Creature Flying System Documentation

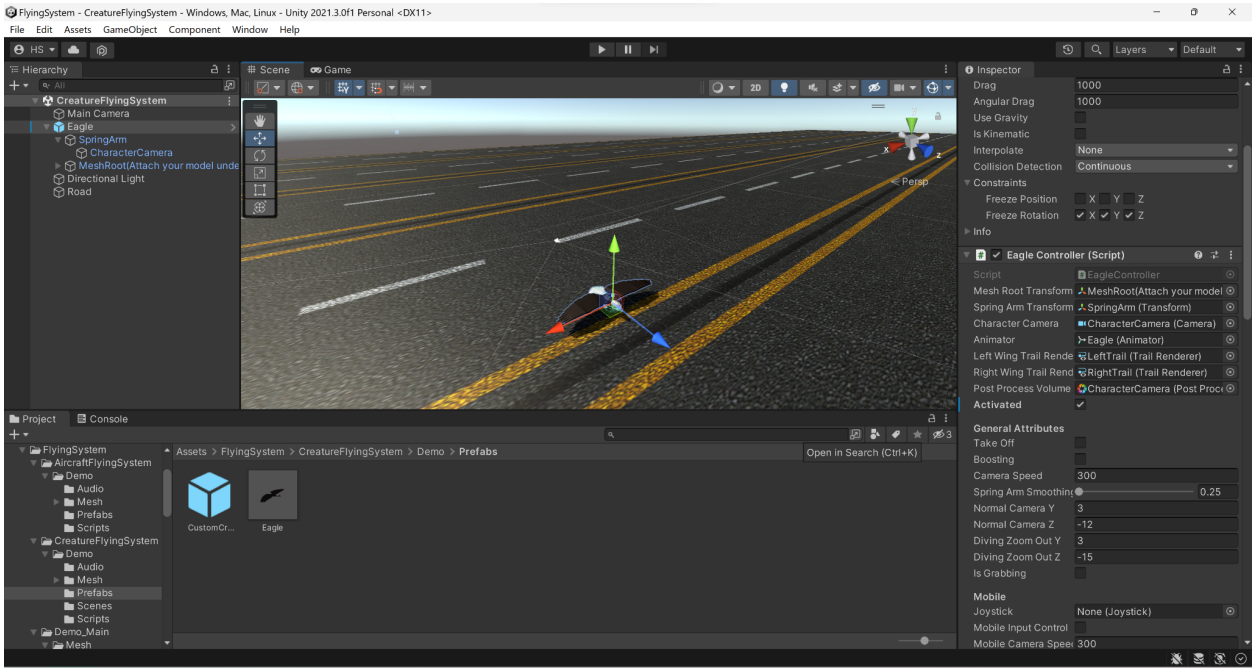
Description

This system is used for simulating flying behaviors of birds and mythical creatures that can fly. Diving and gliding are supported.

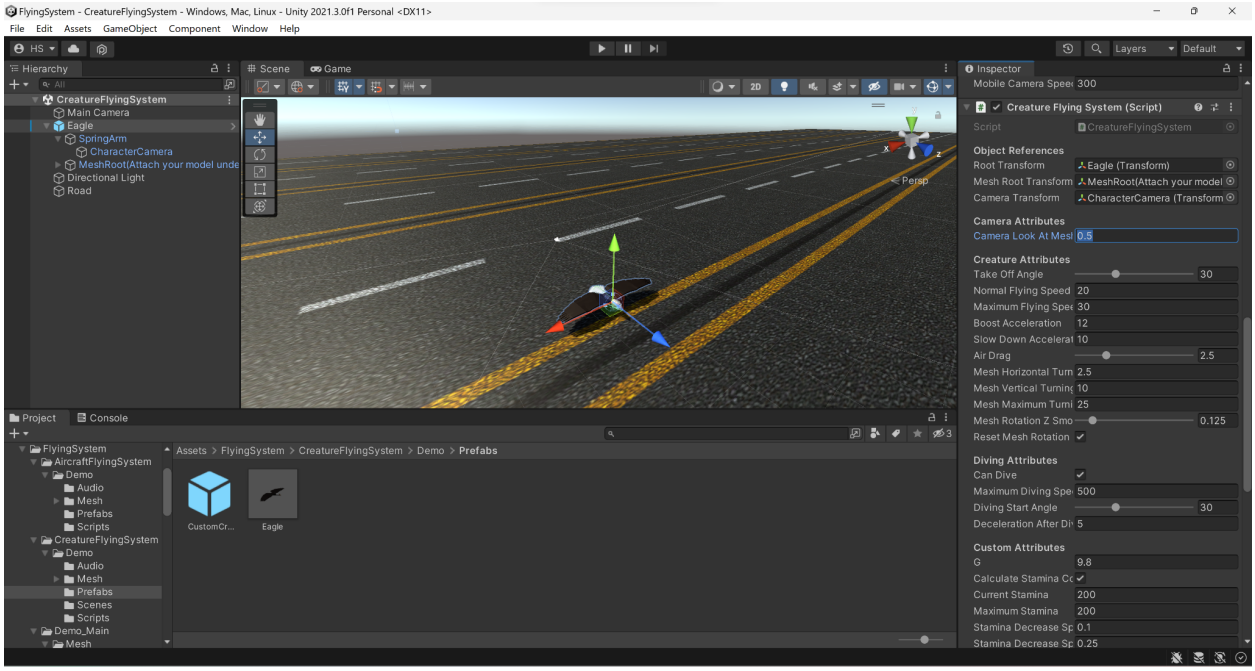
How to dive and glide: When controlling the flying creature, move the camera to look down until reaching some degree, then the creature will start accelerating and diving. Move the camera to look up before the creature touches the ground, it will gain horizontal speed and glide for a certain distance.

