

## Mô hình MVC

Trong Unity, mô hình MVC (Model-View-Controller) cũng có các thành phần tương tự nhưng thường được tổ chức một cách linh hoạt và phù hợp với cách làm việc của Unity. Dưới đây là một cách tổ chức thông thường cho mô hình MVC trong Unity:

### Model:

Model đại diện cho dữ liệu của ứng dụng.

Có thể là các lớp hoặc kịch bản (script) chứa dữ liệu và phương thức để thao tác với dữ liệu đó.

Ví dụ: Một lớp Player để lưu trữ thông tin về người chơi như điểm số, mạng sống, vị trí, v.v.  
View:

View là phần giao diện người dùng của ứng dụng.

Thường là các đối tượng GameObject trong Unity chứa các thành phần UI như Text, Image, Button, v.v.

View không chứa bất kỳ logic ứng dụng nào mà chỉ hiển thị dữ liệu từ Model và tương tác với người chơi.

Ví dụ: Giao diện người chơi hiển thị điểm số và mạng sống của người chơi.

### Controller:

Controller là thành phần điều khiển luồng logic của ứng dụng.

Trong Unity, Controller thường là các kịch bản (script) gắn với GameObject để xử lý sự kiện và tương tác với Model và View.

Controller có thể điều khiển cả Model và View, quản lý trạng thái của ứng dụng và xử lý các sự kiện người dùng.

Ví dụ: Một kịch bản GameController điều khiển luồng logic của trò chơi, bao gồm việc cập nhật điểm số, xử lý va chạm và hiển thị thông báo cho người chơi.

### Communication:

Model và View không nên giao tiếp trực tiếp với nhau mà thông qua Controller.

Controller đóng vai trò trung gian, lấy dữ liệu từ Model và cập nhật dữ liệu lên View.

Điều này giữ cho các thành phần của ứng dụng độc lập và dễ dàng kiểm thử và bảo trì.

### ví dụ trong C# unity

#### -model

```
// NumberModel.cs
public class NumberModel
{
    private int number;

    public int GetNumber()
    {
        return number;
    }

    public void IncrementNumber()
```

```
    {  
        number++;  
    }  
}
```

**-View**

// NumberView.cs

using UnityEngine;

using UnityEngine.UI;

public class NumberView : MonoBehaviour

```
{  
    public Text numberText;  
  
    public void DisplayNumber(int number)  
    {  
        numberText.text = number.ToString();  
    }  
}
```

**-controller**

// NumberController.cs

using UnityEngine;

public class NumberController : MonoBehaviour

```
{  
    public NumberModel model;  
    public NumberView view;  
  
    private void Start()  
    {  
        model = new NumberModel();  
        view = GetComponent<NumberView>();  
    }  
  
    private void Update()  
    {  
        if (Input.GetKeyDown(KeyCode.Space))  
        {  
            model.IncrementNumber();  
            int currentNumber = model.GetNumber();  
            view.DisplayNumber(currentNumber);  
        }  
    }  
}
```