Aircraft Flying System Documentation

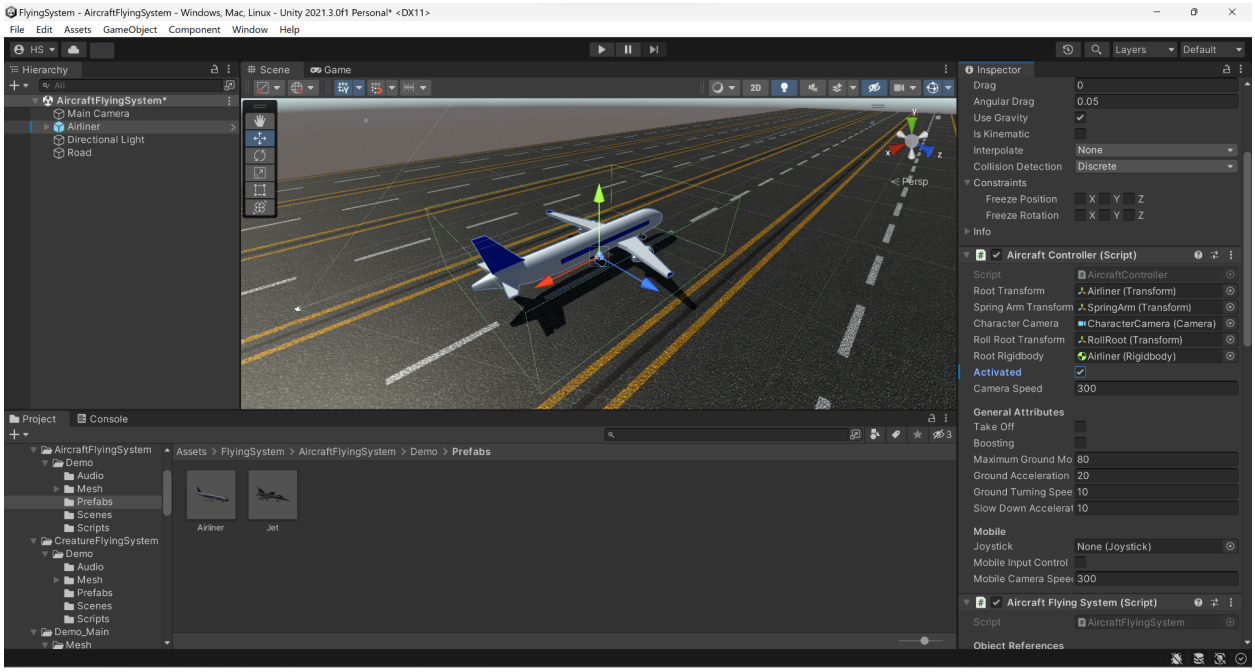
Description

This system is used for simulating flying behaviors of aircrafts like airplane and jet.

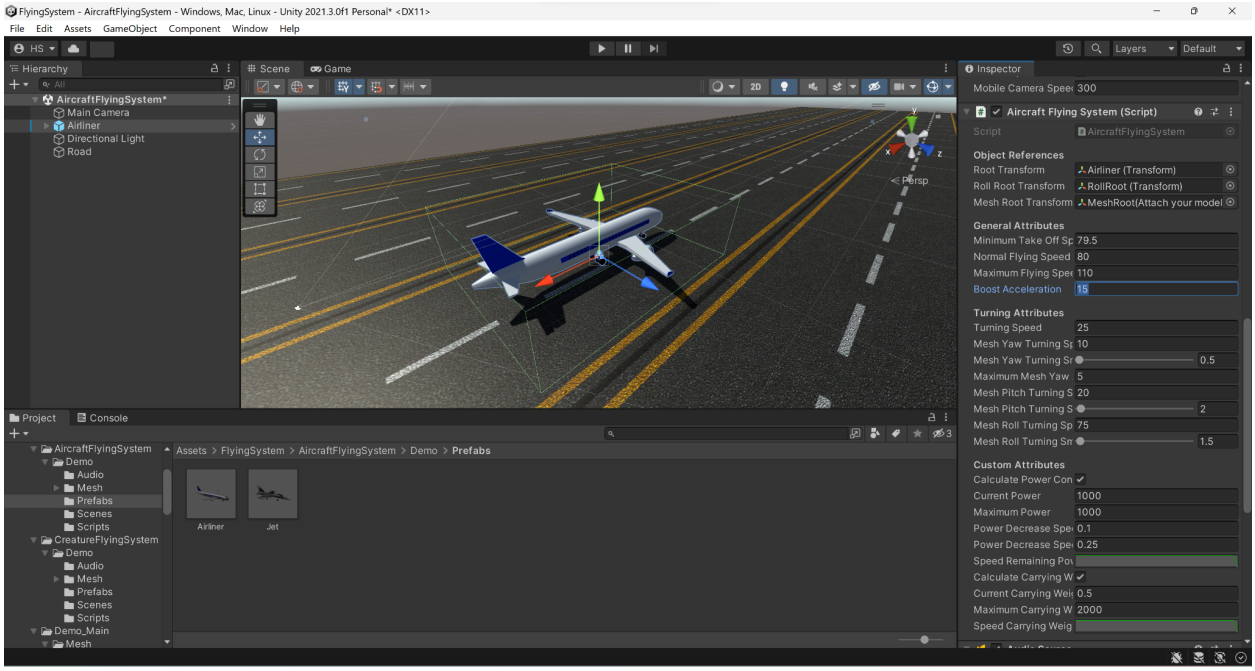
How to take off: This type of aircrafts needs to accelerate to take off. First accelerate on a runway, when it reaches the maximum speed on the ground, press space bar then press Q to point up.