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\CreatureFlyingSystem\Demo\Prefabs folder, drag the Eagle prefab into the scene, select “Activated” for the EagleController script on the Inspector panel.



4. Press the Play button, the flyer would be controllable. Depending on the need, the attributes and methods in Creature 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:

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

1. Eagle: 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. Eagle(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.
meshRootTransform	The Transform component of the mesh root object.
cameraTransform	The Transform component of the third-person camera.

Adjustable Attributes

Camera Attributes

cameraLookAtMeshOffsetY	This will add an offset above the creature mesh and use it as the flying direction.
Creature Attributes	
takeOffAngle	The angle to take off. If it is 0, the creature will take off and fly horizontally. If it is 30, the flying direction will be 30 degrees above the horizon.
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.
slowDownAcceleration	It defines how fast the speed will decrease from maximumFlyingSpeed to normalFlyingSpeed .
airDrag	It only affects the vertical direction of gliding, the greater the value, the smaller vertical acceleration will be.
meshHorizontalTurningSpeed	It defines how fast the creature turns left/right.
meshVerticalTurningSpeed	It defines how fast the creature looks up/down.
meshMaximumTurningRotationZ	It defines the roll value when the creature turns left/right. Mainly used for matching the animation to achieve realisticness.
meshRotationZSmoothingFactor	The interpolated parameter for meshMaximumTurningRotationZ. The smaller the value, the smoother and slower the roll turning will be.
resetMeshRotationAfterLanding	Whether resets all turning values or not after the creature lands.
Diving Attributes	

canDive	Whether the creature can dive or not.
maximumDivingSpeed	The creature can not dive faster than this value.
divingStartAngle	The minimum angle to start the diving. If it is 30, it will start diving when the camera looks down 30 degrees below the horizon.
decelerationAfterDiving	It defines how fast the speed decreases when finishing diving. The smaller the value, the longer distance it can glide.
Custom Attributes	
g	The value of gravity acceleration, only affects the diving.
calculateStaminaConsumption	Whether calculate the stamina consumption of the creature or not.
currentStamina	The current stamina of the creature.
maximumStamina	The maximum stamina of the creature.
staminaDecreaseSpeed	It defines how fast the stamina drops when flying.
staminaDecreaseSpeedWhenBoosting	It defines how fast the stamina drops when flying in boost mode.
staminaRecoverySpeed	It defines how fast the stamina recovers when stops flying(in airflow or landed).
speedTirednessRatioAnimationCurve	An editable curve of the relation between speed and stamina. Normally the lower stamina, the slower speed.
calculateCarryingWeight	Whether calculate the carrying weight of the creature or not.
currentCarryingWeight	The current carrying weight of the creature.
maximumCarryingWeight	The maximum carrying weight of the creature.

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.
inAirflow	Whether it is inside the airflow or not.
stopFlying	Whether it stops flying, only possible when inside the airflow
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.
slowingDown	Whether it is slowing down.
staminaPercentage	The percentage of stamina, equals 1 when stamina is full.
weightPercentage	The percentage of carrying weight, equals 1 when fully carried.

Methods

TakeOff()	Make the creature take off.
Land()	Make the creature land.
FlyForward()	Make the creature fly forward.

SlowDown()	Make the creature slow down.
StopSlowingDown()	Make the creature stop slowing down.
AddYawInput(float value)	Make the creature turn left/right.
AddAirflowForce(float intensity, float acceleration, float fadeOutAcceleration)	Add an airflow force to the creature.
EndAirflowForce()	Stop the airflow force to the creature.
AddWeight(float increaseValue)	Increase the current carrying weight of the creature, 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

Some 3D models in public domain(CC0 license) were used in this project.

Duck: https://opengameart.org/content/rigged-duck
Eagle: https://opengameart.org/content/animated-animales-low-poly
Dragon: https://opengameart.org/content/low-poly-ice-dragon