

Brief Documentation

Project Goal

To make at least playable prototype functional clothes shop within a simulation game reminiscent of 'The Sims' and 'Stardew Valley', with minor bug.

Deployment Environment

The game developed using Unity, with 2d pixel graphic and animation, the game deployed on WebGL build for testing.

Design Selected

Im here not using Architectural Design pattern like MVC to make this, because im not really experience using that pattern, even though im still using some Design pattern like Observer, State, and Singleton.

Im using **Observer** to control the player inputs, Im using **State** for controlling Animation, and **Singleton** to make attached script hold values then can be accessible across project.

Systems

There is some system and mechanic that implement in this prototype such as:

- Movement System
- Pickup Item Mechanic
- Simple Current Date System
- Wind Shader

- Post Processing
- Shop
- Currency
- Inventory
- Clothes

Highlight Showcase

Movement System, im using simple movement system using observer design pattern, actually this is just **WASD** or down, left, up, right arrow.

This is snipped code from movement system

```
[field: SerializeField]
public UnityEvent<Vector2> OnMovementKeyPressed { get; set; }
private void Update()
{
    GetMovementInput();
}
```

```
private void GetMovementInput()
{
    OnMovementKeyPressed?.Invoke(new Vector2(Input.GetAxisRaw("Horizontal"),
Input.GetAxisRaw("Vertical")));
}
```

So what happen here i create unity event and nulled check the event and invoke it. Same as Movement script im using unity event to pickup items:

```
private void GetPickupInput()
{
    if (Input.GetAxisRaw("Pickup") > 0)
    {
        onPickupKeyPressed?.Invoke();
    }
}
```

Why use **Wind Shader**, I want to make the prototype looks good my reference is direct to **Little Sim World**. I even i search music that have same vibes from **Little Sim World**. Also applying post processing is make game looks better.

For **Shop** i make simple shop that buy and sell, the shop also connected with the **inventory** too. On the shop of course use some currency to buy or sell item, here some snippet from currency system that i've been made

```
public class CurrencyManager : MonoBehaviour
{
    public static CurrencyManager instance;
    public int coin;
    public int Coin
    {
        get => coin;
        set
        {
            coin = value;
            UIManager.Instance.UpdateCoinUI(coin);
        }
    }

    private void Awake()
    {
        instance = this;
    }

    private void Start()
    {
        coin = 300;
    }

    private void Update()
    {
        UIManager.Instance.UpdateCoinUI(coin);
    }
}
```

```
}
```

For **Clothes** i make system that based on animation change, in this system take lot of time to spend because have to separate animation one by one, this system usually use for character generator but in this i implemented when user buy item clothes it will automatically equipped.

UI



Inventory UI have some preview character and inventory holder that hold the items.



Realtime Date UI and Coin Holder UI to count the Coins.



Shop panel UI that combined inventory with shop itself.

In summary those are what im do during the development of the prototype for 48 hours, in the end I'm just tried my best here.