RCC_Camera

Public Variables

public CameraTarget cameraTarget = new CameraTarget();   // Target of the camera, which is our player vehicle with custom class. Can be assigned manually with "SetTarget" method.

public bool isRendering = true;   // Currently rendering?

public Camera actualCamera;   // Camera is not attached to this main gameobject. Camera is parented to pivot gameobject. Therefore, we can apply additional position and rotation changes.

public GameObject pivot;   // Pivot center of the camera. Used for making offsets and collision movements.

// Camera Modes.
public CameraMode cameraMode = CameraMode.TPS;

public bool TPSLockX = true;   // Locks X angle to vehicle's X.

public bool TPSLockY = true;   // Locks Y angle to vehicle's Y.

public bool TPSLockZ = true;   // Locks Z angle to vehicle's Z.

public bool TPSFreeFall = true;   // Camera rotation won't track vehicle if it's not grounded.

public bool TPSDynamic = false;   // Use dynamic distance, height, and pitch angle related to vehicle rigidbody velocity.

public bool useTopCameraMode = false;   // Shall we use top camera mode?

public bool useHoodCameraMode = true;   // Shall we use hood camera mode?

public bool useOrbitInTPSCameraMode = true;   // Shall we use orbit control in TPS camera mode?

public bool useOrbitInHoodCameraMode = true;   // Shall we use orbit control in hood camera mode?

public bool useWheelCameraMode = true;   //Shall we use wheel camera mode?

public bool useFixedCameraMode = true;   // Shall we use fixed camera mode?

public bool useCinematicCameraMode = true;   // Shall we use cinematic camera mode?

public bool useOrthoForTopCamera = false;   // Shall we use ortho in top camera mode?

public bool useOcclusion = true;   // Shall we use camera occlusion?

public LayerMask occlusionLayerMask = -1;   // Camera will be ocluded by these layers.

```
public bool useAutoChangeCamera = false;        // Shall we change camera mode by auto? (For
cinematics)

public Vector3 topCameraAngle = new Vector3(45f, 45f, 0f);      // We will use this Vector3 angle for
top camera mode if it's enabled.

public float topCameraDistance = 100f;            // Top camera height / distance.

public float maximumZDistanceOffset = 10f;      // Distance offset for top camera mode. Related with
vehicle speed. If vehicle speed is higher, camera will move to front of the vehicle.

public float TPSDistance = 6f;     // The distance for TPS camera mode.

public float TPSHeight = 2f;       // The height we want the camera to be above the target for TPS
camera mode.

public float TPSRotationDamping = .7f;      // Rotation movement damper.

public float TPSTiltMaximum = 15f;     // Maximum tilt angle related with rigidbody local velocity.

public float TPSTiltMultiplier = 1.5f;     // Tilt angle multiplier.

public float TPSYaw = 0f;       // Yaw angle.

public float TPSPitch = 10f;     // Pitch angle.

public bool TPSAutoFocus = true;       // Auto focus to player vehicle. Adjusts distance and height
depends on vehicle bounds.

public bool TPSAutoReverse = true;      // Auto reverse when player vehicle is at reverse gear.

public bool TPSCollision = true;       // Collision effect when player vehicle crashes.

public Vector3 TPSOffset = new Vector3(0f, 0f, .2f);       // TPS position offset.

public Vector3 TPSStartRotation = new Vector3(0f, 0f, 0f);      // Rotation of the camera will be this
when game starts.

internal float targetFieldOfView = 60f;     // Camera will adapt its field of view to this target field of
view. All field of views below this line will feed this value.

public float TPSMinimumFOV = 40f;     // Minimum field of view related with vehicle speed.

public float TPSMaximumFOV = 60f;       // Maximum field of view related with vehicle speed.

public float hoodCameraFOV = 60f;       // Hood field of view.

public float wheelCameraFOV = 60f;      // Wheel field of view.

public float minimumOrtSize = 10f;         // Minimum ortho size related with vehicle speed.
public float maximumOrtSize = 20f;        // Maximum ortho size related with vehicle speed.

internal int cameraSwitchCount = 0;    // Used in switch case for running corresponding camera mode method.

public float zoomScrollMultiplier = 5f;      //  Zoom scroll multiplier.

public float minimumScroll = 0f;       //  Minimum distane for the zoom.

public float maximumScroll = 5f;       //  Maximum distance for the zoom.

// Minimum and maximum Orbit X, Y degrees.
public float minOrbitY = -15f;
public float maxOrbitY = 70f;

//          Orbit X and Y speeds.
public float orbitXSpeed = 100f;
public float orbitYSpeed = 100f;
public float orbitSmooth = 40f;

//          Resetting orbits.
public bool orbitReset = false;

public bool lookBackNow = false;       //  Camera is looking back now?
```

Public Methods

```
public void ToggleCamera(bool state) {} //          Toggles the camera with on / off state.

public void OnDrag(float x, float y) {}      //          Drags the camera orbit with x and y values.

public void ChangeCamera(CameraMode mode) {}          //          Changes the camera mode to given mode.

public void ChangeCamera() {}   //          Switches to next camera mode.

public void RemoveTarget() {}   //          Removes the camera target. Camera will no longer follow the target.

public void SetTarget(RCC_CarControllerV3 player) {}      //          Sets camera target.
```

Events

```
//  Event when camera spawned.
public delegate void onBCGCameraSpawned(GameObject BCGCamera);
public static event onBCGCameraSpawned OnBCGCameraSpawned;
```