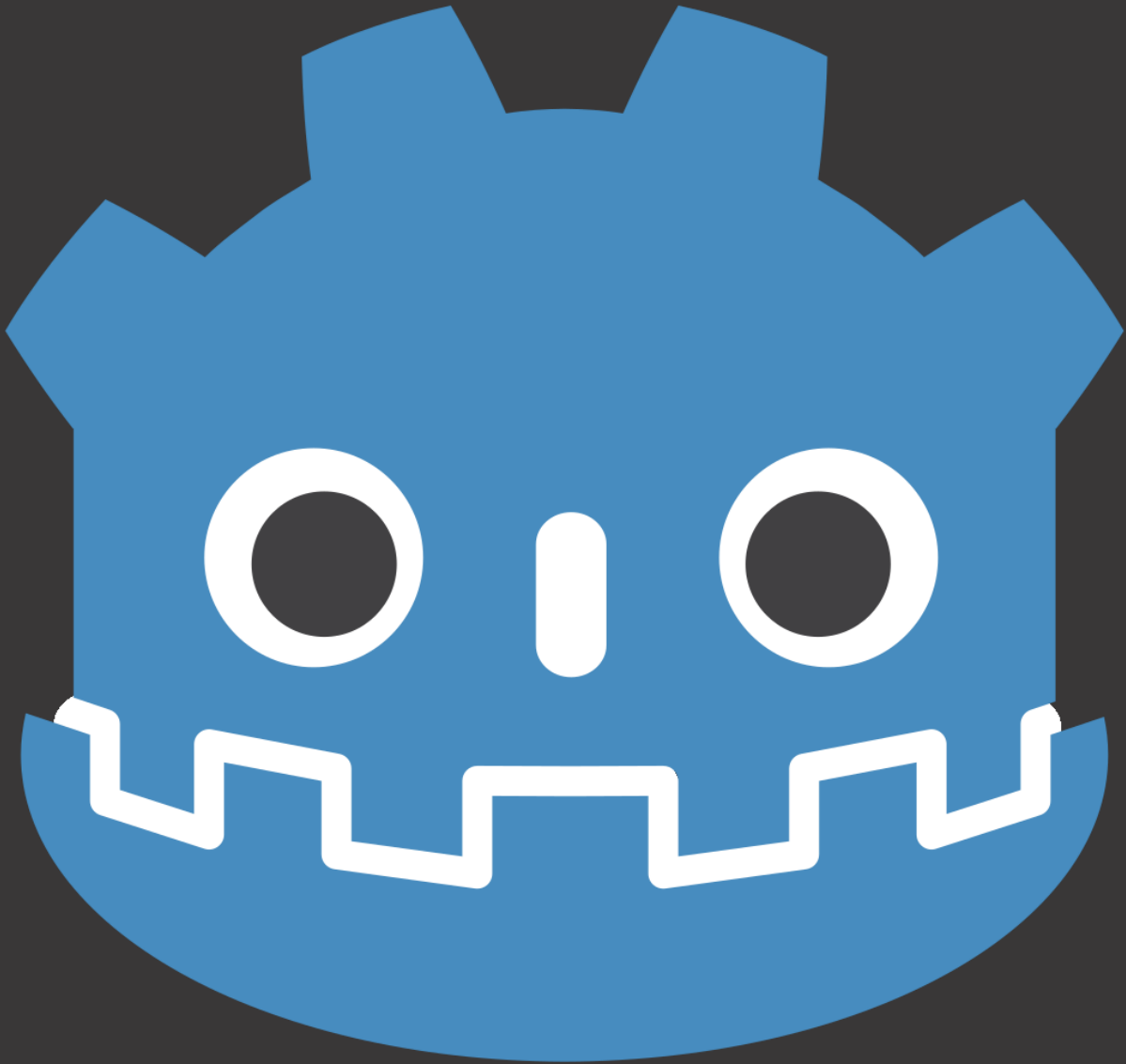


به نام خدا



آموزش گادوت انجین

مقدماتی

مقدمه

اکثر افراد عاشق بازی کردن هستند، آنها بازی می کنند و بازی می کنند. اما عده ای دیگر هم هستند که رویا های خود را دارند و به آنچه دیگران ساخته اند قانع نمی شوند. آنها می خواهند که بازی مخصوص خودشان را داشته باشند و اینطور می شود که آنها بازی ساز می شوند. اگر شما هم از آن افراد هستید با ما همراه باشید تا چگونگی ساخت بازی با موتور قدرتمند و رایگان گادوت انجین (به انگلیسی: *Godot engine*) را یاد بگیرید. پس با ما همراه باشید...

