DRONES

Construcció d'una Estació de Control

Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

ICARUS Research Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

1

Python

python

- Python es un llenguatge de programació molt senzill i pensat per ensenyar i per prototip ràpid. Es fa servir molt en intel·ligència artificial i robòtica.
  - https://www.python.org/
  - https://thonny.org/
  - Nosaltres el farem servir per desenvolupar la nostra estació de control.

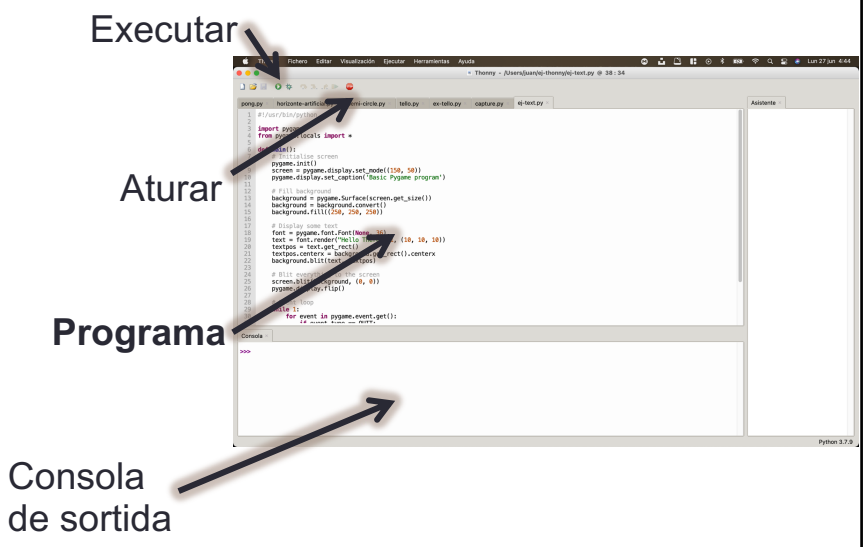
2

Que és un programa?

- Es una “recepta de cuina”
- Li diem a l’ordinador pas a pas que ha de fer
- L’ordinador no s’inventa res… només farà exactament el que nosaltres li diguem.

- En el nostre cas li ensenyarem a que es comuniqui amb el ARDrone i visualitzi els instruments com si estiguessim a bord del drone.

3



4

Primer exemple de programa

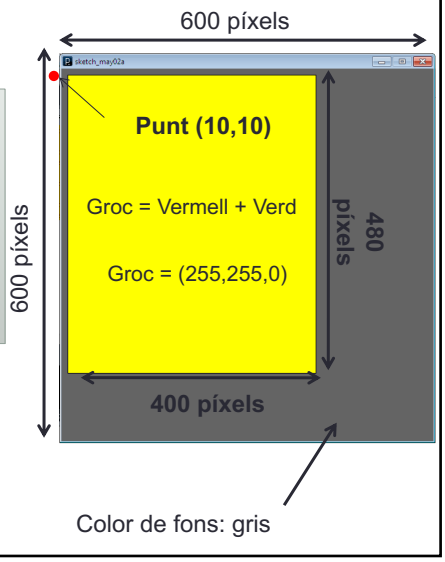
- Dibuix d'un quadrat

```
import pygame

pygame.init()
screen = pygame.display.set_mode((600, 600))

# Draw
screen.fill((0, 0, 0))
pygame.draw.rect(screen, (255, 255, 0), [10, 10, 400,480])

# Blit everything
pygame.display.flip()
```



5

Analitzem el programa (1)

```
import pygame

pygame.init()
screen = pygame.display.set_mode((600, 600))
```

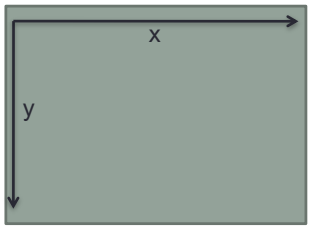
En aquest tros de codi estem:

1- Indicant-li a Python que farem servir la llibreria pygame
2- La arranquem
3- Preparem la nostra pantalla amb una mida de 600 x 600

6

Sistemes de Coordenades

- L'ordinador fa servir un sistema de coordenades cartesianes per poder dibuixar a pantalla.



