Development Equipment Used: Moza 9 and series products (steering wheel, accelerator, and brake pedal)

Equipment Settings: The steering wheel has a maximum steering angle of 720°, with 360° in both clockwise and counterclockwise directions. The accelerator provides linear acceleration.电脑游戏的截图

描述已自动生成

电脑游戏的截图

描述已自动生成

图形用户界面, 网站

描述已自动生成

Unity Version: 2021.3.12f1c2 (the latest input system package needs to be installed) Input System Version: 1.4.3 Official Driver: Moza pit house (for adjusting the maximum rotation angle of the steering wheel, linear acceleration of the throttle, etc.)

Using Moza buttons: Moza R9 base (steering wheel and turn signals): Left turn signal: button13 upper left pad Right turn signal: button14 upper right pad Steering wheel: stick

Moza SRP pedal (accelerator and brake, automatic transmission): Accelerator: Rx Brake: Ry

图片包含 图形用户界面

描述已自动生成

Unity Development, Data Reading

Utilize input action to obtain inputs.

In input action, the action type for left and right turn signals is 'button', while the action type for the accelerator and brake is 'Value', with the Control Type set to 'Any'. For the steering wheel, the Action Type is 'pass through', with Control Type as 'Any'.

Code Implementation

To implement car movement by obtaining parameters through code, the InputSystem needs to be referenced:

csharp

Copy code

using UnityEngine.InputSystem;

In the start or awake methods, bind the corresponding methods to be executed after the action is triggered. Since the left and right turn signals are triggered by buttons, the selected trigger type is 'started':

csharp

Copy code

actionTest.car.right.started += rightLight;

actionTest.car.left.started += leftLight;

Since the steering wheel, accelerator, and brake all continuously obtain data, the selected trigger type is 'performed':

csharp

Copy code

actionTest.car.accelerator.performed += movebefore;

actionTest.car.brake.performed += moveafter;

actionTest.car.direction.performed += direction;

Obtaining Inputs

Inputs are obtained as follows:

csharp

Copy code

void movebefore(InputAction.CallbackContext ctx)

{

accelerator = 0.5f \* ctx.ReadValue<float>();

rightTrigger = accelerator;

}

ctx contains all parameters during input, and the variable type in ReadValue needs to match the input type. No need to obtain inputs for left and right turn signals.

Input Values

The accepted values for the steering wheel (may vary depending on the device and Unity version) are:

Clockwise: starts at -1 and becomes 1 after one rotation.

Counterclockwise: starts at 1 and becomes -1 after one rotation.

The accepted values for the accelerator and brake (adjustable through the driver and may vary depending on the device and Unity version) are:

Accelerator: starts at 0 and maxes out at 2.

Brake: starts at -2 and maxes out at 0.