Setup

1. Import the plugin.
2. Install “Post Processing” using Package Manager if it is not installed.
3. Create a new scene and make a simple terrain, navigate to Assets\FlyingSystem\AircraftFlyingSystem\Demo\Prefabs folder, drag the Airliner prefab into the scene, select “Activated” for the AirlinerController script on the Inspector panel.



4. Press the Play button, the flyer would be controllable. Depending on the need, the attributes and methods in Aircraft Flying System component can be adjusted/executed.



Explanation

Setting the pitch, yaw, roll together for the same GameObject will cause gimbal lock problem(https://en.wikipedia.org/wiki/Gimbal_lock). To solve this problem, two GameObjects(RollRoot, MeshRoot) are used to separate them.

The prefab has the following hierarchy:

- Airliner
    - SpringArm
        - CharacterCamera
    - RollRoot
        - MeshRoot
            - Airliner(3d model)

1. Airliner: It only acts as the root, and doesn't represent the actual collision for the flyer. Rigidbody, C# scripts and audio source are attached to this GameObject.
2. SpringArm: For controlling the third-person camera.
3. CharacterCamera: The actual camera.
4. RollRoot: The roll value of the flyer will be set here.

5. MeshRoot: The pitch and yaw values will be set here.
6. Airliner(3d model): The actual flyer mesh, can be customized. The mesh should be attached to MeshRoot, since the flying rotation will be applied to its parent, not directly to the mesh.

Essential Object References

rootTransform	The Transform component of the root object.
rollRootTransform	The Transform component of the roll root object. There must be a RollRoot object above the MeshRoot object in the Hierarchy. See the examples of “Airliner.prefab” and “Jet.prefab” in Assets\FlyingSystem\Demo\Prefabs\Controllers\AirTransportations folder.
meshRootTransform	The Transform component of the mesh root object.

Adjustable Attributes

General Attributes	
minimumTakeOffSpeed	The minimum speed to take off.
normalFlyingSpeed	The speed of normal flying.
maximumFlyingSpeed	The maximum speed of flying in boost mode.
boostAcceleration	It defines how fast the speed will increase from normalFlyingSpeed to maximumFlyingSpeed.
Turning Attributes	
turningSpeed	It defines how fast the aircraft turns.
meshYawTurningSpeed	The speed of yaw(horizontal) turning.

meshYawTurningSmoothingFactor	The interpolated parameter for meshYawTurningSpeed. The smaller the value, the smoother and slower the turning will be.
maximumMeshYawAngle	The maximum yaw angle of the mesh.
meshPitchTurningSpeed	The speed of pitch(vertical) turning.
meshPitchTurningSmoothingFactor	The interpolated parameter for meshPitchTurningSpeed. The smaller the value, the smoother and slower the turning will be.
meshRollTurningSpeed	The speed of roll turning.
meshRollTurningSmoothingFactor	The interpolated parameter for meshRollTurningSpeed. The smaller the value, the smoother and slower the turning will be.
Custom Attributes	
calculatePowerConsumption	Whether calculate the power consumption or not.
currentPower	The current power.
maximumPower	The maximum power.
powerDecreaseSpeed	It defines how fast the power drops when flying.
powerDecreaseSpeedWhenBoosting	It defines how fast the power drops when flying in boost mode.
speedRemainingPowerRatioAnimationCurve	An editable curve of the relation between speed and remaining power. Normally the lower power, the slower speed.
calculateCarryingWeight	Whether calculate the carrying weight or not.
currentCarryingWeight	The current carrying weight.
maximumCarryingWeight	The maximum carrying weight.
speedCarryingWeightRatioAnimationCurve	An editable curve of the relation between speed and carrying weight. Normally the higher carrying weight, the slower speed.

Other Public Attributes	
enabledFlyingLogic	Whether enable/disable the flying logic.
inAir	Whether it is in the air or not.
flyingDirection	The flying direction, in Vector3.
flyingSpeed	The flying speed.
flyingVelocity	The flying Velocity, in Vector3.
flyingAtNormalSpeed	Whether it is flying in normal speed mode or not.
boosting	Whether it is flying in boost mode or not.
powerPercentage	The percentage of power, equals 1 when power is full.
weightPercentage	The percentage of carrying weight, equals 1 when fully carried.

Methods

TakeOff(float groundMovementSpeed)	Make the aircraft take off, return false if groundMovementSpeed < minimumTakeOffSpeed.
Land()	Make the aircraft land.
AddYawInput(float value)	Make the aircraft turn left/right.
StopYawInput	Make the aircraft stop turning left/right.
AddPitchInput(float value)	Make the aircraft point up/down.
AddRollInput(float value)	Make the aircraft roll left/right.
AddWeight(float increaseValue)	Increase the current carrying weight of the aircraft, the increaseValue can be positive or negative.

Universal Render Pipeline (URP) & High Definition Render Pipeline (HDRP)

The materials can be converted by following this tutorial:
https://www.youtube.com/watch?v=aJ1OpirisGM

References

The 3D models(Airliner.fbx, Jet.fbx) are the original creation by the developer of this project.