

Pygame 101

La librairie `pygame` est une librairie qui permet de construire facilement des jeux avec interface graphique en Python. Ce document vous présente les bases à savoir pour vous mettre dans le bain rapidement sans avoir à chercher dans la documentation.

Aide

La documentation complète (en anglais, cependant) est disponible [ici](#).

1 Squelette de base

Code

```
import pygame

# pygame setup
pygame.init()
screen = pygame.display.set_mode((1280, 720))
clock = pygame.time.Clock()
ouvert = True

while ouvert:
    # recherche d'évènements
    # pygame.QUIT est l'évènement déclenché quand on clique sur la croix X
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            ouvert = False

    # On remplit l'écran avec une couleur douteuse
    screen.fill("purple")

    # On fait tourner le jeu

    clock.tick(60) # limite les FPS à 60

pygame.quit()
```

Un exemple de paramétrage basique de `pygame`

1.1 Gérer la fenêtre

Vous pouvez construire une fenêtre de taille **hauteur x largeur** et la stocker dans une variable **fenetre** avec la commande suivante:

Code

```
fenetre = pygame.display.set_mode((largeur, hauteur))
```

Vous pouvez gérer le titre de la fenêtre avec la commande:

Code

```
pygame.display.set_caption("votre titre")
```

Ce modèle vous permet d'avoir une fenêtre qui s'affiche en violet jusqu'à ce que vous cliquiez sur la croix en haut à droite. C'est bien, mais on veut maintenant pouvoir ajouter des éléments à l'intérieur de cette fenêtre.

2 Dessiner des formes

Au départ, tous les éléments que vous allez afficher à l'écran vont être constitués de combinaisons de formes géométriques de base. Voici comment les construire :

Ligne

Code

```
pygame.draw.line(
    fenetre,          # dans cette fenêtre
    couleur,          # dans cette couleur
    (x_deb, y_deb),   # depuis ce point
    (x_fin, y_fin),   # jusqu'à ce point
    largeur           # avec cette épaisseur de trait (opt)
)
```

Rectangle

Code

```
pygame.draw.rect(
    fenetre,          # dans cette fenêtre
    couleur,          # dans cette couleur
    (x, y, lg, ht),   # depuis (x, y) de largeur lg et hauteur ht
    largeur           # épaisseur du trait (opt)
)
```

Cercle

Code

```
pygame.draw.circle(
    fenetre, # dans cette fenêtre
    couleur, # dans cette couleur
    (x, y),  # les coordonnées du centre
    rayon,   # le rayon du cercle
    largeur  # l'épaisseur du trait (opt)
)
```

D'autres formes sont bien sûres disponibles [dans la documentation](#).

3 Réagir à des événements

Les jeux sont constitués d'événements divers qu'il faut pouvoir gérer. Ici, on se concentre sur les événements qui sont déclenchés à la suite d'une action de l'utilisateur et notamment un appui sur une touche du clavier.

Pour gérer les événements, on va d'abord récupérer la liste des événements qui ont été déclenchés depuis le dernier tour d'horloge et les traiter ensuite un par un.

La liste des évènements est disponible à l'aide de la commande :

Code

```
pygame.event.get()
```

que vous utiliserez cependant plus régulièrement directement dans une boucle :

Code

```
for event in pygame.event.get():  
    # code à exécuter
```

Ensuite, on cherche en général à analyser le type de l'évènement puis son intitulé exact. Le squelette de base vous montre comment gérer l'évènement déclenché par un clic sur la croix de fermeture. Il en existe cependant d'autres types, et notamment :

Code

```
pygame.KEYDOWN          # on appuie sur une touche clavier  
pygame.KEYUP            # on relache une touche clavier  
pygame.MOUSEMOTION      # on bouge la souris  
pygame.MOUSEBUTTONUP     # on relache un bouton souris  
pygame.MOUSEBUTTONDOWN  # on appuie sur un bouton souris
```

Quand vous cherchez à savoir sur quelle touche l'utilisateur a appuyé, vous pouvez comparer la valeur de l'évènement aux touches qui vous intéressent. Par exemple :

Code

```
if event.key == pygame.K_LEFT :  
    # code à exécuter
```

Un index complet des touches de clavier que vous pouvez utiliser est disponible en annexe (section 4).

4 Annexe - Codes des touches clavier

</> Tableau des touches

pygame Constant	ASCII	Description

K_BACKSPACE	\b	backspace
K_TAB	\t	tab
K_CLEAR		clear
K_RETURN	\r	return
K_PAUSE		pause
K_ESCAPE	^[escape
K_SPACE		space
K_EXCLAIM	!	exclaim
K_QUOTEDBL	"	quotedbl
K_HASH	#	hash
K_DOLLAR	\$	dollar
K_AMPERSAND	&	ampersand
K_QUOTE		quote
K_LEFTPAREN	(left parenthesis
K_RIGHTPAREN)	right parenthesis
K_ASTERISK	*	asterisk
K_PLUS	+	plus sign
K_COMMA	,	comma
K_MINUS	-	minus sign
K_PERIOD	.	period
K_SLASH	/	forward slash
K_0	0	0
K_1	1	1
(...)	(.)	(.)
K_9	9	9
K_COLON	:	colon
K_SEMICOLON	;	semicolon
K_LESS	<	less-than sign
K_EQUALS	=	equals sign
K_GREATER	>	greater-than sign
K_QUESTION	?	question mark
K_AT	@	at
K_LEFTBRACKET	[left bracket
K_BACKSLASH	\	backslash
K_RIGHTBRACKET]	right bracket
K_CARET	^	caret
K_UNDERSCORE	_	underscore
K_BACKQUOTE	`	grave
K_a	a	a
K_b	b	b
(...)	(.)	(.)
K_z	z	z
K_DELETE		delete
K_KP0		keypad 0
(...)	(..)	(..)
K_KP9		keypad 9
K_KP_PERIOD	.	keypad period

K_KP_DIVIDE	/	keypad divide
K_KP_MULTIPLY	*	keypad multiply
K_KP_MINUS	-	keypad minus
K_KP_PLUS	+	keypad plus
K_KP_ENTER	\r	keypad enter
K_KP_EQUALS	=	keypad equals
K_UP		up arrow
K_DOWN		down arrow
K_RIGHT		right arrow
K_LEFT		left arrow
K_INSERT		insert
K_HOME		home
K_END		end
K_PAGEUP		page up
K_PAGEDOWN		page down
K_F1		F1
K_F2		F2
(...)		(..)
K_F15		F15
K_NUMLOCK		numlock
K_CAPSLOCK		capslock
K_SCROLLLOCK		scrollock
K_RSHIFT		right shift
K_LSHIFT		left shift
K_RCTRL		right control
K_LCTRL		left control
K_RALT		right alt
K_LALT		left alt
K_RMETA		right meta
K_LMETA		left meta
K_LSUPER		left Windows key
K_RSUPER		right Windows key
K_MODE		mode shift
K_HELP		help
K_PRINT		print screen
K_SYSREQ		sysrq
K_BREAK		break
K_MENU		menu
K_POWER		power
K_EURO		Euro
K_AC_BACK		Android back button