



Manual del profesor:

Contiene las instrucciones para poner en marcha los entornos de trabajo necesarios para Microsoft Windows. También la información técnicas del código Edu-Dron.

1. Utilidad para docentes de Robótica Educacional.
2. Permite ‘programar-jugar’ un dron en distintos entornos de **Microsoft AirSim**, un ‘plugin’ gratuito que se ejecuta en la plataforma de juegos **Unreal Engine** (***no es necesario instalar la plataforma***).
3. Nace de un Trabajo Fin de Máster de la **Universidad de León** (*“Máster en Robótica y Sistemas Inteligentes”*).
4. Esta organizado de forma muy sencilla, de manera que **NO** se requieren conocimientos técnicos por parte del Docente.
5. Está basado en producto **Open Source** por lo que **NO** tiene ningún coste.
6. Esta versión está pensada para ser ejecutada en el sistema operativo **Microsoft Windows 10**.
7. La ‘Api’ es muy sencilla y está basada en **Python**. Aunque se puede iniciar desde un nivel básico que no requiere conocimientos de Python. Puede ser modificada y ampliada por cada Docente o por los alumnos. El límite es el propio producto de Microsoft.
8. En esta serie de videos se explica todo paso a paso (con algunas alternativas) y con contiene un “Manual del Profesor” en PDF.
9. Esto es un primer paso, que podemos ir ampliando según nuestras posibilidades.

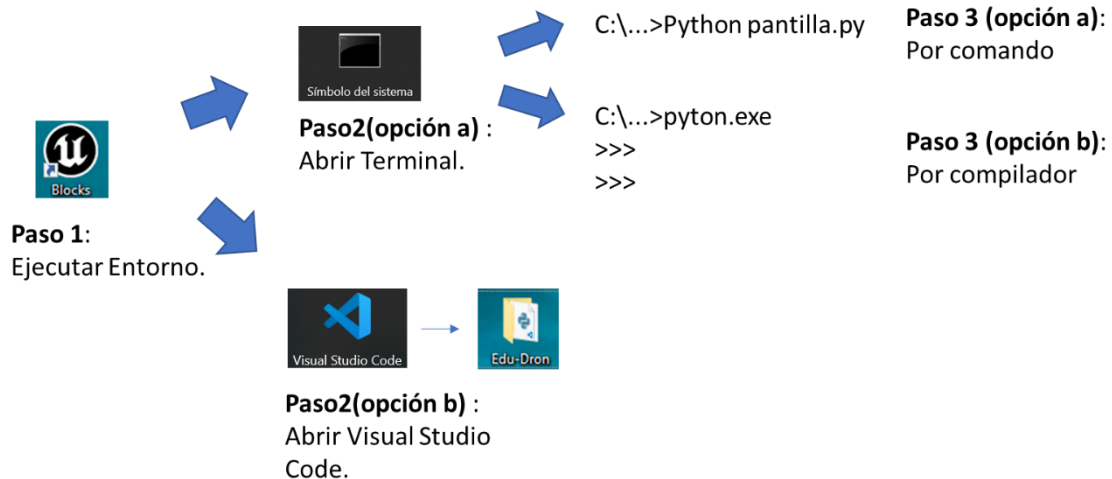


1.- Esquemas y videos:

1.1.- Esquema general de lo que necesitamos:



1.2.- Esquema de ejecución:



1.3.- Videos:

N.	Descripción	Link
1	Introducción	https://www.youtube.com/watch?v=7ywabVjzQF0
2	Instalación Entornos AirSim.	https://youtu.be/kOawHHWgkUo
3	Instalación Python (opción sin Visual Studio Code).	https://youtu.be/aQCRuMWiV_Q
4	Instalación AirSim + Requerimientos	https://youtu.be/igjiwGFyvfM
5	Instalación Edu-Dron	https://youtu.be/-SPRkelXOrM
6	Ejecución entorno y Edu-dron. (desde terminal)	https://youtu.be/pH-IKLbc3-4
7	Instalación Visual Studio + Python	https://youtu.be/J3JcXuviMUI
8	Ejecución entorno y Edu-dron. (desde Visual Studio Code).	https://youtu.be/kHtznzvgOw



2.- Instalación

2.1.- Instalación requerimientos técnicos AirSim.

Paquete	Comando
msgpack-rpc-python	pip install msgpack-rpc-python
numpy	pip install numpy
PyAudio	pip install PyAudio
opencv-contrib-python	pip install opencv-contrib-python
airsim	pip install airsimsim

2.2.- Instalación Python.

<https://www.python.org/downloads/windows/>

2.3.- Opción con Visual Studio Code:

<https://docs.microsoft.com/es-es/windows/python/beginners/>

2.4.- Instalación Edu-Dron.

