CameraController

Namespace : `CameraSystem`

Inherits

[‘MonoBehaviour’]

Descrition

The CameraController class is responsible for controlling the camera in Unity.

Definition

```
public class CameraController : MonoBehaviour
```

Members

`m_Target` (Transform)

A reference to a Transform object that represents the camera’s target object.

```
[SerializeField] private Transform m_Target;
```

`m_distance` (float)

The distance between the camera and its target object.

```
[SerializeField] private float m_distance;
```

`m_offset` (Vector2)

A Vector2 that represents the camera’s offset from its target object in the x and y directions.

```
[SerializeField] private Vector2 m_offset;
```

`m_assetName` (string)

Ser Member

```
[SerializeField] private string m_assetName;
```

`m_assetPath` (string)

Ser Member

```
[SerializeField] private string m_assetPath;
```

`m_cameraLerpTime` (float)

The time it takes for the camera to interpolate between its current position and its target position

```
[SerializeField] private float m_cameraLerpTime;
```

`m_sensitivity` (Vector2)

A Vector2 that represents the camera's sensitivity to movement in the x and y directions.

```
[SerializeField] private Vector2 m_sensitivity;
```

`m_useYawLimit` (bool)

A bool that indicates whether or not the camera should be limited by a minimum and maximum yaw value.

```
[SerializeField] private bool m_useYawLimit;
```

`m_usePitchLimit` (bool)

A bool that indicates whether or not the camera should be limited by a minimum and maximum pitch value.

```
[SerializeField] private bool m_usePitchLimit;
```

`m_enableCameraCollision` (bool)

A bool that indicates whether or not the camera should detect and avoid collisions with other objects in the scene.

```
[SerializeField] private bool m_enableCameraCollision;
```

`m_cameraCollisionLayer` (LayerMask)

A LayerMask that determines which layers the camera should collide with.

```
[SerializeField] private LayerMask m_cameraCollisionLayer;
```

`m_active` (bool)

A bool that indicates whether or not the camera is currently active.

```
[SerializeField] private bool m_active;
```

`m_transitionTime` (float)

The time it takes for the camera to transition between two states.

```
[SerializeField] private float m_transitionTime;
```

`m_TransitionCurve` (AnimationCurve)

An AnimationCurve that determines the curve of the camera's transition animation.

```
[SerializeField] private AnimationCurve m_TransitionCurve;
```

`m_CameraSettingsToLoad` (CameraSettings)

A CameraSettings object that stores camera settings.

```
[SerializeField] private CameraSettings m_CameraSettingsToLoad;
```

`m_CameraType` (CameraType)

An enum that represents the type of camera being used.

```
[SerializeField] private CameraType m_CameraType;
```

`m_TargetLockOn` (Transform)

A reference to a Transform object that represents the target object the camera should lock on to.

```
[SerializeField] private Transform m_TargetLockOn;
```

`m_yaw` (float)

Current camera yaw.

```
[SerializeField] private float m_yaw;
```

`m_pitch` (float)

Current camera pitch.

```
[SerializeField] private float m_pitch;
```

Properties

`Active` (bool)

A bool that indicates whether or not the camera is currently active.

```
public bool Active { get; }
```

`Type` (CameraType)

An enum that represents the type of camera being used.

```
public CameraType Type { get; }
```

Methods

`SetPitchYaw` (void)

- Change camera's pitch and yaw

Arguments

- `look` (Vector2)
  - yaw, pitch increment

```
public void SetPitchYaw(Vector2 look)
```

`SetPitchYaw` (void)

- Change camera's pitch and yaw

Arguments

- x (float)
  - pitch increment
- y (float)
  - yaw increment

```
public void SetPitchYaw(float x, float y)
```

SetPitchYaw (void)

- Change camera's pitch and yaw (uses legacy input system)

```
public void SetPitchYaw()
```

ThirdPersonCamera (void)

- Third Person Camera

```
public void ThirdPersonCamera()
```

ActivateLockOn (void)

- Activates camera lock

Arguments

- targetLock (Transform)
  - target to lock on

```
public void ActivateLockOn(Transform targetLock)
```

DeactivateLockOn (void)

- Remove camera lock

```
public void DeactivateLockOn()
```

SetCameraSettings (void)

- Set Camera new settings immediately

Arguments

- settings (CameraSettings)
  - camera's new settings

```
public void SetCameraSettings(CameraSettings settings)
```

BlendBetweenCameraSettings (void)

- Activate transition between old camera settings and new camera settings.

Arguments

- `settings` (CameraSettings)
  - new camera settings to transition to

```
public void BlendBetweenCameraSettings(CameraSettings settings)
```

`StopBlend` (void)

- Stop camera transition

```
public void StopBlend()
```