Challenge 1
Plane Programming

Challenge Overview:
Use the skills you learned in the driving simulation to fly a plane around obstacles in the sky. You will have to get the user’s input from the up and down arrows in order to control the plane’s pitch up and down. You will also have to make the camera follow alongside the plane so you can keep it in view.

Challenge Outcome:
- The plane moves forward at a constant rate
- The up/down arrows tilt the nose of the plane up and down
- The camera follows along beside the plane as it flies

Challenge Objectives:
In this challenge, you will reinforce the following skills/concepts:
- Using the Vector3 class to move and rotate objects along/around an axis
- Using Time.deltaTime in the Update() method to move objects properly
- Moving and rotating objects in scene view to position them the way you want
- Assigning variables in the inspector and initializing them in code
- Implementing Input variables to control the movement/rotation of objects based on User input

Challenge Instructions:
- Open your Prototype 1 project
- Download the "Challenge 1 Starter Files" from the Tutorial Materials section, then double-click on it to Import
- In the Project Window > Assets > Challenge 1 > Instructions folder, use the Outcome video as a guide to complete the challenge
<table>
<thead>
<tr>
<th>Challenge</th>
<th>Task</th>
<th>Hint</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The plane is going backwards</td>
<td>Make the plane go forward</td>
</tr>
<tr>
<td>2</td>
<td>The plane is going too fast</td>
<td>Slow the plane down to a manageable speed</td>
</tr>
<tr>
<td>3</td>
<td>The plane is tilting automatically</td>
<td>Make the plane tilt only if the user presses the up/down arrows</td>
</tr>
<tr>
<td>4</td>
<td>The camera is in front of the plane</td>
<td>Reposition it so it’s beside the plane</td>
</tr>
<tr>
<td>5</td>
<td>The camera is not following the plane</td>
<td>Make the camera follow the plane</td>
</tr>
</tbody>
</table>

**Bonus Challenge**

<table>
<thead>
<tr>
<th>Task</th>
<th>Hint</th>
</tr>
</thead>
<tbody>
<tr>
<td>The plane’s propeller does not spin</td>
<td>Create a script that spins the plane’s propeller</td>
</tr>
</tbody>
</table>
Challenge Solution

1. In PlayerControllerX.cs, in Update, change `Vector3.back` to `Vector3.forward`

   ```csharp
   // move the plane forward at a constant rate
   transform.Translate(Vector3.back * speed);
   ```

2. In PlayerControllerX.cs, in Update, add `* Time.deltaTime` to the Translate call

   ```csharp
   // move the plane forward at a constant rate
   transform.Translate(Vector3.forward * speed * Time.deltaTime);
   ```

3. In PlayerControllerX.cs, include the `verticalInput` variable to the Rotate method:

   ```csharp
   // tilt the plane up/down based on up/down arrow keys
   transform.Rotate(Vector3.right * rotationSpeed * verticalInput * Time.deltaTime);
   ```

4. Change the camera’s position to (30, 0, 10) and its rotation, to (0, -90, 0)

5. To assign the `plane` variable, select Main Camera in the hierarchy, then drag the Plane object onto the “Plane” variable in the inspector

   - To assign the `offset` variable, add the value as a new Vector3 at the top of FollowPlane.cs:
     ```csharp
     private Vector3 offset = new Vector3(30, 0, 10);
     ```
**Bonus Challenge Solution**

**X1** Create a new Script called “SpinPropellerX.cs” and attach it to the “Propellor” object (which is a child object of the Plane):

![Image of Unity editor with a new script added](image)

**X2** In RotatePropellerX.cs, add a new propellorSpeed variable and Rotate the propeller on the Z axis

```csharp
private float propellorSpeed = 1000;

void Update()
{
    transform.Rotate(Vector3.forward, propellorSpeed * Time.deltaTime);
}
```