

Helicopter Flying System Documentation

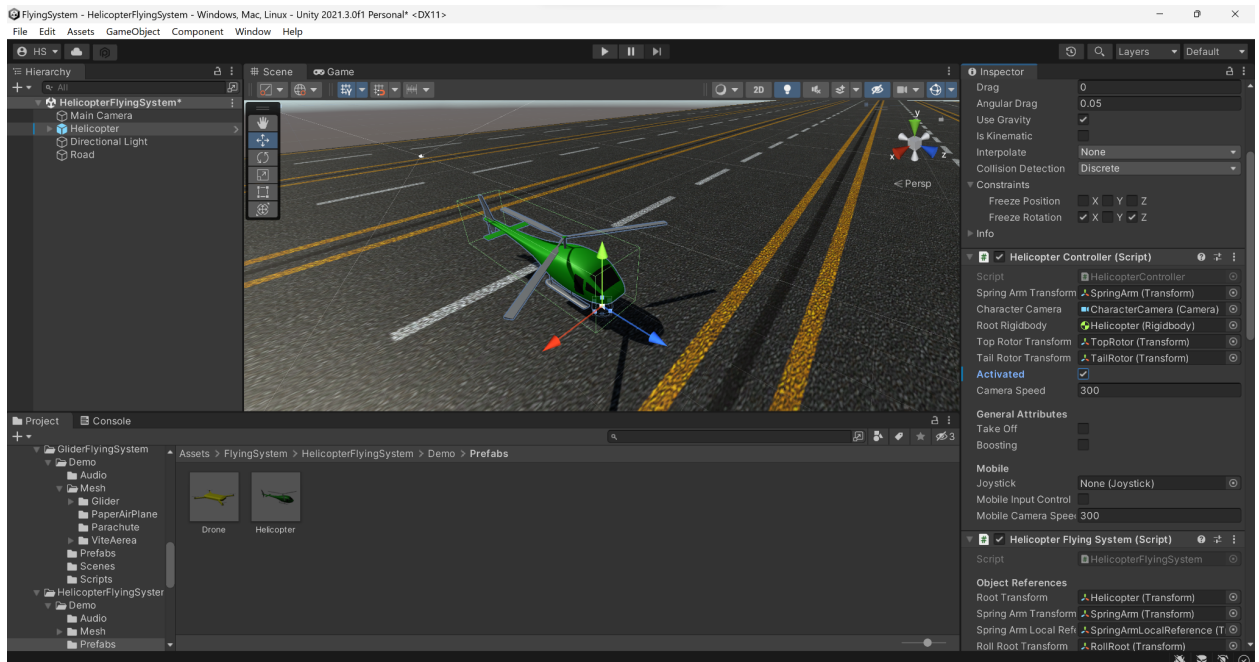
Description

This system is used for simulating flying behaviors of helicopter and drone.

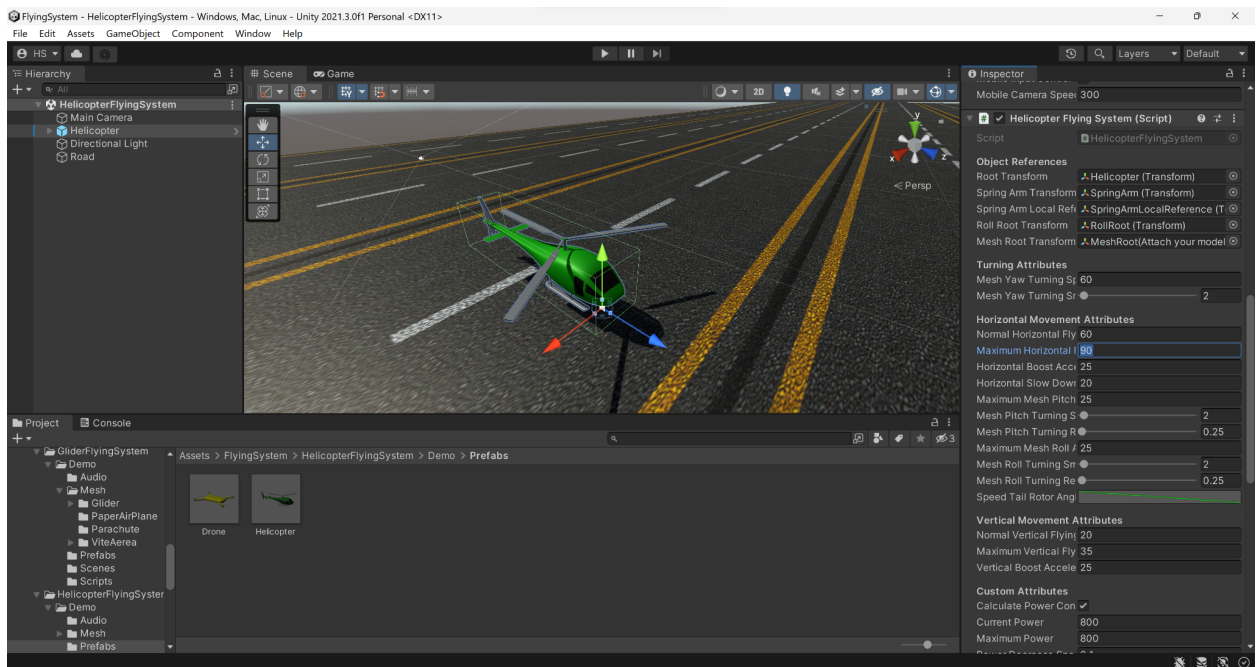
How to take off: Press W to Ascend.