<https://github.com/iarobotic/edu-dron>

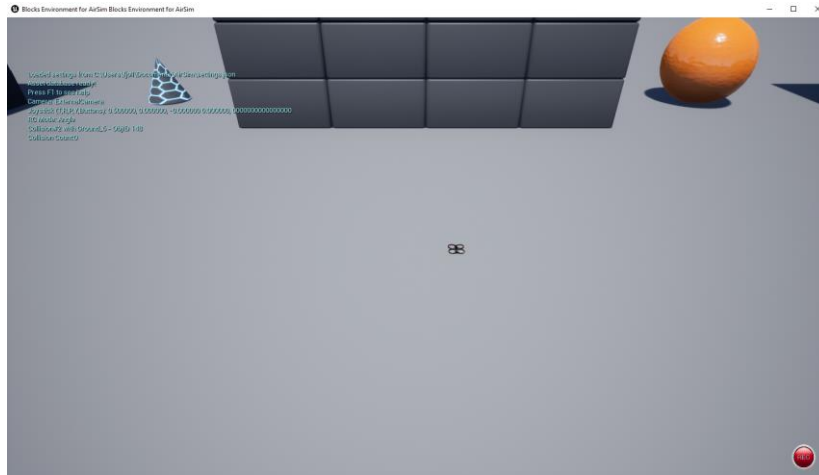
2.5.- Descarga de los mundos ya contruidos.

<https://github.com/Microsoft/AirSim/releases>



3.- Ejecución de los entornos

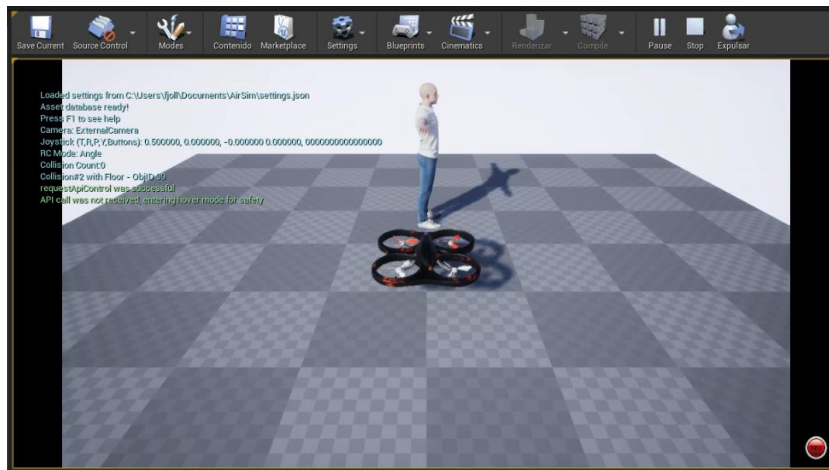
Entorno de Blocks.



Entorno de City.



Entorno propio.





4.- Ejecución Código propio

Dentro de la carpeta Edu-Dron encontrarás el fichero ***plantilla.py*** que es una plantilla de ejemplo.

La carpeta ***api***, contiene el fichero ***dron.py*** que es la api (interface) necesario para ejecutar la plantilla. El fichero `__init__.py` y la carpeta `__pycache__` son estructura técnica y no deben modificarse.

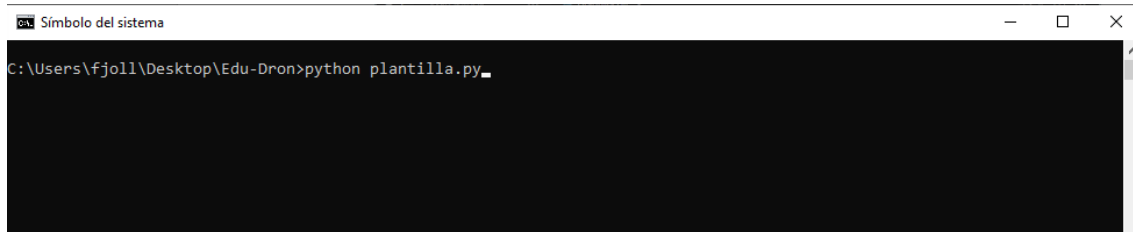


4.1.- EJECUCIÓN DEL PROGRAMA:

Una vez iniciado el entorno de Airsim (Unreal Engine) que queramos, y cuando ya aparezca el dron en nuestra pantalla, podremos ejecutar el código.

