3D RESEARCH

unity

Flat Calendar

v1.0

User Guide

Intro

Flat Calendar is a simple and useful calendar with a modern and stylish interface similar to Google Material Design.

With Flat Calendar you can:

- Navigate in your calendar changing month, year and days
- Add unlimited number of events for a specific day
- Change theme or create your own style
- Be notified when an user interface event occour.

Flat Calendar is made with Unity 5 and is compatible with all platform.

Inside Flat Calendar

Flat Calendar comes with these components:

- **FlatCalendar.cs**: which rapresent the logic of Flat Calendar
- **FlatCalendarStyle.cs**: for theme management
- **FlatCalendarInspector.cs**: which override the Inspector of Flat Calendar gameobject
- **FlatCalendar_ExampleDemo**: A demo scene of Flat Calendar
- **FlatCalendar.prefab**: Flat Calendar prefab object

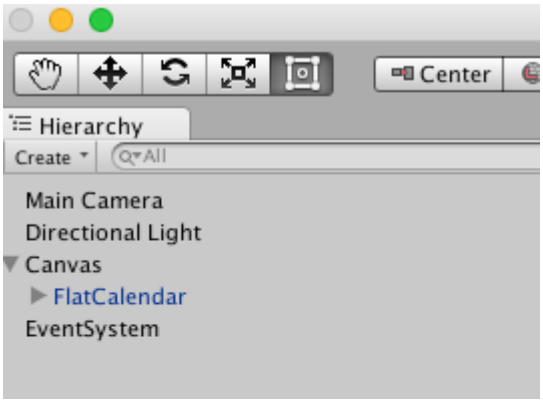
and these folders:

- **Ui/icons**: contains used icon images
- **Ui/fonts**: contains used text fonts

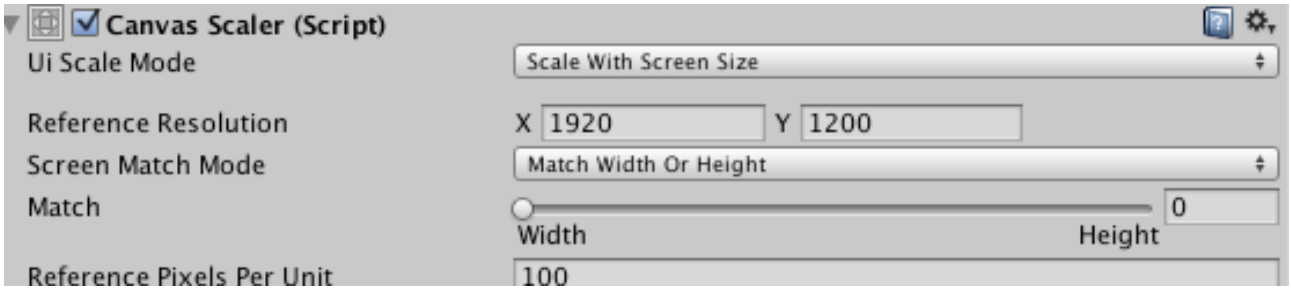
How to Use Flat Calendar

To start use Flat Calendar, you can load a demo scene or follow these step:

1) *Create a scene, add unity UI Canvas, add Flat Calendar Prefab to Canvas.*



2) *Select Canvas gameobject and set "Canvas Scaler" properties to "Scale with Screen Size" on the inspector side, and set resolution for your target device. (In my case 1920x1200 for Nexus 7 Device)*



3) Define Flat Calendar Object

```
// Declare FlatCalendar
FlatCalendar flatCalendar;
```

4) Get Flat Calendar Component

```
// Get Flat Calendar Instance
flatCalendar = GameObject.Find("FlatCalendar").GetComponent<FlatCalendar>();
```

5) Initialize Flat Calendar

```
// Initialize Flat Calendar
flatCalendar.initFlatCalendar();
```

6) Optionally, install demo data

```
// Install Demo Event List
flatCalendar.installDemoData();
```

