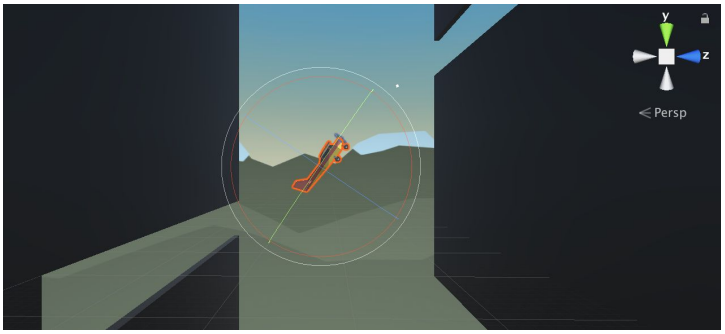
unity

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

Challenge		Task	Hint
1	The plane is going backwards	Make the plane go forward	`Vector3.back` makes an object move backwards, `Vector3.forward` makes it go forwards
2	The plane is going too fast	Slow the plane down to a manageable speed	If you multiply a value by `Time.deltaTime`, it will change it from 1x/frame to 1x/second
3	The plane is tilting automatically	Make the plane tilt only if the user presses the up/down arrows	In PlayerControllerX.cs, in Update(), the `verticalInput` value is assigned, but it's never actually used in the `Rotate()` call
4	The camera is *in front of* the plane	Reposition it so it's beside the plane	For the camera's position, try `X=30, Y=0, Z=10` and for the camera's rotation, try `X=0, Y=-90, Z=0`
5	The camera is not following the plane	Make the camera follow the plane	In FollowPlayerX.cs, neither the plane nor offset variables are assigned a value - assign the `plane` variable in the camera's inspector and assign the `offset = new Vector3(0, 30, 10)` in the code

Bonus Challenge		Task	Hint
X	The plane's propeller does not spin	Create a script that spins the plane's propeller	There is a "Propeller" child object of the plane - you should create a new "SpinPropellerX.cs" script and make it rotate every frame around the Z axis.

Challenge Solution

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

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

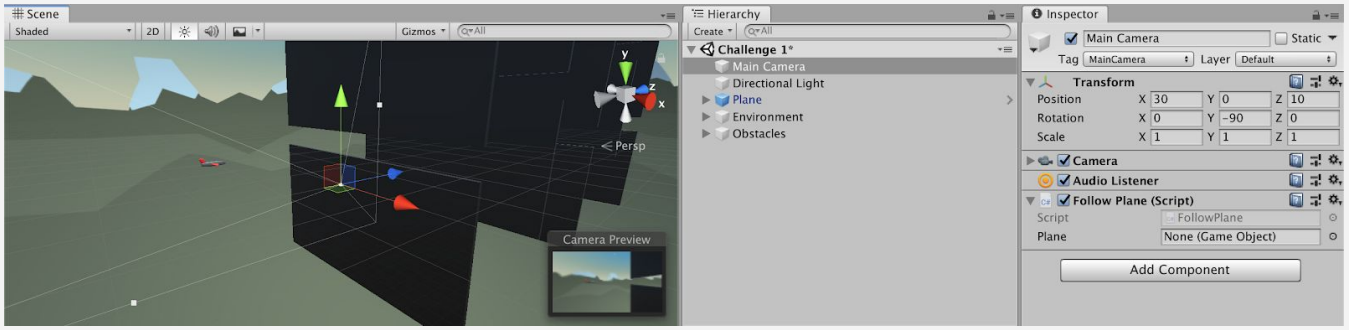
2 In PlayerControllerX.cs, in Update, add * *Time.deltaTime* to the Translate call

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

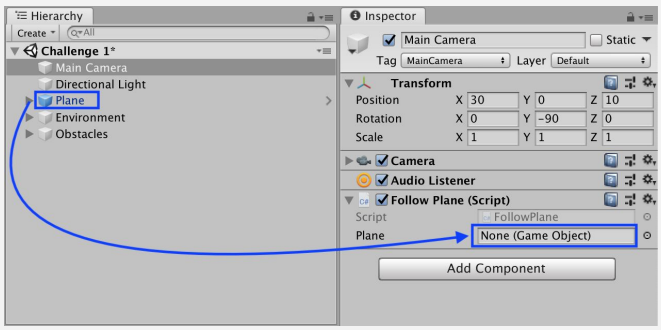
3 In PlayerControllerX.cs, include the *verticalInput* variable to the Rotate method:

```
// 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

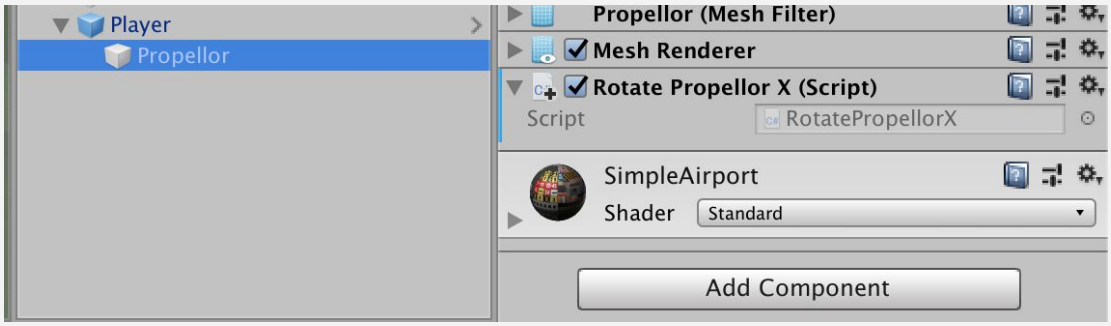


To assign the *offset variable,* add the value as a new Vector3 at the top of FollowPlane.cs:

```
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):



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

```
private float propellorSpeed = 1000;

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