(width-1,height-1)

7

Analitzem el programa (2)

```
import pygame

pygame.init()
screen = pygame.display.set_mode((600, 600))

# Draw
screen.fill((0, 0, 0))
pygame.draw.rect(screen, (255, 255, 0), [10, 10, 400,480])
```

- Omplim la pantalla del color (0,0,0). Quin serà?
- Fixeu-vos que les linies que comencem amb # son comentaris, serveixen perque despres nosaltres, no l'ordinador, ens enrecordem que feia aquest tros de codi

8

Color

- Un color s'emmagatzema a la ordinador com 3 enters entre 0 i 255 on cada numero representa el % d'un color primari: vermell, verd i blau.
  - 0 equival a 0% d'aquest color i 255 a 100%.
- Si voleu buscar un color concret podeu buscar "color picker" a Internet. Els valors RGB que us sortin són els percentatges de vermell, verd i blau.

9

Analitzem el programa (3)

```
import pygame

pygame.init()
screen = pygame.display.set_mode((600, 600))

# Draw
screen.fill((0, 0, 0))
pygame.draw.rect(screen, (255, 255, 0), [10, 10, 400,480])
```

- Dibuixarem un rectangle amb pygame.draw.rect.
- Pygame te més ordres per dibuixar altres coses que veurem mes endavant
- Els 3 primers números entre parentesi són el color
- Els 4 segons números entre claus son un "rectangle":

  [coordenadaX, coordenadaY, ample, alt]

10

Analitzem el programa (4)

```
import pygame

pygame.init()
screen = pygame.display.set_mode((600, 600))

# Draw
screen.fill((0, 0, 0))
pygame.draw.rect(screen, (255, 255, 0), [10, 10, 400,480])

# Blit everything
pygame.display.flip()
```

- Finalment hem de posar tot el que hem dibuixat a la superficie a la pantalla, aquesta operación es diu blit en angles.

11

Ordres per dibuixar

Linia

```
pygame.draw.line(superficie, color, (x1, y1), (x2, y2))
```

Circle

```
pygame.draw.circle(superficie, color, (x, y), radi)
```

Elipse

```
pygame.draw.ellipse(superficie,color,(x1,y1,ample,alt))
```

Rectangle

```
pygame.draw.rect(superficie,color,(x1,y1,ample,alt))
```

12

Ordres per dibuixar només llindar

Les ordres d'abans ens dibuixaven figures solides, si Nomes volem el llindar hem d'afegir el gruix de la línea.

Circle

```
pygame.draw.circle(superficie,color,(x,y),radi,gruix)
```

Elipse

```
pygame.draw.ellipse(superficie,color,(x1,y1,ample,alt),gruix)
```

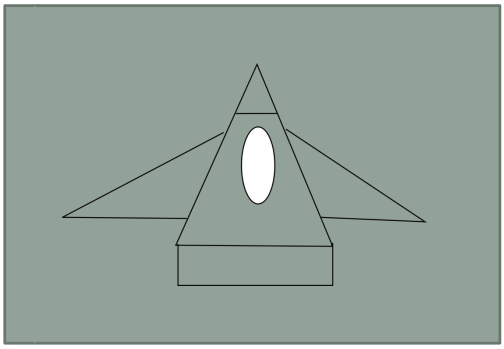
Rectangle

```
pygame.draw.rect(superficie,color,(x1,y1,ample,alt),gruix)
```

13

Activitat 1

- Farem que l'ordinador dibuixi aquest avió.



