Flying Drone Toolkit for Unity
Virtual Reality Extensions

Clockworks Games

Version 2.0 – June 2016

- Enabled Flying Drone Toolkit for VR compatibility, in particular with the Oculus Rift CV1.

Introduction

The original Flying Drone Toolkit for Unity (version 1) is improved upon for virtual reality (VR) support (version 2).

Initial support is for Oculus Rift CV1. Future releases of the Flying Drone Toolkit may support additional VR platforms, if there is interest from the community.

The Toolkit makes it easy to add flying drones to Virtual Reality experiences. In addition, an included Unity package file makes it very easy to turn any Unity 3D environment into a VR drone experience. You can fly a drone in third-person perspective through any environment purchased from the Unity Asset Store after just a few minutes of setup. Of course, the same is true if you create your own 3D Unity environments. This image shows a snapshot of a virtual reality drone adventure through the Dark City[1] asset from Manufactura K4.

HOSTEL
cloud
data protection
best security systems
N.U.P.C
WHITE LABEL

[1] https://www.assetstore.unity3d.com/en/#!/search/page=1/sortby=relevance/query=Dark&City

This manual gives brief instructions for trying the features of the VR-enabled Flying Drone Toolkit for Unity.

Before starting, you should have appropriate Unity and Oculus software installed and configured. The version used and tested are Unity 5.3.4p1, the Oculus OVRPlugin for Unity,[2] and Oculus SDK & Runtime 1.4.0.

Try the Sample Scene

After loading the Flying Drone Toolkit (version 2.0 or later), open the DroneCityVR sample scene.

Also, make sure that the following settings are configured for VR and joystick control:

- Edit -> Project Settings -> Player -> Virtual Reality Supported – should be checked.
- Edit -> Project Settings -> Input -> several Axes should have Type: Joystick Axis,[3] including DroneRotate, DroneForwardBack, DroneUpDown, DroneRightLeft, and DroneForce.

Running the sample scene should project the Drone City into the Rift headset.

Joystick Control

Several configurations of joystick control mappings were experimented with. The following was selected, but are easy to change in InputManager if you have different preferences.

- Left Stick Y Axis – forward / backward.
- Right Stick X Axis – rotate
- D-Pad Y Axis – up / down
- D-Pad X Axis – strafe left / right
- Left Trigger – faster speed for all movement

Putting the forward / backward and rotate controls on the same stick seems intuitive, but was also more difficult to control accurately. This led to the decision to put one of these functions on each of the two sticks.

Motion Sickness Warning

Please note that as a Toolkit, this package allows you infinite flexibility. You should consider the VR guidelines[4] to avoid or minimize motion sickness for your users. For example, rapid yaw rotation may induce some simulator sickness in many users.

Switching Between VR and Non-VR Mode.

Two sample scenes are included: DroneCity and DroneCityVR, which have slightly different configurations for non-VR and VR. Project Settings also need to be adjusted, because these are not saved with the individual scenes.

[2] https://developer.oculus.com/downloads/game-engines/1.3.0/OVRPlugin_for_Unity_5/

[3] These axes can alternately be set to Key or Mouse Button, although the keyboard is more difficult to use than a joystick in VR.

[4] https://developer.oculus.com/documentation/intro-vr/latest/concepts/bp_app_simulator_sickness/

- Edit -> Project Settings -> Player -> Virtual Reality Supported – Check or not, depending on whether using VR.
- Edit -> Project Settings -> Input -> several Axes (including DroneRotate, DroneForwardBack, DroneUpDown, DroneRightLeft, and DroneForce) – These should have Type: Joystick Axis when using a game controller rather than the keyboard, which can be used for VR or non-VR mode. Additionally, Invert should be checked for DroneForwardBack and DroneUpDown when using a game controller.

Try a Different 3D Environment, using the Included Dome Drone

Open Unity with a 3D environment of your own choosing, perhaps purchased from the Unity Asset Store, or of your own creation.