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\HelicopterFlyingSystem\Demo\Prefabs](#) folder, drag the Helicopter prefab into the scene, select “Activated” for the HelicopterController script on the Inspector panel.



- Depending on the need, the attributes and methods in Helicopter 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:

- Helicopter
 - SpringArm
 - CharacterCamera
 - RollRoot
 - MeshRoot
 - Helicopter(3d model)
- Helicopter: 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.
 - SpringArm: For controlling the third-person camera.
 - CharacterCamera: The actual camera.
 - RollRoot: The roll value of the flyer will be set here.

5. MeshRoot: The pitch and yaw values will be set here.
6. Helicopter(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.
springArmTransform	The Transform component of the spring arm object.
springArmLocalReferenceTransform	The Transform component of the spring arm local reference object. There must be a SpringArmLocalReference object under the SpringArm object in the Hierarchy. See the examples of “Helicopter.prefab” and “Drone.prefab” in Assets\FlyingSystem\Demo\Prefabs\Controllers\AirTransportations folder.
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

Turning Attributes	
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.

Horizontal Movement Attributes	
normalHorizontalFlyingSpeed	The speed of normal horizontal flying.
maximumHorizontalFlyingSpeed	The maximum speed of horizontal flying in boost mode.
horizontalBoostAcceleration	It defines how fast the speed will increase from normalHorizontalFlyingSpeed to maximumHorizontalFlyingSpeed .
horizontalSlowDownAcceleration	It defines how fast the speed will decrease from maximumHorizontalFlyingSpeed to normalHorizontalFlyingSpeed .
maximumMeshPitchAngle	The maximum pitch angle.
meshPitchTurningSmoothingFactor	The interpolated parameter for pitch turning. The smaller the value, the smoother and slower the turning will be.
meshPitchTurningRecoverySmoothingFactor	The interpolated parameter for pitch back to 0. The smaller the value, the smoother and slower the turning will be.
maximumMeshRollAngle	The maximum roll angle.
meshRollTurningSmoothingFactor	The interpolated parameter for roll turning. The smaller the value, the smoother and slower the turning will be.
meshRollTurningRecoverySmoothingFactor	The interpolated parameter for roll back to 0. The smaller the value, the smoother and slower the turning will be.
speedTailRotorAngleRatioAnimationCurve	An editable curve of the angle between speed and tail rotor. Helicopter can move in any horizontal direction, but normally the greater angle to the forward direction, the slower speed.

Vertical Movement Attributes	
normalVerticalFlyingSpeed	The speed of normal vertical flying.
maximumVerticalFlyingSpeed	The maximum speed of vertical flying in boost mode.
verticalBoostAcceleration	It defines how fast the speed will increase from normalVerticalFlyingSpeed to maximumVerticalFlyingSpeed .
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.
horizontalFlyingSpeed	The horizontal flying speed.
verticalFlyingSpeed	The vertical flying speed.
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()	Make the aircraft take off.
Land()	Make the aircraft land.
VerticalSlowDown()	Make the aircraft slow down vertically.
AddYawInput(float value)	Make the aircraft turn left/right.
StopYawInput	Make the aircraft stop turning left/right.
AddHorizontalInput(Vector2 direction)	Make the aircraft move horizontally.
AddVerticalInput(float value)	Make the aircraft move vertically, the value can be positive or negative.
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(Helicopter.fbx, Drone.fbx) are the original creation by the developer of this project.