AirSim

The next-gen simulator for Autonomous Systems

Contributors: Sotirios Papadopoulos, Yorgos Basioukas
Aristotle University of Thessaloniki

Artificial Intelligence &
Information Analysis Lab

Setting Up

- Unreal Engine 4.27
- Microsoft Visual Studio 2022 with Windows 10 SDK 10.0.19041 and the latest .NET Framework SDK.
- Binaries: https://microsoft.github.io/AirSim/use_precompiled/
- Build on windows: https://microsoft.github.io/AirSim/build_windows/
- Build on Linux: https://microsoft.github.io/AirSim/build_linux/

Connecting to Drone API

```
import airsim

client = airsim.MultirotorClient()  # connect to the AirSim drone simulator
client.confirmConnection()          # Checks state of connection every 1 sec
client.enableApiControl(True)       # Control drone from the API commands
client.armDisarm(True)              # Arm drone
```

Artificial Intelligence & Information Analysis Lab

Take off

```
client.takeoffAsync().join()        #.join() to wait for the task to complete

airsim.wait_key('Press any key to takeoff') # wait for an input and then
                                              take off
```

Artificial Intelligence & Information Analysis Lab

Landing

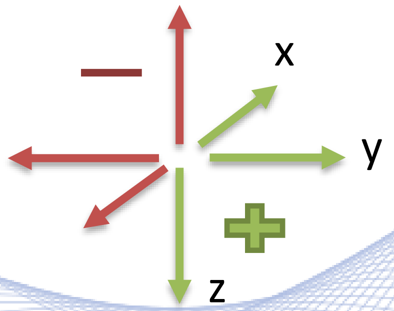
```
client.landAsync().join()             #wait for the task to complete

client.armDisarm(False)               # disarm drone
client.enableApiControl(False)        # deactivate API
```

Artificial Intelligence & Information Analysis Lab

NED coordinates

From a drone POV, most objects of interest are below the aircraft, so it is sensible to define down as a positive number. So NED in meters is the Coordinate system followed by airsim.



Start of the vehicle is always 0,0,0 in Ned

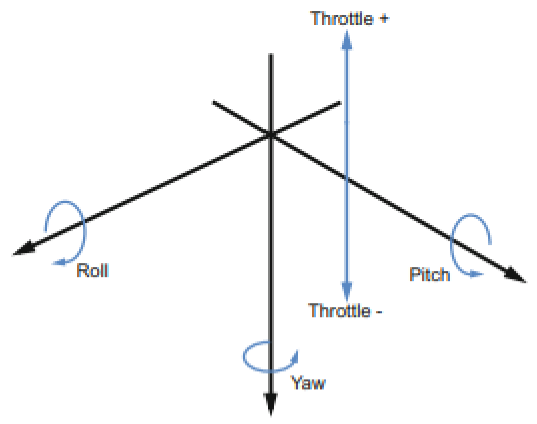
+X is North, +Y is East and +Z is Down

Moving the Drone

client.moveByAngleThrottle	
client.moveByAngleZ	*#rotate by angular velocity while holding the altitude at a fixed Z coordinate*
client.moveByVelocity	
client.moveByVelocityZ	*#moves the drone while holding the altitude at a fixed Z coordinate*
client.moveToPosition	*#moves the drone to the specified position in unreal engine*
client.moveToZ	*#change drone height*
client.moveOnPath	

Moving the Drone

Methods	Parameters
moveByAngle	pitch, roll, z, yaw, duration[a]
moveByVelocity	vx, vy, vz, duration, drivetrain[b], yaw_mode[c]
moveByPath	path, velocity, max_wait_seconds[d], drivetrain, yaw_mode, lookahead[e], adaptive_lookahead[e]
moveToPosition	x, y, z, velocity, max_wait_seconds, drivetrain, yaw_mode, lookahead, adaptive_lookahead
moveByManual	vx_max, vy_max, z_min, duration, drivetrain, yaw_mode



Getting Vehicle Data

getImuData()
getBarometerData()
getMagnetometerData()
getGpsData()
getLidarData()
getMultirotorState()
simGetObjectPose()
simGetCollisionInfo()

Camera API

Image Type field

Scene	= 0,
DepthPlanner	= 1,
DepthPerspective	= 2,
DepthVis	= 3,
DisparityNormalized	= 4,
Segmentation	= 5,
SurfaceNormals	= 6,
Infrared	= 7