14

Nova versió del codi per dibuixar

- Si ens fixem la primera versió es quedava “tonta”
- Hem de dir-li al Python com volem que aturi el programa, farem servir una nova llibreria “app”.
- I posarem el nostre codi per dibuixar a un bloc de codi que anomenem funció

```
import app
import pygame
from pygame.locals import *

def draw():
    screen.fill((0, 0, 0))
    pygame.draw.rect(screen, (255, 255, 0), [10, 10, 400,480])

pygame.init()
screen = pygame.display.set_mode((600, 600))
app.run(draw)
```

15

```
import sys
import pygame

from pygame.locals import *

def run(draw):
    running = True

    while running:
        # Wait for events
        for event in pygame.event.get():
            if event.type == QUIT:
                running = False

        # Draw
        draw()

        # Blit everything
        pygame.display.flip()

    pygame.quit()
    sys.exit()
```

16

Variables

- És una “capsa” on l’ordinador pot guardar una dada.
- Te un nom per poder-la identificar.
- I un valor que pot ser:
  - Un numero: 23
  - Una lletra: ’a’
  - Una frase ”la a es una lletra”
  - Cert o fals, p.e running
- Les definirem al principi del nostre programa.

```
Numero = 23
lletra = “a”
frase = “la a es una lletra”
Running = True
```

17

Una animació

- Si volem fer-les servir dins d’un bloc haurem de posar: `global nomVariable`
- Podrem posar el nom d’una variable allá on posavem un numero
- Tambe podem canviar el valor d’una variable amb un de nou

```
import app
import pygame
from pygame.locals import *


# Variables
y = 0

pygame.init()
screen = pygame.display.set_mode((600, 600))

def draw():
    global y
    screen.fill((0,0,0));
    pygame.draw.line(screen, (255, 0, 0), [0, y], [600, y])
    y = y + 1

app.run(draw)
```

18

Activitat 2

- 1. Modificarem l'animació anterior perquè la línea vagi de dalt a baix?
- 2. Com faríem perquè anés d'esquerra a dreta?
- 3. I perquè anés canviant de color?

19

Condicions

- Podem fer que l'ordinador calculi quelcom i decideixi si ha de fer una cosa o un altre.

```
x = 2

if x+5 < 10:
    print("El resultat es mes petit que 10")
else:
    print("El resultat es mes gran que 10")
```

- Es molt important posar correctament els espais!!

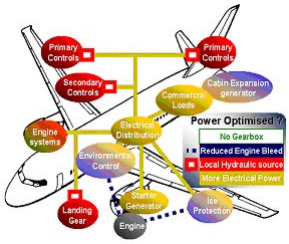
20

Activitat 3

- 1. Modifica el programa anterior perquè al arribar al final de l'animació torni al principi.

21

Aviònica



- Fa referència als sistemes de comunicacions, sistemes de navegació, indicadors i elements de control de la nau.
- En general, "l'aviònica" d'una aeronau inclou:
  - Des de qualsevol LED o sensor als sistemes de pilot automàtic.
  - Sistemes de comunicacions, navegació, monitoratge, controls de l'aeronau, "Collision avoidance", caixes negres, meteorologia i Flight Management System.
- Tota l'informació es plasma en el Cockpit

22

Aviònica (Cockpit)

Space Shuttle, NASA

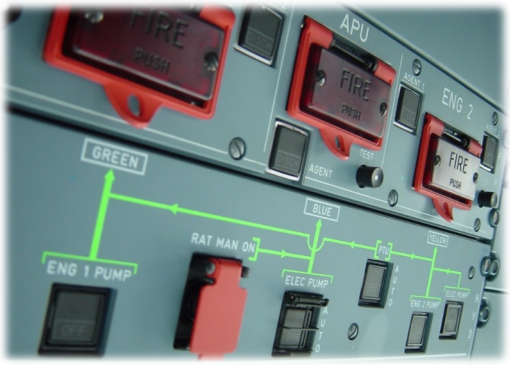
23

Aviònica (Cockpit)

