Ultimate Drone Physics

“Yue Ultimate FPV Drone Physics” is an asset to create a FPV Drone Controller.

For a video **tutorial and additional information** check out https://www.youtube.com/c/YueBeifong

Setup

To create a FPV Controller in Unity with “Yue Ultimate FPV Drone Physics” the following steps are required.

- Use the input-setting in the “Settings” Folder. (It changes “Mouse X” and “Mouse Y”)
- Add the “YueDronePhysics” and “YueInputModule” component to your Drone object
- Create a script that populate the “YueInputModule” with inputs from your controller.

Note: The controller inputs should be between –1 to 1 and be assigned to the raw values, e.g. “RawLeftHorizonal”. The values should match their names for the drone to respond correctly.

- Select your preferred “Flight Mode” in “YueInputModule” and your “Flight Configuration” in the “YueDronePhysics”.
- In the final step you need to tune your "Rates Config” in the input-module and the “Physics Config” in the “YueDronePhysics” component.

Tipp: Copy the setting from the prefabs to have a good starting point.

Now the Quadcopter should be setup properly.

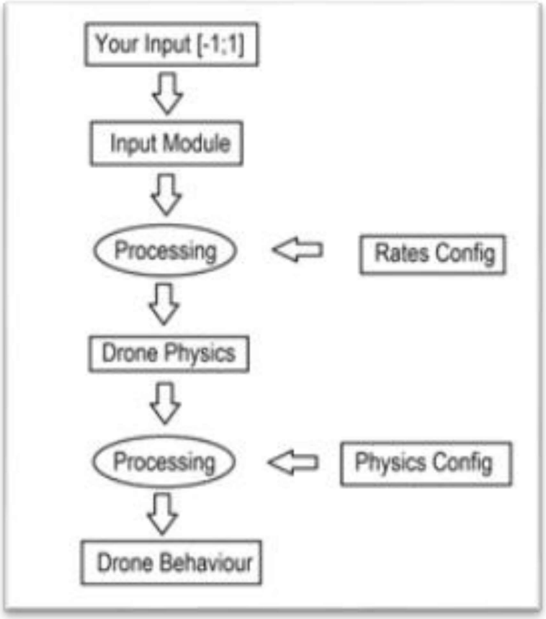
YueInputModule

- “Gains [Deg /s]” describes the amount of change in orientation of the target angle over time in deg. This only affects the Acro-Mode. (And the self-leveling mode on the yaw axis).
- “Flight Modes" flight modes define what joystick input affects what axis on the aircraft. The general default is “Mode 2”.
- “Max Angle [Deg]” refers to the maximum angle and the pitch and roll axis in self-leveling and altitude-hold Mode.

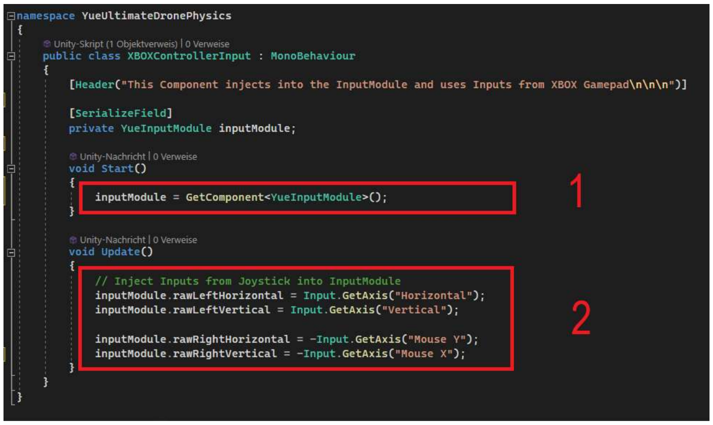
YueDronePhysics

- “Flight Config” defines the flight configuration such as “Acro-Mode”, “Self-Leveling” and “Altitude-Hold”.
- “Armed” is a boolean to turn the drone on and off.
- “Max Thrust” when raw thrust input is “1”.
- “Physics” properties to will be applied to the rigid body.
- “PID” describes the strength of the forces and torque that keeps the drone in its target orientation and altitude. “P” (proportional) could be described as a spring force and “D” (derivative) as the damper.

Data Processing Structure



Populate Input Module Example



```
namespace YueUltimateDronePhysics
{
    Unity-Skript (1 Objektverweis) | 0 Verweise
    public class XBOXControllerInput : MonoBehaviour
    {
        [Header("This Component injects into the InputModule and uses Inputs from XBOX Gamepad\n\n\n")]

        [SerializeField]
        private YueInputModule inputModule;

        Unity-Nachricht | 0 Verweise
        void Start()
        {
            inputModule = GetComponent<YueInputModule>();
        }

        Unity-Nachricht | 0 Verweise
        void Update()
        {
            // Inject Inputs from Joystick into InputModule
            inputModule.rawLeftHorizontal = Input.GetAxis("Horizontal");
            inputModule.rawLeftVertical = Input.GetAxis("Vertical");

            inputModule.rawRightHorizontal = -Input.GetAxis("Mouse Y");
            inputModule.rawRightVertical = -Input.GetAxis("Mouse X");
        }
    }
}
```

1. Get component “YueInputModule”
2. Populate raw inputs in the module with your input