OBJECTIFS : L'objectif de ce TP est de réutiliser ce qui a été vu jusqu'à présent pour concevoir un premier jeu. On utilisera quelque chose de nouveau : les **évènements**.

DOCUMENT A RENDRE : Ce travail est évalué. Vous en rédigerez un compte-rendu numérique en utilisant un logiciel de traitement de texte (*Word ou Libre Office*). Le fichier constitué sera appelé *tp6.doc* ou *tp6.odt* et devra être transféré en fin d'activité **par l'intermédiaire** du site *nsibranly.fr* : se loguer et transférer en utilisant le code **tp6** . Ce compte-rendu contiendra :

1. les réponses aux différentes questions posées,
2. les captures d'écran **des morceaux de codes** écrits **et** celles **des résultats des exécutions** données dans le shell. Pour faire ces captures, utiliser *l'Outil Capture d'écran* de Windows.

1. Code qui calcule un écart positif entre 2 nombres :

Le code ci-dessous est incomplet :

exercice_1.py

```
# *****  FONCTIONS **************************
def ecart(

# ****** PROGRAMME PRINCPAL ****************
assert ecart(5,1) == 4
assert ecart(1.5) == 4
```

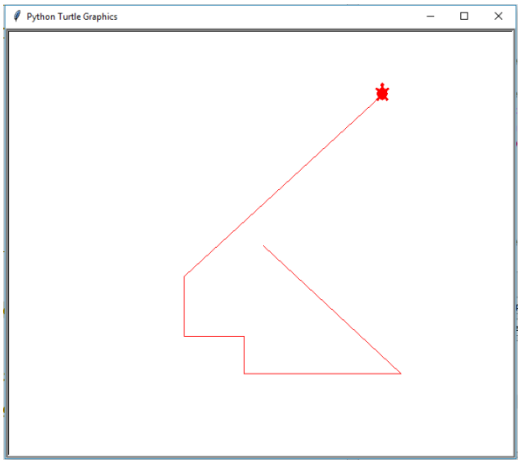
En l'exécutant, cela donne :

```
>>> (executing file "exercice_1.py")
```

⇨ Recopier et compléter ce code pour obtenir le même résultat à l'exécution.

2. Utilisation d'évènements avec Turtle

⇨ Recopier le code ci-contre dans un fichier à enregistrer sous le nom exercice_2. L'exécuter :



exercice_2.py

```
##----- Importation des Modules -----##
from turtle import *
##----- Définition des Fonctions -----##
def deplacement():
    forward(10)

def tourne_gauche():
    left(90)

def tourne_droite():
    left(-90)

def direct(x, y):
    goto(x,y)

##----- Main -----##
# Apparence de la tortue :
shape('turtle')
color('red')
# Evènements liés aux touches du clavier
onkey(tourne_gauche, 'Left')
onkey(tourne_droite, 'Right')
onkey(deplacement, 'space')
# Evènements liés à la souris
onscreenclick(direct)
##--Permet aux évènements de fonctionner
listen()
mainloop()
```

Ce code est composé des parties suivantes :

- Une gestion de l'apparence de la tortue :

```
17 # Apparence de la tortue :
18 shape('turtle')
19 color('red')
```

- Une gestion de 3 évènements liés à 3 touches du clavier et qui utilisent la fonction *onkey()* de *Turtle* :

```
20 # Evènements liés aux touches du clavier
21 onkey(tourne_gauche, 'Left')
22 onkey(tourne_droite, 'Right')
23 onkey(deplacement, 'space')
```

  - L'appui sur la touche *espace* provoque l'exécution de la fonction *deplacement()*
  - L'appui sur la touche ⇦ provoque l'exécution de la fonction *tourne_gauche()*
  - L'appui sur la touche ⇨ provoque l'exécution de la fonction *tourne_droite()*