7) Define your function for callbacks event

```
public void dayUpdated(FlatCalendar.TimeObj time)
{
    Debug.Log("Day has changed");
    time.print();
}

public void monthUpdated(FlatCalendar.TimeObj time)
{
    Debug.Log("Month has changed");
    time.print();
}

public void eventsDiscovered(FlatCalendar.TimeObj time, List<FlatCalendar.EventObj> list)
{
    Debug.Log("You have selected a day with: "+list.Count+ "events");
    for(int i = 0; i < list.Count; i++)
        Debug.Log("Event: " + i + " ==> " + "Name: " + list[i].name + " Description: " + list[i].description);
}

public void backHome(FlatCalendar.TimeObj time)
{
    Debug.Log("You have come back at home");
    time.print();
}
```

8) Register your function to Flat Calendar callbacks event

```
// Add Events Callbacks
flatCalendar.setCallback_OnDaySelected(dayUpdated);
flatCalendar.setCallback_OnMonthChanged(monthUpdated);
flatCalendar.setCallback_OnEventSelected(eventsDiscovered);
flatCalendar.setCallback_OnNowday(backHome);
```

9) Set your favorite UI Style

```
// Set UI Style
flatCalendar.setUIStyle(1);
```

Flat Calendar Scripts

FlatCalendar/FlatCalendar.cs

This script control the logic of Flat Calendar: navigation between month and year, day selection, callbacks management.

FlatCalendar/FlatCalendarStyle.cs

This script is used for Theme management.

Editor/FlatCalendarInspector.cs

This script override the default inspector view of "Calendar" gameobject

How to Create and Add Events

Flat Calendar define an EventObj structure for store a single event.

```
/**
 * Event Structure Object
 */
public struct EventObj
{
    public string name;
    public string description;

    public EventObj(string _name, string _description)
    {
        name        = _name;
        description = _description;
    }
}
```

string **name** : Is the name of event
string **description**: Is the description of event

Flat Calendar map all event in a specific structure named "events_list":

```
/**
 * Event List
 */
public static Dictionary<int,Dictionary<int,Dictionary<int,List<EventObj>>>> events_list;
```

this structure is composed by a series of HashMap which defines Year,Month, Day and a List of Event of a specific day.

For example:

```
List<EventObj> events = events_list[year][month][day]
```

return a list of all event for a specific year,month and day.

Flat Calendar provide to developer user a series of utility function for manage all calendar events:

Add a specific Event

```
public void addEvent(int year, int month, int day, EventObj ev)
```

This call add an event for a specific date

Remove a specific Event

```
public void removeEvent(int year, int month, int day, EventObj ev)
```

This call remove an event of a specific date

Remove all event of specific date

```
public void removeAllEventOfDay(int year, int month, int day)
```

This call remove all events of a specific date

Remove all event in the calendar

```
public void removeAllCalendarEvents()
```

This call remove all events in the calendar

Get all list of event of a specific date

```
public static List<EventObj> getEventList(int year, int month, int day)
```

This call get a list of all event stored in a specific date

How to Use Callbacks

Flat Calendar provide 4 callbacks in order to notify the user when an event in the user interface occour.

In order to use callback, you need to add your function as parameter in one of three setCallback function of Flat Calendar:

```
// ============================================================
// ================ SET DELEGATE CALLBACKS ================
// ============================================================

public void setCallback_OnDaySelected(Delegate_OnDaySelected func)
{
    delegate_ondayselected = func;
}

public void setCallback_OnEventSelected(Delegate_OnEventSelected func)
{
    delegate_oneventselected = func;
}

public void setCallback_OnMonthChanged(Delegate_OnMonthChanged func)
{
    delegate_onmonthchanged = func;
}

public void setCallback_OnNowday(Delegate_OnNowDay func)
{
    delegate_onnowday = func;
}
```

setCallback_OnDaySelected(Your Function): Call your function when user select a day in calendar

setCallback_OnEventSelected(Your Function): Call your function when user select a day that contain events

setCallback_OnMonthChanged(Your Function): Call your function when user change month in calendar

setCallback_OnNowDay(Your Function): Call your function when user tap on the home button

How to Select a Theme Style

Flat Calendar provide out of the box 4 types of theme.

