Simple Drone Physics

AIKODEX

Documentation

Controls : WASD/Arrow Keys for movement. Left Shift for lift, Left Control/Command for descending, C for camera change, L for Toggling Lights, B for Blinker/Indicator change and R for Reset.

Simple Drone Physics is a package which will allow you to create your own drone game in no time. This package has been built by carefully observing drone behavior. Taking the basic principle of a flying drone which is resultants of vector math, we can easily add various forces and directions and the **Built-in 3D physics** (**Nvidia PhysX engine**) can handle the resultants for us.
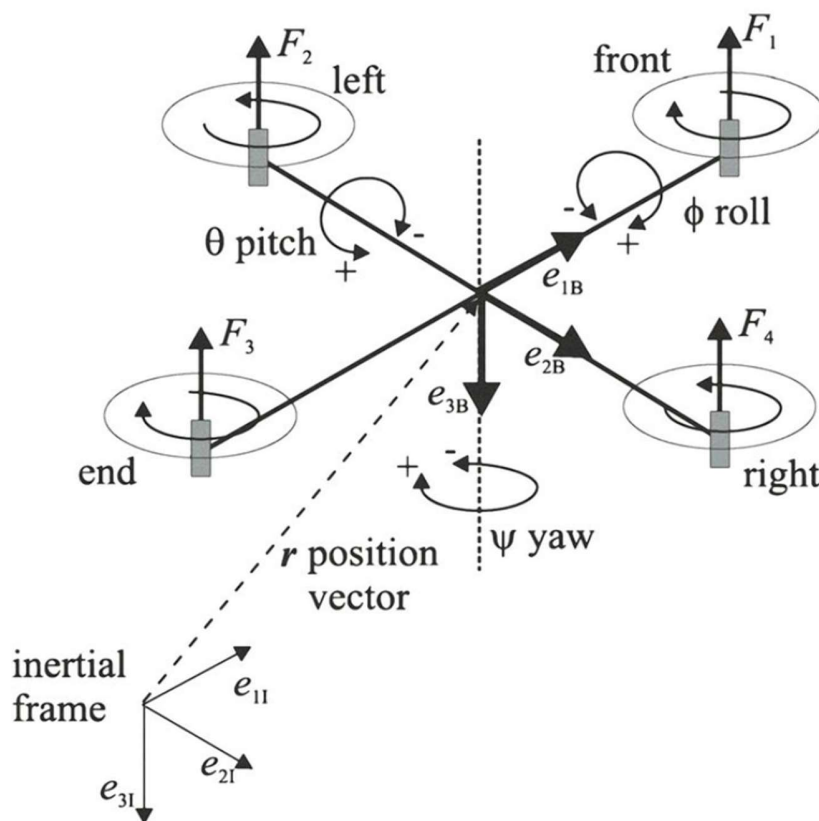
In the package, the GameObjects FL, FR, RL, RR correspond to the F_1, F_2, F_3, F_4 in the figure above respectively. By changing the fan rpm values of FL, FR, and RL, RR the pitch value is obtained. Similarly, by changing the fan rpm values of FL, RL and FR, RR the roll value is obtained. For yaw however, changing the down fan rpm of FL, FR, RL, RR does not cause the desired yaw affect. In a quadcopter, the fans rotating clockwise or anti clockwise in a specific manner causes the quadcopter to rotate on the axis perpendicular to the ground. Hence the approach to rotate the quadcopter on this axis is adding a force parallel to the ground and perpendicular to the direction of the quadcopter

In this package, the main script for controlling drifting is Drone Controller (C#) which handles the physics of a flying drone.