- Une gestion d'un évènement lié au click sur la souris et qui utilise la fonction *onscreenclick()* de *Turtle*. L'appui sur le bouton gauche de la souris provoque l'exécution de la fonction *direct(x,y).*

```
24 # Evènements liés à la souris
25 onscreenclick(direct)
```

⇨ Insérer une copie d'écran liée à l'exécution de votre code. Ecrire avec la tortue la lettre :

3. Comment créer plusieurs tortues ?

On a vu précédemment que pour créer une tortue, on écrivait :

```
1 from turtle import *
2
3 shape("turtle")
4 color("red")
5
6 exitonclick()
```

Si on veut créer 2 tortues, on utilise une syntaxe légèrement différente :

```
1  from turtle import *
2
3  t1 = Turtle()
4  t1.shape("turtle")
5  t1.color("red")
6
7  t2 = Turtle()
8  t2.shape("turtle")
9  t2.color("blue")
10
11
12
13 # -- A LAISSER --
14 mainloop()
15 exitonclick()
```

Les tortues créées deviennent ici ce que l'on appelle des **objets** qui sont notés ici t1 et t2.

Si on veut déplacer ces objets séparément, on utilise les commandes habituelles de *Turtle*, mais précédées du nom de l'objet tortue avec un point. Exemple :

```
from turtle import *

t1 = Turtle()
t1.shape("circle")
t1.color("red")

t2 = Turtle()
t2.shape("turtle")
t2.color("blue")

t1.goto(100,100)
t2.up()
t2.forward(300)
t2.right(90)
t2.forward(200)

# -- A LAISSER --
mainloop()
exitonclick()
```

Pour la forme de la tortue, la fonction shape() peut prendre en argument les chaines de caractères :

- ''turtle''
- ''square''
- ''circle''

Il est possible de remplacer l'icône de la tortue par une image au format .gif .

⇨ Télécharger le fichier *tp6_images.zip* dans lequel vous trouverez les 3 images .gif utilisées dans la suite et les copier **dans le répertoire** ou est **enregistré** votre fichier .py (***il est nécessaire de l'enregistrer***).

Pour remplacer l'icône par une image, on exécute la fonction *Screen().addshape()* , puis on met en argument de la fonction *shape()* le nom du fichier :

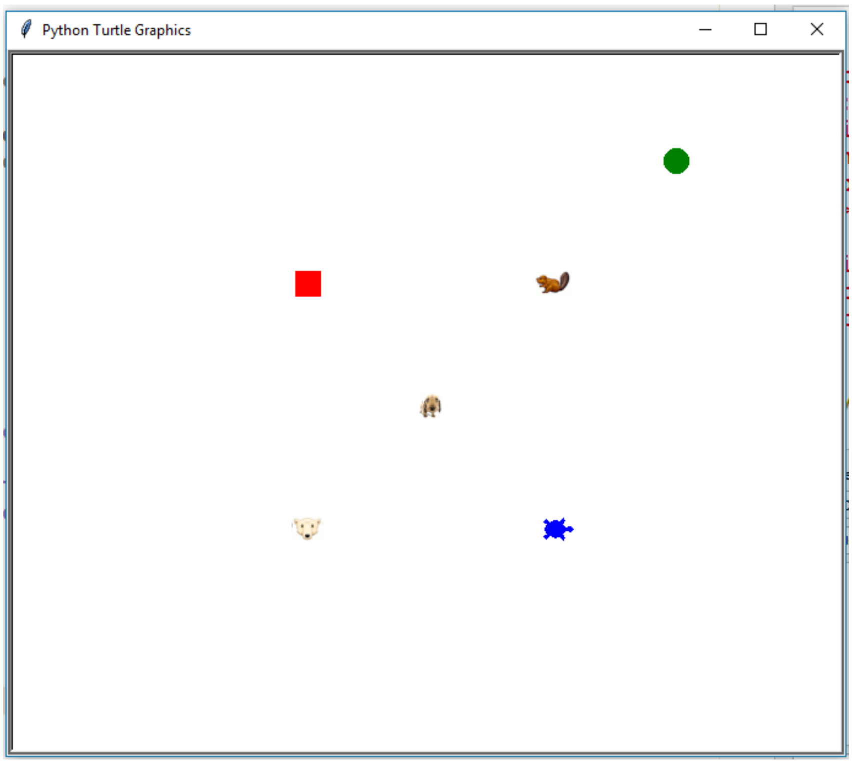
```
from turtle import *

t1 = Turtle()
Screen().addshape("ecureuil.gif")
t1.shape("ecureuil.gif")


# -- A LAISSER --
mainloop()
exitonclick()
```

