

راهنمای ساخت افزونه برای کانستراکت ۲

تنظیمات پلاگین

```
function GetPluginSettings() {  
    return {  
        "name":          "Sprite",  
        "id":            "Sprite",  
        "version":       "1.0",  
        "description":   "An animated object.",  
        "author":        "Scirra",  
        "help url":      "http://www.scirra.com",  
        "category":      "General",  
        "type":          "world",  
        "rotatable":     true,  
        "flags":          pf_animations |  
                          pf_position_aces |  
                          pf_size_aces |  
                          pf_angle_aces |  
                          pf_appearance_aces  
    }  
}
```

شما باید در فایل edittime.js تابعی با نام GetPluginSettings بسازید که یک شیء با خصیصه‌های زیر را برگرداند:

name: اسم پلاگین، اگر خواستید بعداً می‌توانید تغییرش دهید؛ این کار هیچ مشکلی برای بازی‌هایی که با اسم قدیمی پلاگین ساخته شده اند ایجاد نمی‌کند. این اسمی است که در ویرایشگر کانستراکت ۲ به کاربران نمایش داده می‌شود.

id: اسم اصلی پلاگین که باید منحصر به فرد باشد و اصلاً تغییر نکند، در صورت تغییر آیدی، کانستراکت ۲ پلاگین مورد نظر را به عنوان یک پلاگین جدید می‌شناسد.

version: نسخه‌ی پلاگین را در قالب X.Y مشخص می‌کند. در صورتی که سورسی با ورژن جدید یک پلاگین ساخته شده باشد و کاربری نسخه‌ی قدیمی آن را داشته باشد، کانستراکت ۲ به کاربر هشدار می‌دهد.

description: توضیحی کوتاه که کار اصلی پلاگین را به صورت خلاصه بیان می‌کند. این توضیح، در هنگام انتخاب پلاگین در پنجره‌ی Insert New Object به نمایش در می‌آید.

author: نام شما و یا شرکت شما

"help url": اگر کاربر برای پلاگین شما روی help کلیک کند به لینکی که در این قسمت نوشته اید منتقل می‌شود.

category: در ویرایشگر کانستراکت تمام افزونه‌ها دسته‌بندی شده‌اند. این مورد می‌گوید که افزونه‌ی مورد نظر ما در کدام دسته قرار می‌گیرد. می‌توانید افزونه را در یکی از دسته‌بندی‌های موجود قرار دهید یا خودتان دسته‌بندی دلخواهی را بنویسید. نام دسته‌بندی‌ها به کوچکی و بزرگی حروف انگلیسی حساس است.

type (برای رفتارها استفاده نمی‌شود): این مورد بر اساس نوع پلاگینی که دارید می‌سازید می‌تواند یکی از مقادیر زیر را بپذیرد:

"world": پلاگین در لیوت نمایش داده می‌شود، بنابراین چیزی مثل تصویر یا متن را در صفحه نمایش می‌دهد.
"object": پلاگین در لیوت نمایش داده نمی‌شود و این پلاگین چیزی را در صفحه نمایش نمی‌دهد (مثل شیء آرایه). کاربر نمی‌تواند آن را در لیوت قرار دهد و فقط از طریق نوار پروژه و نوار اشیاء قابل دسترسی است.

rotatable (برای رفتارها استفاده نمی‌شود): اگر type روی "object" تنظیم شده باشد این تنظیم نادیده گرفته می‌شود. اگر type روی "world" تنظیم شده باشد کاربر می‌تواند زاویه‌ی پلاگین را تغییر داده و آن را بچرخاند، مثلاً چون اسپریت قابل چرخش است مقدار این تنظیم در آن true می‌باشد ولی برای شیء متن false است.

dependency: این یکی در نمونه کد بالا استفاده نشده است. ولی در مواقعی که می‌خواهید فایل‌هایی اضافی به افزونه‌ی خود اضافه کنید باید آن‌ها را توسط این خصیصه معرفی کنید تا بعد از خروجی گرفتن، آن فایل‌ها هم ذخیره شوند و مشکلی در عملکرد افزونه‌تان نداشته باشید. این فایل‌ها را می‌توانید با این تنظیم معرفی کنید مثلاً بنویسید:

```
"dependency": "file1.js;file2.js;file3.html"
```

شما باید این فایل‌ها را در کنار سایر فایل‌های افزونه‌تان قرار دهید.