Asset Overview and Description

In the Simple Drone Physics Folder, you will find :
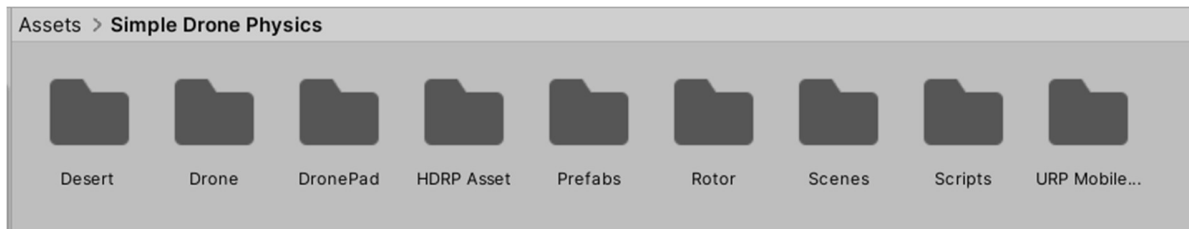
Desert : This folder contains the materials, models, and textures for the desert prefab in the hierarchy. In the materials we have sand which has two normal maps and one albedo for the main texture applied to the Model named Dunes.

Drone : This folder contains the materials, models, and textures for the drone prefab in the hierarchy. In the materials we have body on which we have two albedo maps for the main texture. One is the main texture; the other texture is the dirt texture for the secondary albedo if you want to apply dirt to it.

DronePad : This folder contains the materials, models, and textures for the DronePad prefab in the hierarchy.

HDRP Asset : This folder contains the Simple Drone Physics HDRP unitypackage which is crucial if you want to upgrade to HDRP. Simply open a new Project > Package Manager > install HDRP. Then import the Simple Drone Physics HDRP unitypackage.

Prefabs : Contains drag and drop Drone, DronePad, Dunes ready to be used. All prefabs are set to position (0,0,0), rotation (0,0,0) and scale (1,1,1)

Rotor : This folder contains the materials, models, and textures for the display rotors used in the drone prefab in the hierarchy. All the blurred textures are used in the drone controller.cs script for blurring the rotors.
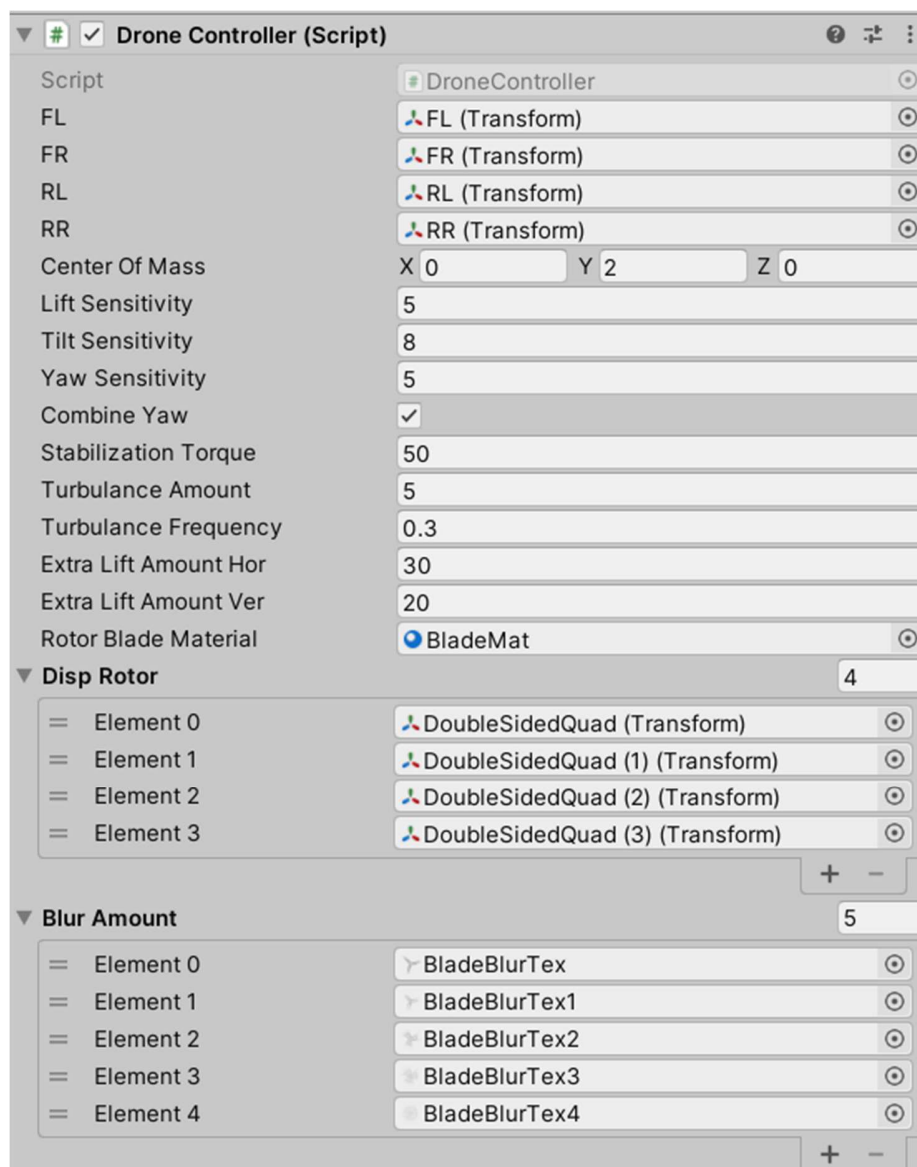
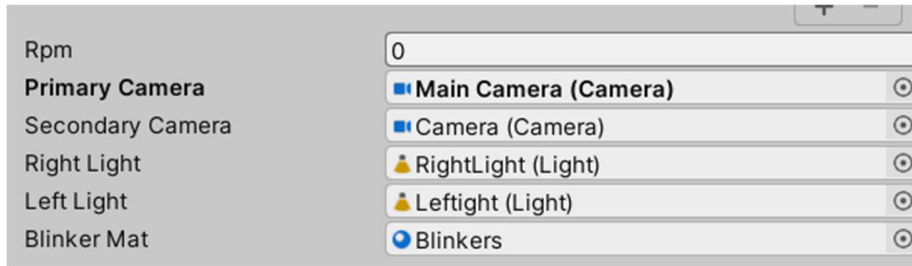**Scenes :** Contains the Example Scene with hierarchy

