Car Controller tutorial

This shows how to make a controller for a car model.

1. Import Car Model

A Car Model imported requires 4 wheels colliders If nonvisible follow the next steps:

1. Create an Empty parent and create four empty children attaching a wheel collider to each
2. Name each empty after a wheel on the car, EG. FrontLeftWheel, FrontRightWheel
3. Move each collider to its corresponding wheel.

2. Create the car controller and grab the Input

Create a new script called ***CarController*** and delete both void Start and void Update.

Add a float for horizontal and vertical input(**horizontalInput, verticalInput**) and create a void FixedUpdate() and GetInput() methods

The FixedUpdate Input is where we will put all of our methods and the GetInput() is where we'll store our keyboard inputs so let's start with that:

```
private void GetInput()
    {
        horizontalInput = Input.GetAxis("Horizontal");
        verticalInput = Input.GetAxis("Vertical");
        IsBreaking = Input.GetKey(KeyCode.Space);
    }
```

This will let us get each of the inputs that we will need.

3. Put references to all wheel components and others.

We will need to grab a reference to the wheel components and transforms to do that we will need four wheel colliders and four wheel transforms.

```
  [SerializeField] private WheelCollider  FrontLeftWheeleCollider;
    [SerializeField] private WheelCollider FrontRightWheeleCollider;
    [SerializeField] private WheelCollider rearLeftWheeleCollider;
    [SerializeField] private WheelCollider rearRightWheeleCollider;
    [SerializeField] private Transform FrontLeftWheeleTransform;
    [SerializeField] private Transform FrontRightWheeleTransform;
    [SerializeField] private Transform rearLeftWheeleTransform;
    [SerializeField] private Transform rearRightWheeleTransform;
```

To keep the code clean we will also need multiple other references such as:

```
private float currentSteerAngle;
    private float BreakForce;
    private bool IsBreaking;
```

and

```
private float motorForce;
private float currentBreakForce;
private float MaxSteerAngle;
```

4. Create Motion and Breaking

So now we would apply motion to the front wheels divergent to the vertical input and is breaking function which, we will use later.

So start by creating a void HandleMotor() and putting this in it:

```
        FrontLeftWheeleCollider.motorTorque = verticalInput * motorForce;
        FrontRightWheeleCollider.motorTorque = verticalInput * motorForce;
        currentBreakForce = IsBreaking ? BreakForce : 0f;
        if (IsBreaking)
        {
            applyBreaking();
        }
```

And we'll use applyBreaking() to assign the currentbreakforce to the breakTorque.

```
FrontRightWheeleCollider.brakeTorque = currentBreakForce;
        FrontLeftWheeleCollider.brakeTorque = currentBreakForce;
        rearRightWheeleCollider.brakeTorque = currentBreakForce;
        rearLeftWheeleCollider.brakeTorque = currentBreakForce;
```

5. Handle the Steering

Now we will control the actual movement of the car by using the steerAngle as a way to coordinate the way the front wheels, being limited by its maxSteeringAngle.

So we will do this through creating yet another void HandleSteering().

```
steerAngle = maxSteeringAngle * horizontalInput;
        frontLeftWheelCollider.steerAngle = steerAngle;
        frontRightWheelCollider.steerAngle = steerAngle;
```

6. Wheel Updater

The code below will change the wheel position for one certain wheel and it will be applied to other wheels through the UpdateWheelPos void

```
private void UpdateWheelPos(WheelCollider wheelCollider, Transform trans)
    {
        Vector3 pos;
        Quaternion rot;
        wheelCollider.GetWorldPose(out pos, out rot);
        trans.rotation = rot;
        trans.position = pos;
    }
```

Now we will apply that to each wheel

```
    private void UpdateWheels()
    {
        UpdateWheelPos(frontLeftWheelCollider, frontLeftWheelTransform);
        UpdateWheelPos(frontRightWheelCollider, frontRightWheelTransform);
        UpdateWheelPos(rearLeftWheelCollider, rearLeftWheelTransform);
        UpdateWheelPos(rearRightWheelCollider, rearRightWheelTransform);
    }
```

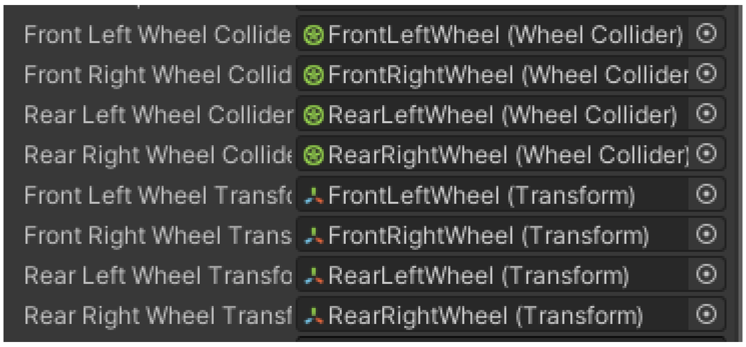
7. Recall each component and set up the wheels to their code reference

Now we will recall each void into the FixedUpdate()

```
    private void FixedUpdate()
    {
        GetInput();
        HandleMotor();
        HandleSteering();
        UpdateWheels();
    }
```

Apply the script onto the car.

Now Put each Wheel component into its allocated slot and use either WASD or the arrow keys to move it around.



Front Left Wheel Collide	FrontLeftWheel (Wheel Collider)
Front Right Wheel Collid	FrontRightWheel (Wheel Collider
Rear Left Wheel Collider	RearLeftWheel (Wheel Collider)
Rear Right Wheel Collide	RearRightWheel (Wheel Collider)
Front Left Wheel Transfo	FrontLeftWheel (Transform)
Front Right Wheel Trans	FrontRightWheel (Transform)
Rear Left Wheel Transfo	RearLeftWheel (Transform)
Rear Right Wheel Transf	RearRightWheel (Transform)

Adjust mass of car and other settings through the inspector accordingly.