

# Simple CalendarDatePicker

*By Kha2Dev*

Documentation v1.0

Email : [rivayx@gmail.com](mailto:rivayx@gmail.com)

Telegram : @rifai44

## Description:

This package contains features that serve to make it easier for users to get the desired date. This package also contains sample scenes that are easy for users to follow.

## Setup:

To use simply add the "CalendarDatePicker" folder to your project (which is on the package). It doesn't need to be on root of your project, so feel free to move it wherever you want.

## How to use:

to use this package is quite easy. Users only need to copy "DemoSceneDatePicker" which is included in the package. Or by following these steps:

1. Add the following header to your script

```
using kha2dev.DatePicker;  
using System;
```

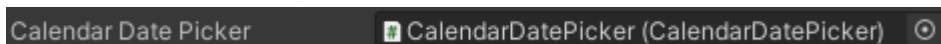
2. Add class "CalendarDatePicker" with class type public or you can use "[SerializeField]", as in the following example

```
[SerializeField] CalendarDatePicker calendarDatePicker;
```

3. Put the CalendarDatePicker prefabs in the "SimpleCalendarDatePicker/Prefabs/CalendarDatePicker" folder into the Canvas in your Scene, try to make this Object the Last Siblings on the Canvas, otherwise the Object will be covered by other objects.



4. Insert the added Prefabs into your Inspector script (the "CalendarDatePicker" class that was added earlier).



5. Create a function that is used as a callback with a "DateTime" return value. As in the following example.

```
private void CallbackDatePicker(DateTime result)
{
    textResult.text = "Result: " + result.ToString("dddd, dd MMMM yyyy");
}
```

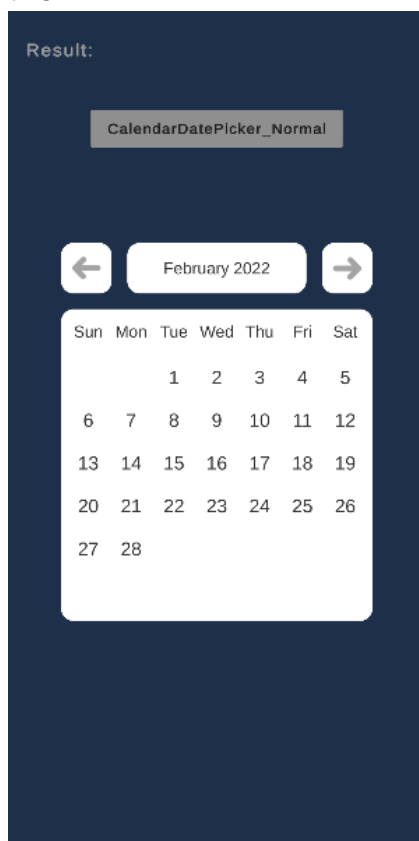
6. Create a function that is used to be registered on the Button. Inside this function please call the function to show the CalendarDatePicker and enter the Callback (CallbackDatePicker) that was created earlier.

```
public void OpenDatePicker_Normal()
{
    calendarDatePicker.Show(CallbackDatePicker);
}
```

7. Please enter the function you just created into the button, and it will run when the button is clicked. You can use either "Inspector" or "onClick.AddListener". In this case I use "onClick.AddListener" as in the following example.

```
private void Start()
{
    buttonOpenDatePicker_Normal.onClick.AddListener(OpenDatePicker_Normal);
}
```

8. If it has been saved, please run your application, and click the button that you registered to display the CalendarDatePicker earlier. if successful, it will appear like this



9. Please select a date and it will appear on the results that have been registered previously

Result: Wednesday, 09 February 2022

10. You can also deactivate the date, limit the date, by registering the class "ParameterDatePicker" and entering it when calling the "Show" function on the "CalendarDatePicker"

```
private void OpenDatePicker_Limited()
{
    List<DateTime>disableDate = new List<DateTime>();
    disableDate.Add(new DateTime(2022, 3, 15));
    disableDate.Add(new DateTime(2022, 4, 4));
    disableDate.Add(new DateTime(2022, 4, 21));

    ParameterDatePicker dateLimited = new ParameterDatePicker()
    {
        defaultDate = new DateTime(2022, 3, 1),
        startDateFrom = new DateTime(2022, 2, 20),
        finishDateUntil = new DateTime(2022, 5, 20),
        listDisableDate = disableDate,
        disableFriday = true,
        disableSunday = true
    };

    calendarDatePicker.Show(CallbackDatePicker, dateLimited);
}
```

11. Here are some features that can be used to disable dates:

```
/// <summary>
/// the month that will appear when CalendarDatePicker is opened for the first time
/// </summary>
24 references
public DateTime defaultDate = DateTime.Today;

/// <summary>
/// if startDateFrom is filled then, the date before startDateFrom will be disabled or unselectable
/// </summary>
2 references
public DateTime startDateFrom = DateTime.MinValue;

/// <summary>
/// if finishDateUntil is filled then, the date after finishDateUntil will be disabled or unselectable
/// </summary>
2 references
public DateTime finishDateUntil = DateTime.MaxValue;

/// <summary>
/// a list of dates that will be disabled or unselectable
/// </summary>
2 references
public List<DateTime> listDisableDate = new List<DateTime>();

/// <summary>
/// if disableSunday is set to true, every Sunday will be disabled or unselectable, if it is not set, it will still be selectable
/// </summary>
2 references
public bool disableSunday = false;

/// <summary>
/// if disableMonday is set to true, every Monday will be disabled or unselectable, if it is not set, it will still be selectable
/// </summary>
1 reference
public bool disableMonday = false;

/// <summary>
/// if disableTuesday is set to true, every Tuesday will be disabled or unselectable, if it is not set, it will still be selectable
/// </summary>
1 reference
public bool disableTuesday = false;

/// <summary>
/// if disableWednesday is set to true, every Wednesday will be disabled or unselectable, if it is not set, it will still be selectable
/// </summary>
1 reference
public bool disableWednesday = false;

/// <summary>
/// if disableThursday is set to true, every Thursday will be disabled or unselectable, if it is not set, it will still be selectable
/// </summary>
1 reference
public bool disableThursday = false;

/// <summary>
/// if disableFriday is set to true, every Friday will be disabled or unselectable, if it is not set, it will still be selectable
/// </summary>
2 references
public bool disableFriday = false;

/// <summary>
/// if disableSaturday is set to true, every Saturday will be disabled or unselectable, if it is not set, it will still be selectable
/// </summary>
1 reference
public bool disableSaturday = false;
```

12. You can also change the date color when enabled or disabled, as well as CultureInfo through the inspector in the "CalendarDatePicker" script.