URP Mobile Asset : To migrate to mobile check page 6.

Scripts : Scripts that are used for controlling the drone behavior, Cameras, and the generation of dune tiles.

Drone Controller (C#)

Note : The script is attached to DroneRB inside Drone in the hierarchy. Please drag and drop DroneRB into the inspector for referencing and not Drone directly.

▼ # ✓ Drone Controller (Script)	❓ ⇄ ⋮
Script	# DroneController ⊙
FL	⚵ FL (Transform) ⊙
FR	⚵ FR (Transform) ⊙
RL	⚵ RL (Transform) ⊙
RR	⚵ RR (Transform) ⊙
Center Of Mass	X 0 Y 2 Z 0
Lift Sensitivity	5
Tilt Sensitivity	8
Yaw Sensitivity	5
Combine Yaw	✓
Stabilization Torque	50
Turbulance Amount	5
Turbulance Frequency	0.3
Extra Lift Amount Hor	30
Extra Lift Amount Ver	20
Rotor Blade Material	🔵 BladeMat ⊙
▼ Disp Rotor	4
☰ Element 0	⚵ DoubleSidedQuad (Transform) ⊙
☰ Element 1	⚵ DoubleSidedQuad (1) (Transform) ⊙
☰ Element 2	⚵ DoubleSidedQuad (2) (Transform) ⊙
☰ Element 3	⚵ DoubleSidedQuad (3) (Transform) ⊙
	+ −
▼ Blur Amount	5
☰ Element 0	↦ BladeBlurTex ⊙
☰ Element 1	↦ BladeBlurTex1 ⊙
☰ Element 2	↦ BladeBlurTex2 ⊙
☰ Element 3	↦ BladeBlurTex3 ⊙
☰ Element 4	↦ BladeBlurTex4 ⊙
	+ −

