unity

1.4 Step into the Driver's Seat

Steps:

Step 1: Allow the vehicle to move left/right

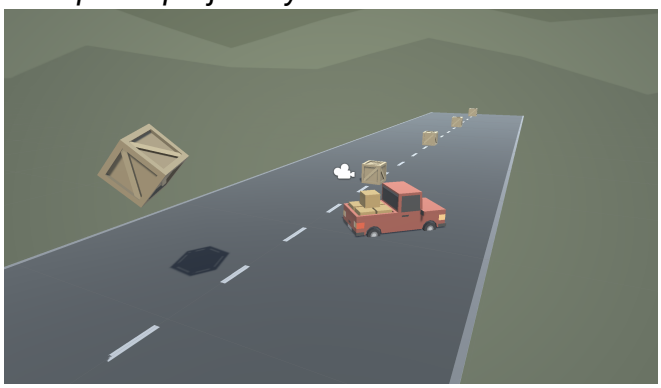
Step 2: Base left/right movement on input

Step 3: Take control of the vehicle speed

Step 4: Make vehicle rotate instead of slide

Step 5: Clean your code and hierarchy

Example of project by end of lesson



Length: 50 minutes

Overview: In this lesson, we need to hit the road and gain control of the vehicle. In order to do so, we need to detect when the player is pressing the arrow keys, then accelerate and turn the vehicle based on that input. Using new methods, Vectors, and variables, you will allow the vehicle to move forwards or backwards and turn left to right.

Project Outcome: When the player presses the up/down arrows, the vehicle will move forward and backward. When the player presses the left/right arrows, the vehicle will turn.

Learning Objectives: By the end of this lesson, you will be able to:

- Gain user input with Input.GetAxis, allowing the player to move in different ways
- Use the Rotate function to rotate an object around an axis
- Clean and organize your hierarchy with Empty objects

Step 1: Allow the vehicle to move left/right

Until now, the vehicle has only been able to move straight forward along the road. We need it to be able to move left and right to avoid the obstacles.

1. At the top of PlayerController.cs, add a ***public float turnSpeed;*** variable
2. In ***Update()***, add ***transform.Translate(Vector3.right * Time.deltaTime * turnSpeed);***
3. Run your game and use the ***turnSpeed*** **variable slider** to move the vehicle left and right

- **New Function:** Vector3.right

```
public float turnSpeed;

void Update()
{
  transform.Translate(Vector3.forward * Time.deltaTime * speed);
  transform.Translate(Vector3.right * Time.deltaTime * turnSpeed);
}
```

Step 2: Base left/right movement on input

Currently, we can only control the vehicle's left and right movement in the inspector. We need to grant some power to the player and allow them to control that movement for themselves.

1. From the top menu, click **Edit** > **Project Settings**, select **Input Manager** in the left sidebar, then expand the **Axes** fold-out to explore the inputs.
2. In **PlayerController.cs**, add a new ***public float horizontalInput*** variable
3. In ***Update***, assign ***horizontalInput = Input.GetAxis("Horizontal");***, then test to see it in inspector
4. Add the ***horizontalInput*** variable to your left/right **Translate method** to gain control of the vehicle
5. In the Inspector, edit the ***turnSpeed*** and ***speed*** variables to tweak the feel

- **New:** Input.GetAxis
- **Tip:** Edit > Project Settings > Input and expand the Horizontal Axis to show everything about it
- **Warning:** Spelling is important in string parameters. Make sure you spell and capitalize "Horizontal" correctly!

```
public float horizontalInput;

void Update()
{
  horizontalInput = Input.GetAxis("Horizontal");

  transform.Translate(Vector3.forward * Time.deltaTime * speed);
  transform.Translate(Vector3.right * Time.deltaTime * turnSpeed * horizontalInput);
}
```

Step 3: Take control of the vehicle speed

We've allowed the player to control the steering wheel, but we also want them to control the gas pedal and brake.

1. Declare a new public ***forwardInput*** variable
2. In ***Update***, assign ***forwardInput = Input.GetAxis("Vertical");***
3. Add the ***forwardInput*** variable to the **forward Translate method**, then test

- **Tip:** It can go backwards, too!
- **Warning:** This is slightly confusing with forwardInput and vertical axis

```
public float horizontalInput;
public float forwardInput;

void Update()
{
  horizontalInput = Input.GetAxis("Horizontal");
  forwardInput = Input.GetAxis("Vertical");

  transform.Translate(Vector3.forward * Time.deltaTime * speed * forwardInput);
  transform.Translate(Vector3.right * Time.deltaTime * turnSpeed * horizontalInput);
}
```

Step 4: Make vehicle rotate instead of slide

There's something weird about the vehicle's movement… it's slides left to right instead of turning. Let's allow the vehicle to turn like a real car!

1. In ***Update***, call ***transform.Rotate(Vector3.up, horizontalInput)***, then test
2. **Delete** the line of code that **translates Right**, then test
3. Add **** turnSpeed * Time.deltaTime***, then test

- **New:** transform.Rotate
- **Tip:** You can always trust the official Unity scripting API documentation

```
void Update()
{
  horizontalInput = Input.GetAxis("Horizontal");
  forwardInput = Input.GetAxis("Vertical");

  transform.Translate(Vector3.forward * Time.deltaTime * speed * forwardInput);
  transform.Rotate(Vector3.up, turnSpeed * horizontalInput * Time.deltaTime);
  transform.Translate(Vector3.right * Time.deltaTime * turnSpeed * horizontalInput);
}
```

Step 5: Clean your code and hierarchy

We added lots of new stuff in this lesson. Before moving on and to be more professional, we need to clean our scripts and hierarchy to make them more organized.

1. In the hierarchy, *Right-click > Create Empty* and rename it "Obstacles", then **drag** all the obstacles into it
2. **Initialize** variables with values in **PlayerController**, then make all variables **private** (except for the **player** variables)
3. Use **//** to add **comments** to each section of code

- **New:** Empty Object
- **Tip:** You don't actually need to type "private", it defaults to that
- **Tip:** Comments are important, especially for your future self

```
public private float speed = 20.0f;
public private float turnSpeed = 45.0f;
public private float horizontalInput;
public private float forwardInput;

void Update() {
  horizontalInput = Input.GetAxis("Horizontal");
  forwardInput = Input.GetAxis("Vertical");
  // Moves the car forward based on vertical input
  transform.Translate(Vector3.forward * Time.deltaTime * speed * forwardInput);
  // Rotates the car based on horizontal input
  transform.Rotate(Vector3.up, turnSpeed * horizontalInput * Time.deltaTime);
}
```

Lesson Recap

New Functionality	• When the player presses the up/down arrows, the vehicle will move forward and backward • When the player presses the left/right arrows, the vehicle turns
New Concepts and Skills	• Empty objects • Get user input • Translate vs Rotate