Camera field

front_center	= 0
front_right	= 1
front_left	= 2
Fpv	= 3
back_center	= 4

Getting Images

There are 2 ways to get an image from the API:

simGetImage()
simGetImages()

Examples:

```
png_image = client.simGetImage("0", airsim.ImageType.Scene)

responses = client.simGetImages([
        airsim.ImageRequest(0, airsim.ImageType.Scene),                  # png format
        airsim.ImageRequest(1, airsim.ImageType.Scene, False, False),    # uncompressed RGB array bytes
        airsim.ImageRequest(1, airsim.ImageType.DepthPlanner, True)])    # floating point uncompressed image
```

Change Gimbal Settings

The *simSetCameraOrientation* sets the pose for the specified camera while taking an input camera and a quaternion in NED frame. The handy *airsim.to_quaternion()* function allows to convert pitch, roll, yaw to quaternion.

```
client.simSetCameraOrientation("0", airsim.to_quaternion(0.5, 0.5, 0.1))
```

Change Weather

By default all weather effects are disabled. To enable weather effect, first call:

```
simEnableWeather(True)
```

To change the weather from API:

```
client.simSetWeatherParameter(airsim.WeatherParameter.Rain, 0.25);
```

Example:

```
airsim.wait_key('Press any key to enable snow at 50%')
client.simSetWeatherParameter(airsim.WeatherParameter.Snow, 0.50);
```

Weather options

Rain, Snow, MapleLeaf, Dust, Fog

More examples:

```
client.simSetWeatherParameter(airsim.WeatherParameter.Rain, 0.25);
client.simSetWeatherParameter(airsim.WeatherParameter.Rain, 0.75);
client.simSetWeatherParameter(airsim.WeatherParameter.MapleLeaf, 0.50);
client.simSetWeatherParameter(airsim.WeatherParameter.Dust, 0.50);
client.simSetWeatherParameter(airsim.WeatherParameter.Fog, 0.50);
```