Rpm	0
Primary Camera	◉ Main Camera (Camera)
Secondary Camera	◉ Camera (Camera)
Right Light	RightLight (Light)
Left Light	Leftight (Light)
Blinker Mat	Blinkers

- FL, FR, RL, RR (Transforms) : These are points at which the forces are applied. The center of mass is balanced against these forces.
- Center Of Mass : Center of Mass of the RigidBody set via RB.centerOfMass
- Lift, Tilt and Yaw Sensitivity (Or in drone terms. Roll, Pitch and Yaw) : Higher Values correspond to higher lift, tilt, and Yaw.
- Combine Yaw : Combines the Yaw values and Tilt (Pitch) value to make the game more playable.
- Stabilize Torque : Does not let the Drone tip over. Zero Torque value still does not ensure tiltCheck() to be switched off. Also adds a forward/Backwards force in order to return to equilibrium realistically.
- Turbulence Amount & Frequency : Amount here is the amplitude and frequency is the density of packed turbulent air. Can also interchangeably used for power fluctuation in the rotors.
- ExtraLiftAmount Hor/Ver : The drone dips when stabilization torque is too high. This parameter adds an extra horizontal downforce to counteract the drone dipping too much.
- Rotor Blade Material : Rotor/Materials/BladeMat
- Disp Rotor [] : Array of quads that are supposed to represent the fans in the drone. These quads have an RPM value which is not connected to downforce provided by the fans and are just for display purpose.
- Blur Amount [] : An array that stores textures. Based on arbitrary values of RPM the textures change the blurriness amount.
- Rpm : For displaying the blurred rotor texture. Does not have a physical significance
- Primary/ Secondary Camera : Accepts a gameobject with Camera script attached. Primary camera is set to default third person view of the drone. Secondary Camera is connected to the Cam gameobject inside the Drone Gameobject.

Helicopter Camera (C#)
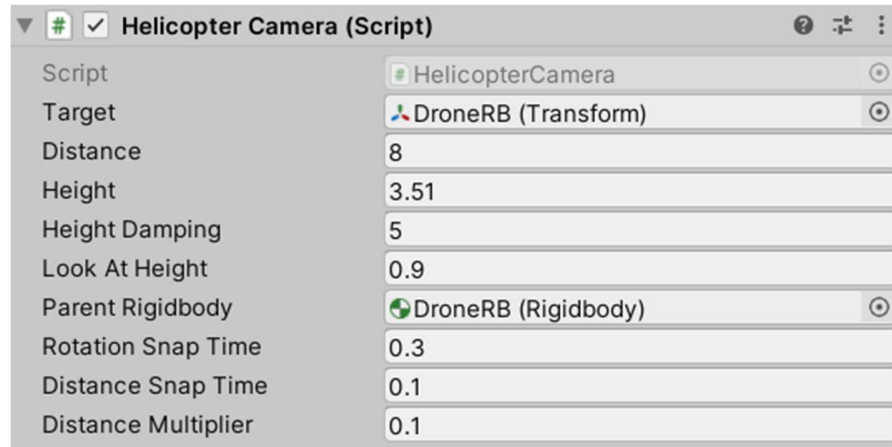
- **TARGET** – GameObject to follow around.
- **DISTANCE** – Horizontal distance from the camera
- **HEIGHT** – Vertical Distance from the camera
- **HEIGHT DAMPING** – Time taken for the camera to reach that height
- **LOOKATHEIGHT** – Lookat offset for Height (0,Y,0)
- **PARENT RIGIDBODY** – Distance multiplier Calculations
- **ROTATION/DISTANCE SNAP** – Time taken to reach specified rotation and distance values
- **DISTANCE MULTIPLIER** – Move away effect of the camera as the object speeds up

Perfect Mouse Look (C#)

When the mouse moves the perfect mouse look script is activated and the user can look around with the mouse around the target object. When the movement is stopped, the rotation snaps back to the original position smoothly.