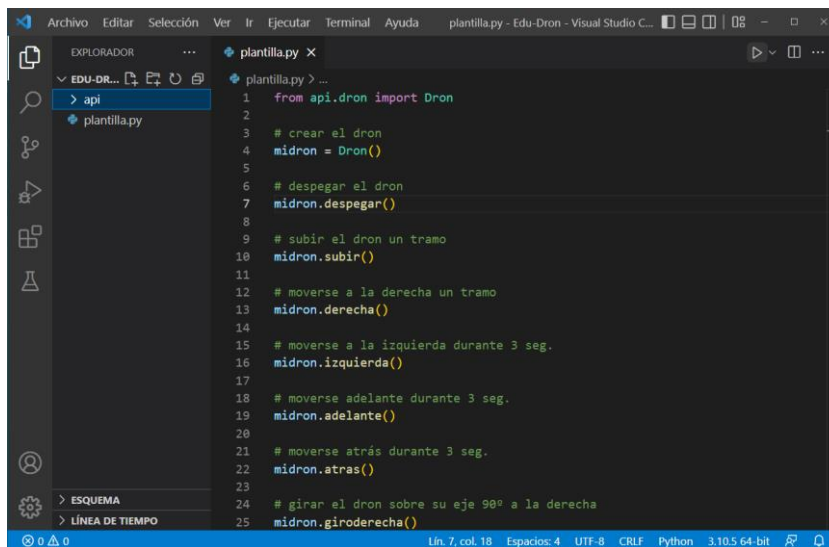
La ejecución del código puede hacerse desde un editor de código o directamente desde una 'consola de windows'.

Opción 1: Desde consola de Windows:



```
Símbolo del sistema
C:\Users\fjoll\Desktop\Edu-Dron>python plantilla.py
```

Opción 2: Aquí vemos ejecutándolo desde [Visual Studio Code] (ver instrucciones de instalación):



```
plantilla.py
1 from api.dron import Dron
2
3 # crear el dron
4 midron = Dron()
5
6 # despegar el dron
7 midron.despegar()
8
9 # subir el dron un tramo
10 midron.subir()
11
12 # moverse a la derecha un tramo
13 midron.derecha()
14
15 # moverse a la izquierda durante 3 seg.
16 midron.izquierda()
17
18 # moverse adelante durante 3 seg.
19 midron.adelante()
20
21 # moverse atrás durante 3 seg.
22 midron.atras()
23
24 # girar el dron sobre su eje 90º a la derecha
25 midron.giroderecha()
```



Explicación fichero plantilla.py, es un ejemplo de ejecución y consta de :

```
from api.dron import Dron
```

es la forma de indicar que se va a utilizar la api y el fichero dron, y lo renombramos como 'Dron'. Esta línea es obligatoria para todas las plantillas.

1.- Lo primero que necesitamos es 'crear un dron', para eso usamos :

```
# crear el dron  
midron = Dron()
```

Puedes usar otro nombre distinto a 'midron', pero tendrás que adaptar todas las llamadas a ese nuevo nombre.

2.- Despegar el dron. Suele ser la primer instrucción una vez creado (aunque podría usarse varias veces en un mismo programa).

```
# despegar el dron  
midron.despegar()
```

3.- Aterrizar el dron. Suele ser la última instrucción del programa, que haga uso del dron.

```
# aterrizar el dron  
midron.aterizar()
```

4.- El resto de instrucciones se pueden colocar en el orden que se quiera, según el propósito del programa a ejecutar. Y son:

```
# subir el dron un tramo  
midron.subir()  
  
# moverse a la derecha un tramo  
midron.derecha()  
  
# moverse a la izquierda durante 3 seg.  
midron.izquierda()  
  
# moverse adelante durante 3 seg.  
midron.adelante()  
  
# moverse atrás durante 3 seg.  
midron.atras()  
  
# girar el dron sobre su eje 90º a la derecha  
midron.giroderecha()  
  
# girar el dron sobre su eje 90º a la izquierda  
midron.giroizquierda()
```



Nota1: Podemos crear tantos programas como queramos usando esta plantilla, pero creándolos dentro de la carpeta principal 'Edu-Dron'.

Nota2: Podemos incluir las instrucciones que queramos, y por supuesto código Python para complementar nuestros programas.

Modificaciones que puedes realizar como ejercicios:

- 1.- Trazar un cuadrado con el dron.
- 2.- Hacer un trazado esquivando bloques.
- 3.- Crear un recorrido de ida para el dron, y que cree de forma automática el 'regreso' (vuelta a casa).



4.2.- MODIFICAR, AMPLIAR LAS FUNCIONALIDADES DEL DRON:

Para esto será necesario algunos conocimiento de Python, y modificaremos el fichero 'dron.py' (que está creado como una Clase Python).