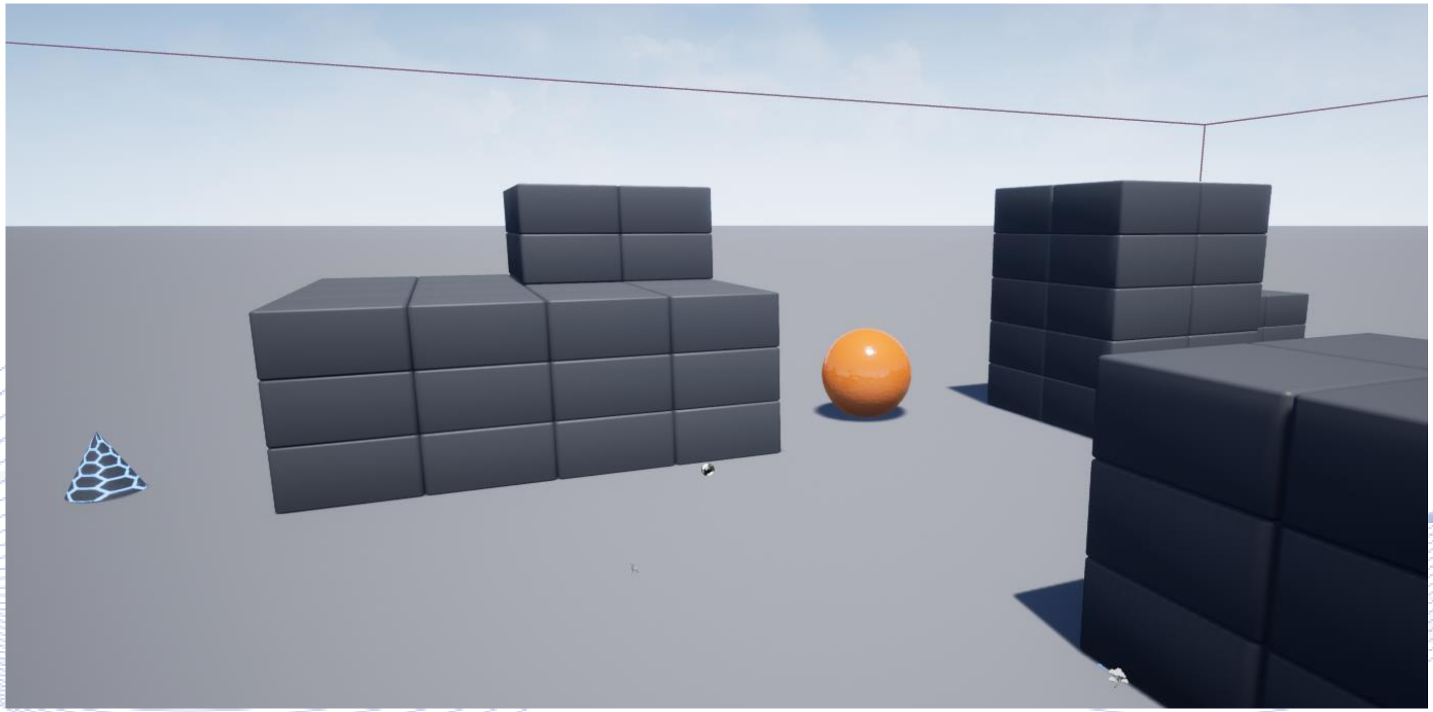
Other settings

- Airsim features a *settings.json* file that contains various simulator related variables
- Among the things we can configure:
  - Setting a wind force.
  - Camera FOV, resolution, stabilization, etc.
- An example settings file: https://microsoft.github.io/AirSim/settings/

Artificial Intelligence & Information Analysis Lab

Project 1

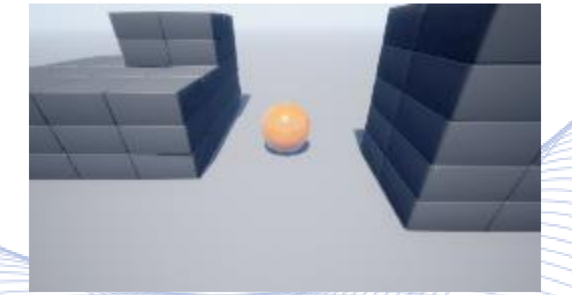
- Drive the drone inside the yellow ball.



Artificial Intelligence & Information Analysis Lab

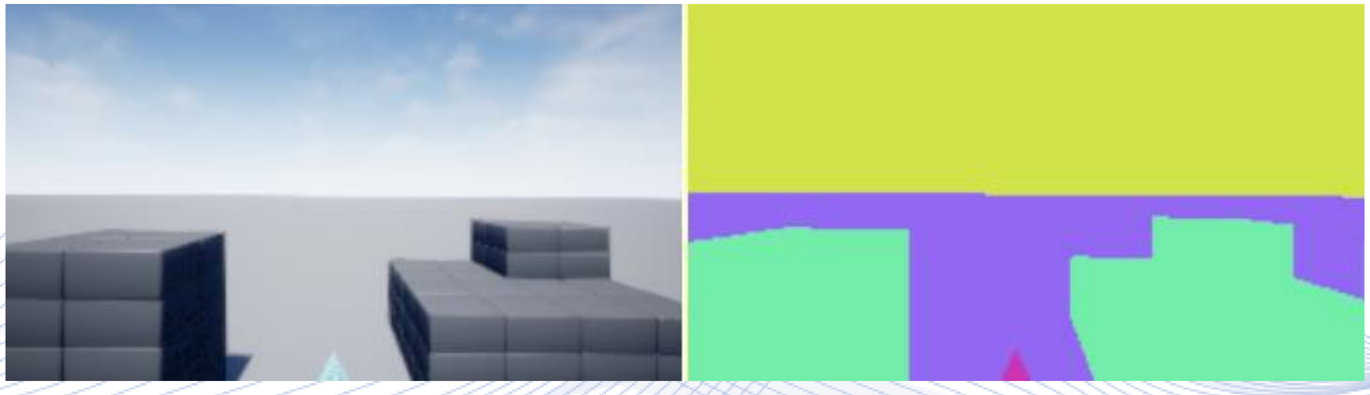
Project 2

- Take a picture of the yellow ball from the drone camera.



Project 3

- Create a semantic segmentation dataset and train a basic CNN semantic segmentation network on it.



Q & A

Thank you very much for your attention!

Contact: Prof. I. Pitas
pitas@csd.auth.gr