Pseudo Procedural Dunes (C#)

Instantiates tiles around the player and updates them where the player goes on the (x,y,0) plane. This is so that the tiling looks endless.
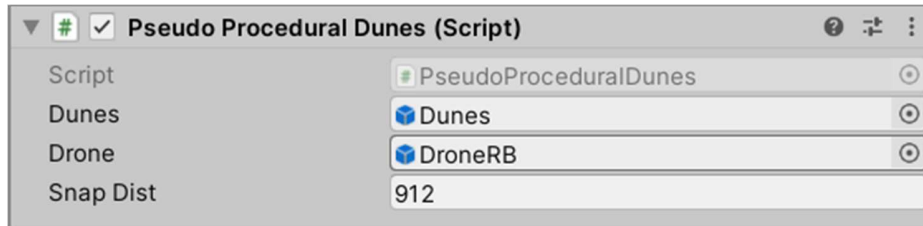
- **Dunes** : References the Dunes GameObject
- **Drone** : References the Object which is supposed to be moving in the map.
- **Snap Dist** : The distance for the tiles to be placed. Too less a distance, the tiles will overlap, too great a distance, there will be gaps in the tiling.

MIGRATING TO MOBILE

1. Open a new unity project
2. Go to Package Manager > install URP
3. Import the asset given in Simple Drone Physics/ URP Mobile Asset
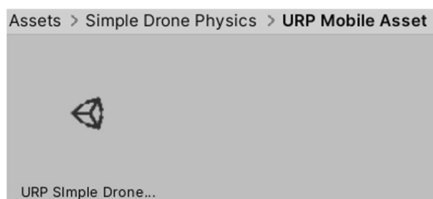
4. Switch Platform to Android/iOS in the build settings
5. Connect phone and click on Build and Run

TROUBLESHOOTING (FAQs)

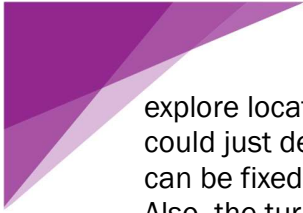Q. I want my drone to go faster, how can I make that happen?

Ans : There are two ways you can make the drone go faster, first by increasing the stabilization Torque (with the stabilization Torque you may also want to increase the Extra Lift Amount Ver) or you may very slightly decrease the center of mass Y to about 1.9.

Q. The game looks too plain. I want the same realistic look as in the videos. Is it possible for me to get it?

Ans: Yes! The videos are recorded in the HDRP project which has out of the box configurable fog, sandy winds, and color corrections.

Q. The Drone simply looks too big. I want a smaller drone for my game. How can I achieve that?

Ans: The drone modelled in this game is supposed to be a big >10kg professional camera equipment holder drone usually used to get in unreachable places like hot or icy deserts and

explore locations in high definition. However, if you wanted a smaller drone, fret not, you could just decrease the scale of the drone. You'll notice that the controls seem a bit off. That can be fixed if you adjust the center of mass (bring it down) and the mass of the Rigidbody. Also, the turbulence might affect the drone too much at a smaller scale so be sure to reduce that too.

Q. In the asset photos I saw different textures on to the drone. How can I apply dirt and a custom skin on my drone?

Ans: The drone ships with a few custom procedural skins. You can apply them to the body material albedo texture. If you go down below in the material, you will find secondary map or detail map depending on the RP (built in or HDRP). You can place the white Dirt skin or the black dirt skin which ever you prefer to get dirt in the drone.

Q: When I drag and drop the Drone prefab from the hierarchy onto the main camera nothing happens or there's a weird behavior. Same with procedural dunes. How can I fix this?

Ans: You need to drag and drop **DroneRB (child of Drone) and not Drone** on to the main camera or procedural dunes script. That should fix it.

Q How can I make the Drone fly instantly when I press L shift/ On mobile?

Ans: In the fixed update, simply remove the motorswitch code or set that to true.

If you have any questions or suggestions please mail them to alternativemathsthinking@gmail.com and we will surely be happy to attend to them.

Happy flying!

Offered by AiKodex