- Green Sea
- Orange Juice
- Red Carpet
- Google Material

Themes are defiined in "FlatCalendarStyle.cs":

```
/*
 * UI Colors Configuration
 */
public enum COLORS_TYPE { GREEN_SEA, RED_CARPET, ORANGE_JUICE, GOOGLE_MATERIAL};
```

You can switch between theme using a combobox placed in the Inspector, or call in your script function:

```
// Apply UI Color style
FlatCalendarStyle.changeUIStyle(numberOfStyle);
```

where "numberOfStyle" is a position (zero based) of style in "COLORS_TYPE" enumaration.

How to Create Your Own Theme Style

In order to create a new style, you need these steps:

*1) In **"FlatCalendarStyle.cs"** Add new type style in **"COLORS_TYPE"** enumaration.*

*2) In **"FlatCalendarStyle.cs"** Add your colour rules for your new style.*

Example of style rules from "Green Sea" Theme:

```
// ================================================
// ================== GREEN SEA ===================
// ================================================
if(style == (int) FlatCalendarStyle.COLORS_TYPE.GREEN_SEA)
{
    color_header                = new Color(  0.0f/255.0f,112.0f/255.0f,113.0f/255.0f,255.0f/255.0f);
    color_subheader             = new Color(255.0f/255.0f,255.0f/255.0f,255.0f/255.0f,100.0f/255.0f);
    color_body                  = new Color(  0.0f/255.0f,125.0f/255.0f,126.0f/255.0f,255.0f/255.0f);
    color_footer                = new Color( 67.0f/255.0f, 77.0f/255.0f, 87.0f/255.0f,255.0f/255.0f);
    color_dayTextNormal         = new Color(255.0f/255.0f,255.0f/255.0f,255.0f/255.0f,255.0f/255.0f);
    color_dayTextEvent          = new Color(  0.0f/255.0f,112.0f/255.0f,113.0f/255.0f,255.0f/255.0f);
    color_bubbleEvent           = new Color(255.0f/255.0f,255.0f/255.0f,255.0f/255.0f,255.0f/255.0f);
    color_bubbleSelectionMarker = new Color(255.0f/255.0f,255.0f/255.0f,255.0f/255.0f,255.0f/255.0f);
    color_numberEvent           = new Color(238.0f/255.0f,105.0f/255.0f,105.0f/255.0f,255.0f/255.0f);
    color_year                  = new Color(255.0f/255.0f,255.0f/255.0f,255.0f/255.0f,255.0f/255.0f);
    color_month                 = new Color(255.0f/255.0f,255.0f/255.0f,255.0f/255.0f,255.0f/255.0f);
    color_day                   = new Color(255.0f/255.0f,255.0f/255.0f,255.0f/255.0f,255.0f/255.0f);;
    color_dayOfWeek             = new Color(255.0f/255.0f,255.0f/255.0f,255.0f/255.0f,255.0f/255.0f);
    color_Events                = new Color(255.0f/255.0f,255.0f/255.0f,255.0f/255.0f,255.0f/255.0f);
    color_ButtonRight           = new Color(255.0f/255.0f,255.0f/255.0f,255.0f/255.0f,255.0f/255.0f);
    color_ButtonLeft            = new Color(255.0f/255.0f,255.0f/255.0f,255.0f/255.0f,255.0f/255.0f);
    color_Home                  = new Color(255.0f/255.0f,255.0f/255.0f,255.0f/255.0f,255.0f/255.0f);
}
```

Credits

Author: Gerardo Ritacco

email: gerardo.ritacco@3dresearch.it

Company: 3DResearchsrl

website: http://www.3dresearch.it