⇨ Ecrire ces différents bouts de code et tester le bon fonctionnement de chacun d'eux. Insérer à chaque fois une copie d'écran dans le compte-rendu.

4. Ecriture d'une fonction qui crée une tortue ?

```
from turtle import *

# -- FONCTIONS --
def creation_tortues(forme,couleur,x,y) :









# -- MAIN --
hector = creation_tortues("ecureuil.gif","",100,100)
guillaume = creation_tortues("chien.gif","",0,0)
yann = creation_tortues("ours.gif","",-100,-100)
antoine = creation_tortues("turtle","blue",100,-100)
quentin = creation_tortues("square","red",-100,100)
agatha = creation_tortues("circle","green",200,200)

# -- A LAISSER --
listen()
mainloop()
exitonclick()
```

Le code ci-dessus est incomplet. En l'exécutant, on obtient la fenêtre graphique *Turtle* ci-contre :



⇨ Compléter ce code afin d'obtenir le fonctionnement souhaité.

5. Ecriture d'une fonction qui permet de déplacer une tortue :

⇨ Si on reprend le fichier réalisé précédemment et que l'on écrit le script suivant dans la partie programme principal, que se passe-t-il ? Essayer et donner une copie d'écran de la fenêtre graphique.

```
28 # -- MAIN --
29 t = creation_tortues("turtle","red",100,100)
30 dx = 3
31 dy = 2
32
33 for i in range(10000) :
34     x,y = t.pos()
35     if x < -200 or x > 200 :
36         dx = - dx
37         t.left(360)
38     if y < -200 or y > 200 :
39         dy = - dy
40         t.left(360)
41     x = x + dx
42     y = y + dy
43     t.goto(x,y)
```

⇨ On veut réaliser à présent les mêmes actions, mais en insérant à présent ce bout de code dans une fonction *deplac()* . Compléter le script de cette fonction et modifier le code global.

```
15 def deplac(t,dx,dy,limite,angle) :
16
17
18
19
20
21
22
23
24
25
26
27
28 # -- MAIN --
29 t = creation_tortues("turtle","red",100,100)
30 dx_t = 3
31 dy_t = 2
32
33
34 for i in range(10000) :
35    dx_t , dy_t = deplac(t,dx_t,dy_t,200,360)
36
37
38 # -- A LAISSER --
39 listen()
40 mainloop()
41 exitonclick()
```

6. Ecriture d'une fonction qui permet de déplacer plusieurs tortues :

⇨ La tortue créée dans le programme principal peut avoir n'importe quel nom. Par exemple, on peut écrire :

```
28 # -- MAIN --
29 mickael = creation_tortues("ecureuil.gif","",100,100)
30
31 dx_mickael , dy_mickael = 2 , 3
32
33 for i in range(10000) :
34    dx_mickael , dy_mickael = deplac(mickael,dx_mickael,dy_mickael,200,360)
```

⇨ Modifier le code de l'exercice précédent, afin qu'il puisse faire apparaitre les tortues ''*chien.gif*'' et ''*turtle*'' qui se déplace dans un carré de 100 px de coté pour la tortue ''chien.gif'' et 250px de coté pour la tortue ''*turtle*''. Donner une copie d'écran du code correspondant et du résultat obtenu dans la fenêtre graphique.