A320, Airbus

24

Aviònica

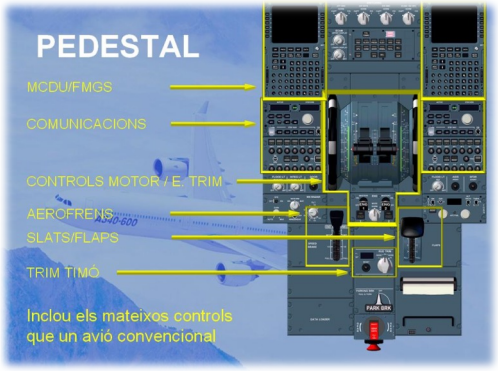


25

Aviònica

26

Aviònica



27

Horitzó Artificial

- Mostra l'actitud que té l'avió respecte l'horitzó. A partir d'aquest, el pilot sap com s'està comportant l'aeronau.
- Podem obtenir els angles de pitch i roll (balanceig i capcineig)

28

Horitzó Artificial

- El pilot coneix com està encarat l'avió respecte un horitzó fictici que representa l'estabilitat, paral·lel al terra i amb inclinació de 0 graus.
- L'horitzó està representat de manera fixa amb la línia horitzontal vermella.
- L'esfera interior mostra l'angle de capcineig i l'exterior el de balanceig.

29

DJITelloPy

- Comunicació amb Tello
- Com obtenir comunicació amb el drone:
  - Arrancar el drone
  - Des del PC, establim connexió amb el Wi-Fi del drone

```
from djitellopy import tello

me = tello.Tello()
me.connect()

print(me.get_battery())
```

Obtenim angles de l'avió, GPS, bateria, etc.

30

Obtenció dels angles del Tello

- Informació rellevant:
  - Pitch, roll & yaw
  - Estat de bateria
  - Velocitat horitzontal i vertical
  - Altitud
- Aquesta informació l'obtenim dels *sensors*:
  - Tenim informació dels 6 graus de llibertat

```
from djitellopy import tello

me = tello.Tello()
me.connect()

battery = me.get_battery()
roll = me.get_roll()
pitch = me.get_pitch()
yaw = me.get_yaw()
height = me.get_height()

speedX = me.get_speed_x()
speedY = me.get_speed_y()
speedZ = me.get_speed_z()
```

31

Obtenció dels angles del Tello

```
import app
import pygame
from pygame.locals import *
from djitellopy import tello

me = tello.Tello()
me.connect()

pygame.init()
screen = pygame.display.set_mode((600, 600))
font = pygame.freetype.SysFont("Arial", 12)

def draw():
    global battery, roll, pitch, yaw, height

    # Get data
    battery = me.get_battery()
    roll = me.get_roll()
    pitch = me.get_pitch()
    yaw = me.get_yaw()
    height = me.get_height()

    # Draw
    screen.fill((0, 0, 0))
    font.render_to(screen, (40, 40), f"Battery: {battery}", (0, 255, 0))
    font.render_to(screen, (40, 60), f"Roll: {roll}", (0, 255, 0))
    font.render_to(screen, (40, 80), f"Pitch: {roll}", (0, 255, 0))
    font.render_to(screen, (40, 100), f"Yaw: {roll}", (0, 255, 0))
    font.render_to(screen, (40, 120), f"Height: {height}", (0, 255, 0))

app.run(draw)
```

32

Activitat 4

- Fem un indicador de batería per la nostra estació de terra
  - La llargaria del rectangle ha de ser proporcional a la bateria
  - Podem posar numeros per facilitar la lectura
  - Podem canviar el color quan baixi d'un cert nivell per avisar.

33

Moure el tello automaticament

```
from djitellopy import tello

# Arrancar comunicacions amb Tello
me = tello.Tello()
me.connect()

# Obtenir bateria
battery = me.get_battery()
print(battery)

# Enlairament, moviments i aterrar
me.takeoff()
me.move_forward(50)
me.move_back(50)
me.land()
```

34