Airplane Controller

By: Jess Case

The purpose of this document is to provide information on how the “Airplane Controller” works along with how to use and modify its components. The “Airplane Controller” is an open source flight controller for the Unity Engine. The included “AirplaneController”, “Checkpoint”, and “CheckpointCounter” scripts are free to use in both commercial and non-commercial applications. Giving attribution is not required, but is greatly appreciated.

NOTICE:
The demo scene includes artwork from the “POLYGON - Starter Pack” asset, and thus is not to be used in commercial applications.

If you have a problem that you cannot solve or have added a feature that you would like to see added to this package email jesscasedev@gmail.com.

Included In This Package

AirplaneController- Folder, contains package

- AirplaneController - Script, contains main code for the package (Put on object to be used as prefab)
- Checkpoint - Script, contains code used to check for player flying through checkpoints
- CheckpointCounter - Script, optional script that holds a list of all rings in a scene and updates a counter that can be placed on the screen. Shows how many rings the player has flown through.
- Prefabs - Folder, contains an airplane and ring prefab. **These assets are created with Polygon assets and are not free to use.**
- DemoScene - Folder, contains everything used in the demo scene
    - PolygonStarter - Folder, package downloaded from Unity Asset Store and used in creation of the demo scene (Contains only assets used in Demo Scene, not whole package)
    - Demo - Scene

Creation

To place the controller in your Unity scene, simply drag the provided plane prefab into your scene or hierarchy. The prefab must have all the given components to work:

- A collider
- Rigidbody
- Airplane Controller Script

If you want to use the Airplane Controller Script with any other plane asset, you can attach the script to another object just make sure it has these components,

Default movement keys are as follows:

- W and S: controls acceleration
- A and D: controls roll
- Q and E: controls yaw
- Mouse movement up and down: controls pitch

These values can be changed in Project Settings -> Input Manager by changing to their corresponding values

- W and S: Vertical
- A and D: Roll
- Q and E: Yaw
- Mouse movement up and down: Mouse Y

Editor

The “Airplane Controller” contains a custom editor that holds every available variable for users to change. These variables are sectioned into the categories “Movement Setup”, “Propeller Setup”, and “Camera Setup”, and some sections may contain subsections. All features, such as maxSpeed and even the amount of propeller, can be directly changed in the editor.

List of Properties

Movement Setup

Property Name	Type	Function
Current Speed	float	Displays the current flight speed. Read only.
Max Speed	float	Determines the fastest

		possible flight speed when moving forward.
Max Reverse Speed	float	Determines the fastest possible speed when moving backwards.
Speed Ramp Up	float	Determines how fast the airplane gains speed while accelerating.
Speed Ramp Down	float	Determines how fast the airplane loses speed when acceleration stops.
Idle Speed	float	Determines the minimum speed of the airplane when acceleration stops in midair.
Rotations		
Invert Pitch	bool	When true,moving down with the mouse causes the plane to go up and moving up causes the plane to go down. When false, the opposite is true.
Pitch Rotation Speed	float	Determines the speed of the plane's up and down movement.
Yaw Rotation Speed	float	Determines the speed of the plane's left and right movement.
Roll Rotation Speed	float	Determines the speed of the plane's rotation along the Z axis.
Miscellaneous		
Direction Modifier	float	Adjusts forward direction of movement. Allows for a more controlled takeoff

		time. Shown as a black line.

Propeller Setup

Property Name	Type	Function
Propeller Amount	Int	Number of props on the plane.
Propeller	Transform Object	A child object of the plane, the propeller's rotation is controlled by code when "Propeller Amount" is greater than 0.
Propeller 1	Transform Object	Same as "Propeller".
Propeller 2	Transform Object	Same as "Propeller".
Speed Cap	float	Determines the max rotation speed of the propellers.

Camera Setup

Property Name	Type	Function
Camera Arm	Transform Object	Placed in the middle of the plane object. Directly controls the movement of the player camera.
Camera Pullback	bool	When active, the camera fall behind the plane during acceleration and catch uyp when not accelerating.
Min Camera Distance	float	The closest the camera will be to the plane. Camera starts at this distance and sits here when the player is not moving.
Max Camera Distance	float	The farthest the camera will

		be from the plane. While accelerating, the camera moves toward this distance.
Lag Rotation	bool	When active, the camera will rotate slower than the plane. This causes a “lag” effect.
Lag Speed	float	Determines the speed at which the camera lags and catches up when active.
Max Lag Distance	float	Determines the maximum rotation distance the camera will lag behind the plane when rotating.

Scripts

The “Airplane Controller” Script contains a number of private variables used for internal representations. These should not be changed unless changing systems in the script directly. It also contains private methods used in conjunction with Update() and FixedUpdated() for organization proposes.

The “Checkpoint” and “CheckpointCounter” are free and easy to use as well. Check the demo scene for an example of how to use these.