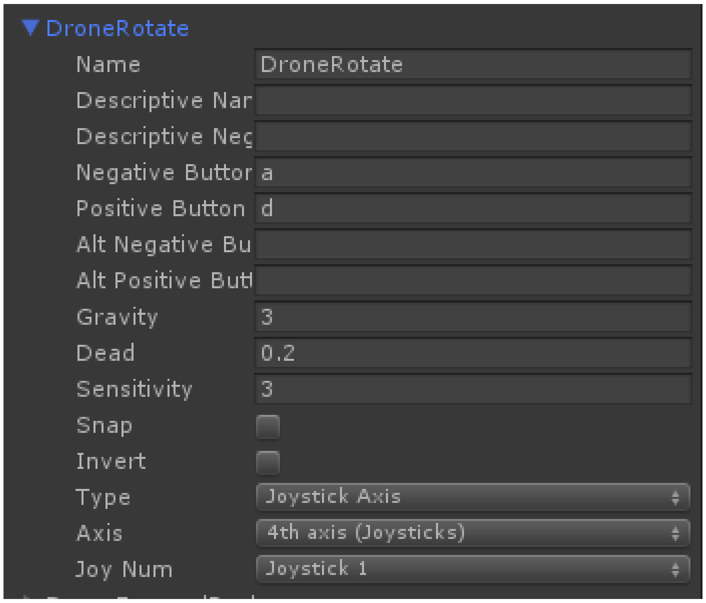
Follow the following 5 steps to turn this environment into the VR drone experience:

Step 1: Configure your project to support VR.

- Edit -> Project Settings -> Player -> Virtual Reality Supported – should be checked.

Step 2: Configure the control axes.

- Edit -> Project Settings -> Input -> several Axes need to be set, including DroneRotate, DroneForwardBack, DroneUpDown, DroneRightLeft, and DroneForce using the following values:



▼ DroneRotate

Name	DroneRotate
Descriptive Nar	
Descriptive Neg	
Negative Buttor	a
Positive Button	d
Alt Negative Bu	
Alt Positive Butt	
Gravity	3
Dead	0.2
Sensitivity	3
Snap	□
Invert	□
Type	Joystick Axis
Axis	4th axis (Joysticks)
Joy Num	Joystick 1

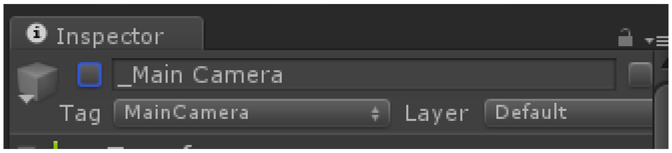
- Use these Axis values for each to use the joystick mappings described above:
    - DroneRotate: 4th axis
    - DroneForwardBack: Y axis
    - DroneUpDown: 8th axis
    - DroneRightLeft: 7th axis
    - DroneForce: 3rd axis

An easy way to set the 5 axes is to copy the InputManager.asset file from the ProjectSettings folder of the Flying Drone Toolkit (be sure to leave the original file in place), and put this copy in the ProjectSettings folder of your new project. CAUTION: Don't take this short cut if you have other non-standard input settings in your new project.

Step 3: Import the FlyingDroneVR Unity package file from the Packages in the Flying Drone Toolkit.

- Double click on the FlyingDroneVR file.
- Click "Import" on the resulting dialog.

Step 4: Disable any other cameras in your scene. (The prefab we will use below has its own camera.)



□ _Main Camera

Tag MainCamera Layer Default

Step 5: Add a VR drone to your scene.

- Drag the fdDomeDroneVR prefab from your Project window (in FlyingDroneToolkit -> Prefabs) to your scene.
- Use the Unity editor to position the drone so that it is above ground level and does not intersect with any other game objects.

At this point, if you run your scene, it should be rendered in your rift. *Explore your own environment in VR!*

Use a Third-Party Drone Model

You are not limited to using the simple 3D drone models included with the Flying Drone Toolkit. The Unity Asset Store has alternative, more detailed drone models for sale that can be used with the Toolkit, leveraging the same movement scripts and other features of the Flying Drone Toolkit.[5]

For example, the Drone Pack,[6] sold in the Unity Asset Store by ConeFin, provides several alternate drone models, such as this one:

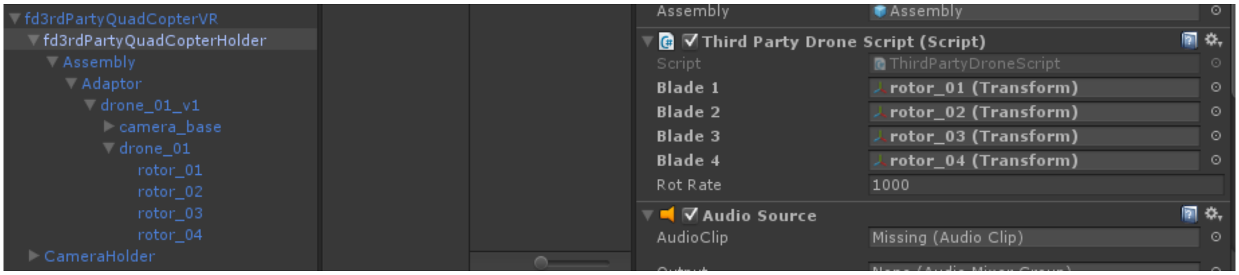
To use this Drone Pack, first follow the instructions above, since most of the configuration is the same. You need to perform steps 1-4 above. These instructions replace step 5. If you have already performed step 5, simply disable the drone prefab in the Inspector.

Then:

[5] These other drone models are sold by other companies. Clockworks Games, which makes the Flying Drone Toolkit, has no connection to these other companies, but strives to be compatible with them, so that different drone models can be used.

[6] https://www.assetstore.unity3d.com/en/#!/content/42489

- As an example, purchase and download the Drone Pack made by ConeFin from the Unity Asset Store. (Other drone models can be used, with some variation of the procedures listed here.)
- Add to the scene the fd3rdPartyQuadCopterVR prefab from the FlyingDroneToolkit Prefabs folder.
- Expand the new prefab to display the levels:
  - Fd3rdPartyQuadCopterVR
    - Fd3rdPartyQuadCopterHolder
      - Assembly
        - Adaptor
- From the Drone Pack, drag one of the prefabs under the Adaptor level listed above.
- In the fd3rdPartyQuadCopterHolder, in the Inspector, scroll to the Third Party Drone Script. Note there are four slots for rotating blades. Expand the drone model, and note that there are four corresponding rotors. Drag the rotor objects into the corresponding slots in the Third Party Drone Script.



Assembly: Assembly

☑ Third Party Drone Script (Script)

Script	ThirdPartyDroneScript
Blade 1	rotor_01 (Transform)
Blade 2	rotor_02 (Transform)
Blade 3	rotor_03 (Transform)
Blade 4	rotor_04 (Transform)
Rot Rate	1000

☑ Audio Source

AudioClip	Missing (Audio Clip)

- Optional: Drag some appropriate sound clip to the to the AudioClip slot in the fd3rdPartyQuadCopterHolder.
- In the Adaptor layer, make the following changes, which were determined through experimentation with this specific drone model:
  - Change Rotation Y to 90. This aligns the drone in the appropriate direction.
  - Change Scale X, Y, and Z to 0.025. Otherwise the drone is much too large for the environment.

You should now be able to run, now using the third-party drone model that you selected.

Parameters to Adjust

The following parameters are especially relevant for fine tuning the experience of the Flying Drone Toolkit in VR.

Flying Drone Script:

- User Mode: Should always be Manual when directly controlling the drone.
- Force Multiplier: Overall measure of how much force is applied to the drone to make it move.
- Up Down Factor: Fraction of up / down force compared with forward / back force.
- Left Right Factor: Fraction of side-to-side force compared with forward / back force.
- Rotate Factor: Measure of how much torque is applied for drone yaw rotation.

- Force Factor: The degree to which the acceleration function[7] increases the force.
- Motion Mag Max: Magnitude of the random "drifting" up-and-down motion.
- Motion Time Increment: Measure of the speed of the random "drifting."

Drone Camera Script:

- Height: Distance of following camera above the drone.
- Distance: Overall distance of following camera from the drone.
- Damping: The "elasticity" between the following camera and the drone. Higher numbers approach a very rigid constant distance between the camera and drone. Lower numbers result in a very "loose" connection between the two. From experimentation, 10 seems like a reasonable value.

[7] Set by default to the joystick left trigger.