"cordova-plugins": در این مورد حتی در خود راهنمای سایت سیرا به صورت رسمی حرفی به میان نیامده است! ولی اگر افزونه‌ی شما به یکی از پلاگین‌های Cordova وابسته است و از آن استفاده می‌کند بهتر است نام پلاگین را در قالب این خصیصه بنویسید. با این کار کانستراکت در هنگام خروجی Cordova نام پلاگین را به فایل config.xml اضافه می‌کند؛ در نتیجه Cocoon.io یا Intel XDK در هنگام خروجی می‌توانند پلاگین‌های به کار رفته در پروژه را تشخیص دهند و به صورت خودکار آن‌ها را نصب کنند. در غیر این صورت باید بعد از خروجی بازی پلاگین هم به صورت دستی نصب شود.

flags: این مورد برای انجام تنظیمات اضافی (فلگ‌ها) به کار می‌رود. فلگ‌ها را می‌توانید با عملگر «|» با هم ترکیب کنید (مثلاً pf_position_aces | pf_size_aces)، یا این‌که مقدار flags را برابر 0 قرار دهید تا از هیچ فلگی استفاده نکنید. فلگ‌ها متغیرهایی هستند که به صورت پیش‌فرض توسط کانستراکت تعریف شده‌اند، عبارت pf در شروع اسم فلگ‌ها مخفف plugin flag است. این این فلگ‌ها به شرح زیر هستند:

pf_singleglobal: باعث می‌شود که بیشتر از یک اینستنس از پلاگین به بازی اضافه نشود. بیشتر برای پلاگین‌های ورودی مثل موس و کیبورد و... کاربرد دارد. این فلگ همراه با تنظیم type روی "world" به کار نمی‌رود.

pf_texture: برای حالتی است که پلاگین از یک بافت (تکسچر) استفاده می‌کند، مثلاً شیء Tiled Background از این فلگ استفاده می‌کند. با اضافه کردن این فلگ، هنگام انتخاب پلاگین، پنجره‌ی ویرایشگر تصویر باز می‌شود.

pf_animations: پلاگین از سیستم انیمیشن کانستراکت ۲ استفاده می‌کند. مثلاً شیء اسپریت از این فلگ استفاده می‌کند. با اضافه کردن این فلگ هنگام انتخاب پلاگین، پنجره‌ی ویرایشگر انیمیشن باز می‌شود.

pf_tiling: این فلگ بدون به کار بردن یکی از فلگ‌های pf_texture یا pf_animations بی‌معنی است. این فلگ این امکان را می‌دهد تا تصویر وارد شده را به صورت کاشی‌کاری (تایل) بچینید. مثلاً شیء Tiled Background از این فلگ استفاده می‌کند.

pf_position_aces: این فلگ فقط همراه با تنظیم type بر روی "world" به کار می‌رود. با اضافه کردن این فلگ به صورت خودکار اکشن‌ها، کنديشن‌ها و اکسپرشن‌های مربوط به مکان شیء به پلاگین اضافه می‌شوند (مثل Set X و Set Y).

pf_size_aces: این فلگ فقط همراه با تنظیم type بر روی "world" به کار می‌رود. با اضافه کردن این فلگ به صورت خودکار اکشن‌ها، کنديشن‌ها و اکسپرشن‌های مربوط به اندازه‌ی شیء به پلاگین اضافه می‌شوند (مثل Set Width و Set Height).

pf_appearance_aces: این فلگ فقط همراه با تنظیم type بر روی "world" به کار می‌رود. با اضافه کردن این فلگ به صورت خودکار اکشن‌ها، کنديشن‌ها و اکسپرشن‌های مربوط به ظاهر شیء به پلاگین اضافه می‌شوند (مثل Set Set Opacity و Visible).

pf_zorder_aces: این فلگ فقط همراه با تنظیم type بر روی "world" به کار می‌رود. با اضافه کردن این فلگ به صورت خودکار اکشن‌ها، کنديشن‌ها و اکسپرشن‌های مربوط به Z Order شیء به پلاگین اضافه می‌شوند (مثل Set Layer و Move To Front).

bf_onlyone: این فلگ فقط برای رفتارها به کار می‌رود و باعث می‌شود که رفتار فقط یکبار بتواند به شیء اضافه شود. در حالت عادی کاربر می‌تواند رفتارها را هر چند بار که دلش بخواهد اضافه کند، اما این فلگ باعث می‌شود بعد از این‌که یکبار رفتار اضافه شد مجدداً نشود آن را اضافه کرد. مثلاً رفتار جامد (Solid) از این فلگ استفاده می‌کند، چون دو بار رفتار جامد دادن به یک شیء تأثیری در جامد بودنش ندارد و فقط یکی کافیست.

تغییرات پس از انتشار افزونه

اگر بعد از انتشار افزونه خواستید برای نسخه‌ی جدید آن را ویرایش کرده و تغییر دهید، هیچ‌وقت مقدار id، type، rotatable یا flags را دست نزنید، چون با این کار تمام پروژه‌هایی که از نسخه‌ی قبلی پلاگین شما استفاده می‌کردند با نصب نسخه‌ی جدید خراب می‌شوند. ولی بقیه‌ی تنظیمات هر زمان می‌توانند عوض شوند.

مترجم: مجتبی قاسم‌زاده تهرانی

مؤسس مرجع تخصصی کانستراکت

Construct2.ir