> on applique le bleu
- Tester si la valeur vaut “v” –> on applique le vert
- Tester si la valeur ne correspond ni à “r”, “v” ou “b” –> on laisse la couleur par défaut (noir)

Proposition de code

```
reset

$couleur = ask "Rouge (r), Vert (v) ou Bleu (b) ?"

# Test du Rouge
if ($couleur == "r")  {
    pencolor 255,0,0
}

# Test du Vert
if ($couleur == "v"  {
    pencolor 0,255,0
}

# Test du Bleu
if ($couleur == "b"  {
    pencolor 0,0,255
}

repeat 4 {
    forward 40
    turnright 90
}
```



Attention le contenu des variables est sensible à la casse.
C'est à dire que ”**r**“ est différent de ”**R**“. Il faudra donc tester



les deux si l'on veut que le programme est une batterie de tests exhaustive.

L'opérateur OR (OU)

Rebondissons sur ce qui a été dit en dernier : pour le rouge, on va tester non seulement le "**r**" mais aussi le "**R**"

L'élève proposera de fait le code suivant composé de 2 séquences de test différentes

```
# Test du Rouge minuscule
if ($couleur == "r") {
    pencolor 255,0,0
}
# Test du Rouge majuscule
if ($couleur == "R") {
    pencolor 255,0,0
}
```

Faire réfléchir les élèves à un moyen de proposer un code plus simple, moins chargé, optimisé. Style "***Et si on disait au PC de tester les 2 conditions d'emblée ? Comme avec un OU***".

On présente alors l'***opérateur* or**

Ce qui donne

```
# Test du Rouge
if ($couleur == "r") or ($couleur == "R" {
    pencolor 255,0,0
}
```

Et les deux séquences de test se retrouve inscrites en une seule séquence.

Donc le programme final

```
reset

$couleur = ask "Rouge, Vert ou Bleu ?"

# Test du Rouge
if ($couleur == "r") or ($couleur == "R")  {
    pencolor 255,0,0
}

# Test du Vert
if ($couleur == "v") or ($couleur == "V") {
    pencolor 0,255,0
}
```

```
# Test du Bleu
if ($couleur == "b") or ($couleur == "B") {
    pencolor 0,0,255
}

repeat 4 {
    forward 40
    turnright 90
}
```

Allez plus loin

- Allez plus loin, en demandant de tester aussi avec les mots “rouge” et “ROUGE”. De même pour les autres couleurs.
- Varier le programme avec une gamme de couleur plus importante

Et le else ???

Inducteur : Demander de choisir la figure géométrique à tracer

- Si figure est un carré → on trace un carré
- Si figure est un triangle → on trace un carré
- Si figure est un rectangle → on trace un carré
- SI figure est un losange → on trace un losange



Les programmes s'étoffent, pensez à l’intérieur des différents blocs à **indenter** le code à l'aide de **tabulations** pour faciliter sa lecture !

Vu le bagage qu'il possède, cela ne devrait pas représenter de réels soucis de codage. Sans doute quelques erreurs de syntaxe.



L'erreur la plus courante est de ne pas fermer systématiquement une accolade ouverte

```
{ ...
  bloc d'instruction 1
  # TEST
  {
    ....
```



```
        bloc d'instruction
        ....
    }
}
```

