



Options Dropper Tool

Codding For you

Documentation

1.0.0

Index

| | |
|--------------------|----------|
| Definitions | 2 |
| ODCommunicator | 2 |
| ODInterface | 2 |
| Set up | 3 |
| Custom UI | 6 |
| Support | 7 |

Definitions

ODCommunicator

This file will be the intermediary between the interface and the data controller. It can be created in the context menu of the Project window (Create > Options Dropper > Communicator).

ODInterface

This file stores the prefabs of the interface that will later be created automatically. It can be created in the context menu of the Project window (Create > Options Dropper > Interface). It has the following fields to be filled in:

- **Layout:** This is the prefab of the layout that will have the options menu. It requires the ODLLayout component.
- **Button tab:** This is the prefab of the options menu button that will open the different sections. Required to be TextMeshPro.
- **Label:** This is the prefab of the field labels. Required to be TextMeshPro.
- **Input Fields:** These are the prefabs of the fields that you can use for the options. Requires a component that inherits from Selectable.
- **Categories:** These are the categories that will divide the different sections.

Set up

1

To deploy Option Dropper Tool you must create an empty object inside a Canvas, then add the component **OptionDropperBehaviour**.

2

- In the **File** field you must add the configuration file you want to modify.
- In the **Communicator** field you must add the previously created one.
- (Optional) In the **Interface** field you must add the previously created one.

3

If you want to drop down all fields of the configuration file, click on the **Drop** button next to the File field.

4

Modify the fields to your preference. Assign the Input Fields you have in the scene you created or use the **Generate UI** button to create it from the Interface file.

If you use our **Generate UI** option, you will need to specify which **Input Field** to use from the list of prefabs you have added in the **Interface** file.

•

•

•

5

To use it in your data controller, the following lines of code will be required:

```
using UnityEngine;
using OptionsDropper;

...

[SerializeField] private ODCommunicator _communicator;
[SerializeField] private ConfigData _configData; // Your configuration class
...

private void Awake() {
    ...
    LoadConfig();
    // Assign configuration file on Communicator
    _communicator.File = _configData;
    // Subscribe onValueChangedEvent with your SaveConfig function
    _communicator.onValueChangedEvent.AddListener(SaveConfig);
    ...
}

private void LoadConfig() {
    if (!PlayerPrefs.HasKey("Config")) {
        _configData = new ConfigData();
        return;
    }

    _configData =
    JsonUtility.FromJson<ConfigData>(PlayerPrefs.GetString("Config"));
}
```


Custom UI

You can create your own custom interface based on the attached demo, remember to follow the rules mentioned above.

Support

For assistance, please contact the following e-mail address:

contact@sekruiis.com

Thank you for downloading Options Dropper Tool!