Si vemos el código, comprobamos fácilmente que hay valores de algunas variables que son fijas y que están en castellano. De esas variables no debes modificar (a menos que se haya entendido su significado) las denominadas vx, vy, self.z que representan los ejes de coordenadas del dron y que le indican cómo debe moverse en el espacio.

Veamos cada una de los métodos (empiezan con 'def') que tenemos:

despegar	No tiene detalles sobre funcionamiento. Despega el dron.
aterrizar	No tiene detalle sobre funcionamiento. Aterriza el dron.
subir	Permite al dron subir 7 unidades hasta un máximo de 100. (puedes modificar y probar). Con una duracion de 1 seg y una velocidad de 0.3 (0 a 1). Puedes modificar y probar. VX y VY están a cero para que sólo se desplace en el eje vertical. time.sleep : demora, para esperar que el dron realice la maniobra. Puede adaptarse si se modificaron duracion y/o velocidad .
bajar	Permite al dron bajar 7 unidades hasta un mínimo de 1. (puedes modificar y probar). Con una duracion de 1 seg y una velocidad de 0.3 (0 a 1). Puedes modificar y probar. VX y VY están a cero para que sólo se desplace en el eje vertical. time.sleep : demora, para esperar que el dron realice la maniobra. Puede adaptarse si se modificaron duracion y/o velocidad .
derecha	Permite al dron moverse a la derecha durante 3 segundos (puedes modificar y probar). Con una duracion de 3 seg y una velocidad de 2 (0 a 1). Puedes modificar y probar. VX están a cero para que no se desplace hacia adelante. VY con la velocidad solicitada en negativo. time.sleep : demora, para esperar que el dron realice la maniobra. Puede adaptarse si se modificaron duracion y/o velocidad .
izquierda	Permite al dron moverse a la izquierda durante 3 segundos (puedes modificar y probar). Con una duracion de 3 seg y una velocidad de 2 (0 a 1). Puedes modificar y probar. VX están a cero para que no se desplace hacia adelante. VY con la velocidad solicitada en positivo . time.sleep : demora, para esperar que el dron realice la maniobra. Puede adaptarse si se modificaron duracion y/o velocidad .
adelante	Permite al dron moverse adelante durante 3 segundos (puedes modificar y probar). Con una duracion de 3 seg y una velocidad de 2 (0 a 1). Puedes modificar y probar. VY están a cero para que no se desplace lateralmente. VX con la velocidad solicitada en positivo . time.sleep : demora, para esperar que el dron realice la maniobra. Puede adaptarse si se modificaron duracion y/o velocidad .
atras	Permite al dron moverse atrás durante 3 segundos (puedes modificar y probar). Con una duracion de 3 seg y una velocidad de 2 (0 a 1). Puedes modificar y probar. VY están a cero para que no se desplace lateralmente. VX con la velocidad solicitada en negativo . time.sleep : demora, para esperar que el dron realice la maniobra. Puede adaptarse si se modificaron duracion y/o velocidad .



giroderecha	<p>Permite al dron girar sobre su eje 90º (puedes modificar y probar) a la derecha. Con una duracion de 3 seg y una velocidad de 2 (0 a 1). Puedes modificar y probar. Si supera los 270º se pone a 0º.</p> <p>VX y VY están a cero para que sólo se desplace en ninguna dirección.</p> <p>time.sleep : demora, para esperar que el dron realice la maniobra. Puede adaptarse si se modificaron duracion y/o velocidad.</p>
giroizquierda	<p>Permite al dron girar sobre su eje 90º (puedes modificar y probar) a la izquierda. Con una duracion de 3 seg y una velocidad de 2 (0 a 1). Puedes modificar y probar. Si supera los 270º se pone a 0º.</p> <p>VX y VY están a cero para que sólo se desplace en ninguna dirección.</p> <p>time.sleep : demora, para esperar que el dron realice la maniobra. Puede adaptarse si se modificaron duracion y/o velocidad.</p>

Modificaciones que puedes realizar como ejercicios:

- 1.- Modificar los giros (a derecha e izquierda) indicándoles los grados a girar.
- 2.- Modificar los desplazamientos (adelante, atrás, derecha e izquierda) indicándoles un tiempo de desplazamiento.
- 3.- Modificar los desplazamientos (arriba, abajo) indicándoles un tramo de desplazamiento.
- 4.- Existen muchas más posibilidades que están recogidas en AirSim. Como pueden ser tomar fotos, conocer los datos de los sensores. Todo esto puedes encontrarlo en los enlaces siguientes:

<https://github.com/microsoft/AirSim/tree/main/PythonClient>

<https://github.com/Microsoft/AirSim>