Proposition de code

```
reset

$figure = ask "Carré ? Rectangle ? Losange ? Triangle"

# Test du carré
if ($figure == "carré") {
    repeat 4 {
        forward 40
        turnright 90
    }
}

# Test du triangle
if ($figure == "triangle") {
    repeat 3 {
        forward 40
        turnright 120
    }
}

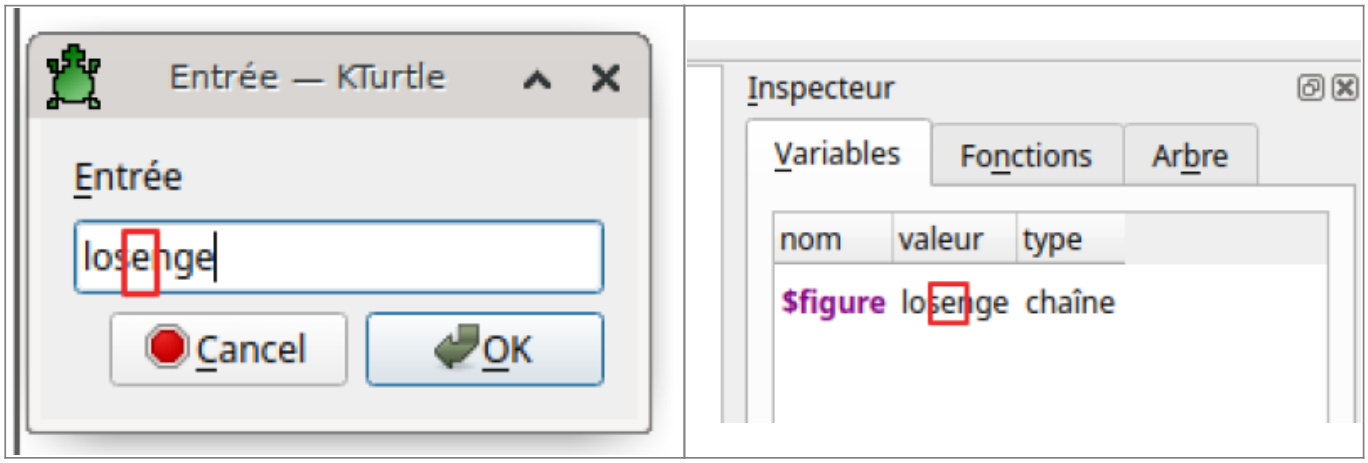
# Test du losange
if ($figure == "losange") {
    repeat 6 {
        forward 40
        turnright 60
    }
}

# Test du rectangle
if ($figure == "rectangle") {
    repeat 4 {
        forward 40
        turnright 90
        forward 80
        turnright 90
    }
}
```

Bien sûr lors de l'invitation de saisie de la pop-up, il arrivera que l'élève ait fait une faute de saisie (orthographe ou frappe).

Dans ce cas, la variante ne passera pas les tests et le programme ne fera rien. Sous doute l'élève répondre : “Non, j'ai bien tapé le mot”, dans ce cas, utilisez l'inspecteur de code afin de bien faire

l'erreur de saisie.



Le programme va donc devoir être amélioré avec quelques lignes basiques de **gestion de l'erreur**.

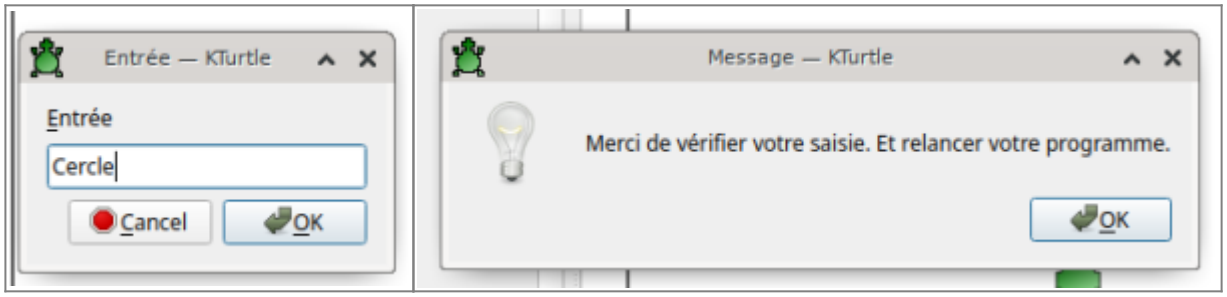
Donc, on va réaliser les tests nécessaires et gérer tout ce qui ne correspond pas à ces tests.

Si (test) **ALORS** (instruction) **Si** (autre test) **ALORS** (autre instruction) ... **Si** (autre test) **ALORS** (autre instruction)

Et enfin

SINON ALORS (instruction)

On introduit alors le mot clef, l'opérateur **ELSE**



Navigation

page précédente	Sommaire	Page suivante
Et le hasard dans tout cela ?	sommaire	

From:
https://cbiot.fr/dokuwiki/ - **Cyrille BIOT**

Permanent link:
https://cbiot.fr/dokuwiki/kturtle:kturtle-activites-11?rev=1578467905

Last update: **2020/01/08 07:18**