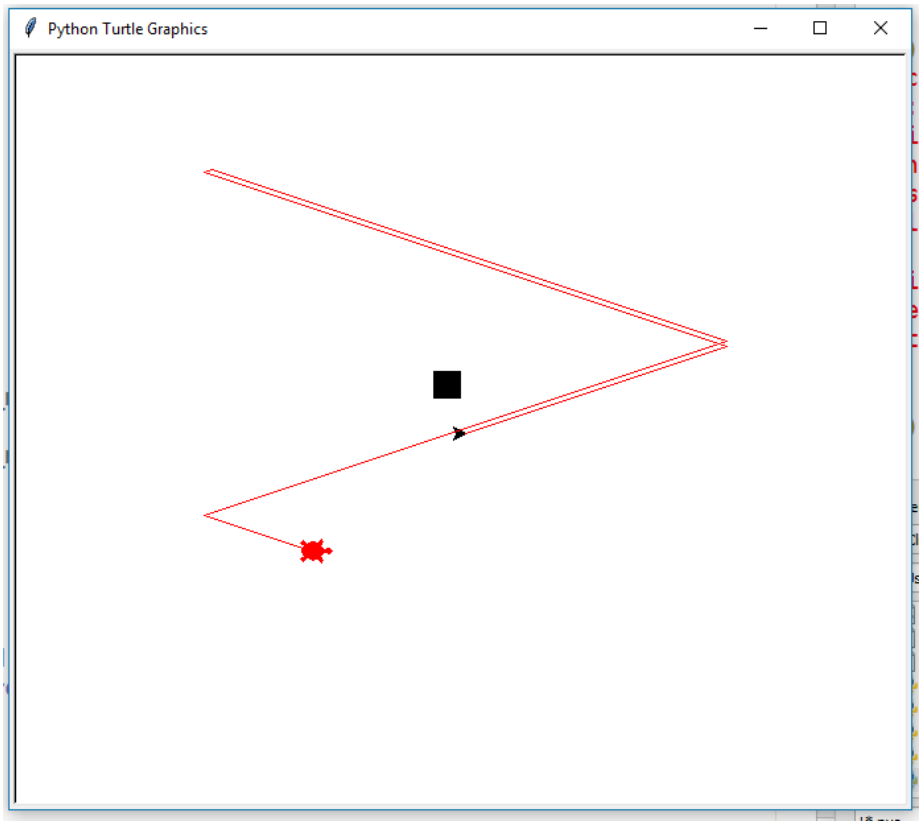
7. Réalisation d'un jeu :

Cet exercice permettra de réaliser une synthèse de tout ce qui a été vu précédemment.

En utilisant les fonctions :

- *ecart()* de l'exercice 1,
- *direct(x,y)* de l'exercice 2,
- *creation_tortues()* de l'exercice 4,
- *deplac()* de l'exercice 6,

on vous demande de réaliser un jeu dans lequel une tortue rouge se déplace automatiquement.



Une autre tortue de couleur noire et de forme carrée peut être déplacée par click sur la fenêtre graphique.

Si le joueur arrive à cliquer sur la tortue en mouvement, le jeu s'arrête et est gagné. On arrête alors le déplacement par un ***break*** dans la *boucle for _ in range()*. On écrit aussi dans le shell, le message :

PARTIE GAGNEE

Enregistrer le script dans un fichier nommé *exercice_7.py*

Conseils pour y arriver :

- Construire le code petit à petit en rajoutant à chaque fois une fonctionnalité. A chaque petite avancée, le tester pour vérifier que le résultat est celui attendu. Ne continuer que si c'est le cas.

- Il est difficile de cliquer exactement sur la tortue en mouvement. On considèrera que le click est bien placé si par exemple, l'écart entre les abscisses des 2 objets et entre leurs ordonnées est inférieur à 2 px. C'est là que vous utiliserez la fonction *ecart()* écrite au début du Tp.