سر فصل ها

1. گادوت انجین چیست؟
2. دانلود و نصب
3. آشنایی با محیط گادوت انجین
4. یادگیری چگونگی ساخت بازی با گادوت انجین
5. یادگیری GDscript
6. ساخت یک بازی پلتفرمر!

گادوت انجین چیست؟

گادوت انجین یک موتور بازی سازی کاملا رایگان و متن باز است که به ما امکان ساخت بازی های دو بعدی و سه بعدی در سبک های مختلف را می دهد.

زبان برنامه نویسی (بازی نویسی) این موتور GDscript نام دارد (که خیلی شبیه به زبان پایتون است)، البته در این موتور با زبان های دیگری چون C# نیز می توان بازی ساخت. در این موتور شما می توانید هم بازی دو بعدی بسازید و هم بازی سه بعدی. البته این موتور برای ساخت بازی های دو بعدی بسیار بهتر است (حتی از یونیتی هم در ساخت بازی های دو بعدی بهتر عمل می کند).

این موتور برخلاف دیگر موتور های بازی سازی که خیلی از آنها را

می شناسیم (مانند: یونیتی، گیم میکر استودیو، کنستراکت) بسیار کم حجم است و شما می



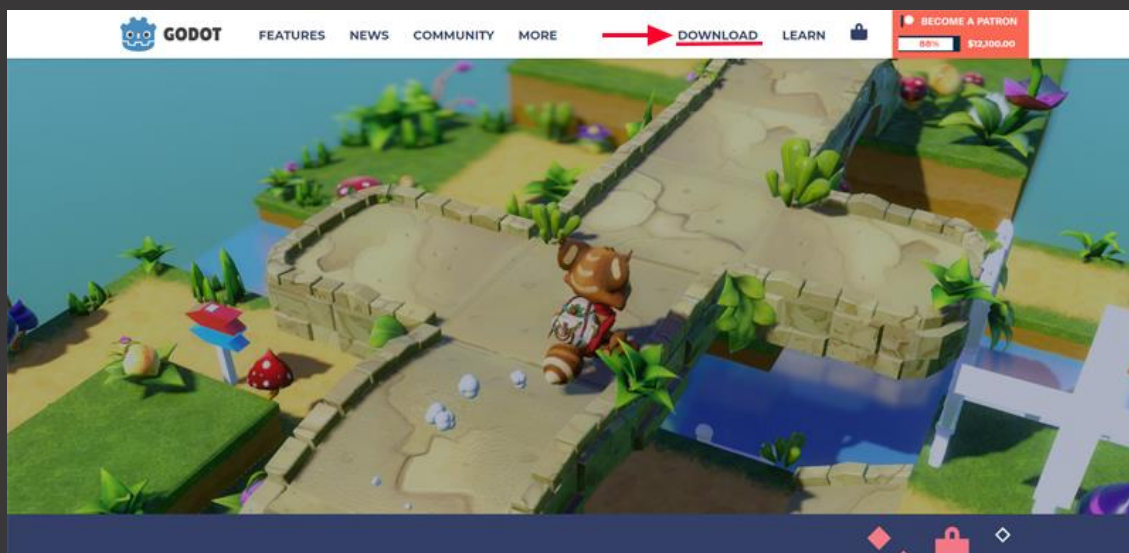
توانید با دانلود حجمی کمتر از 50 مگ با این موتور بازی بسازید و آن را آزمایش کنید. فقط برای خروجی گرفتن به صورت فایل اجرایی نیاز به دانلود یک فایل پر حجم دارید.

دانلود و نصب

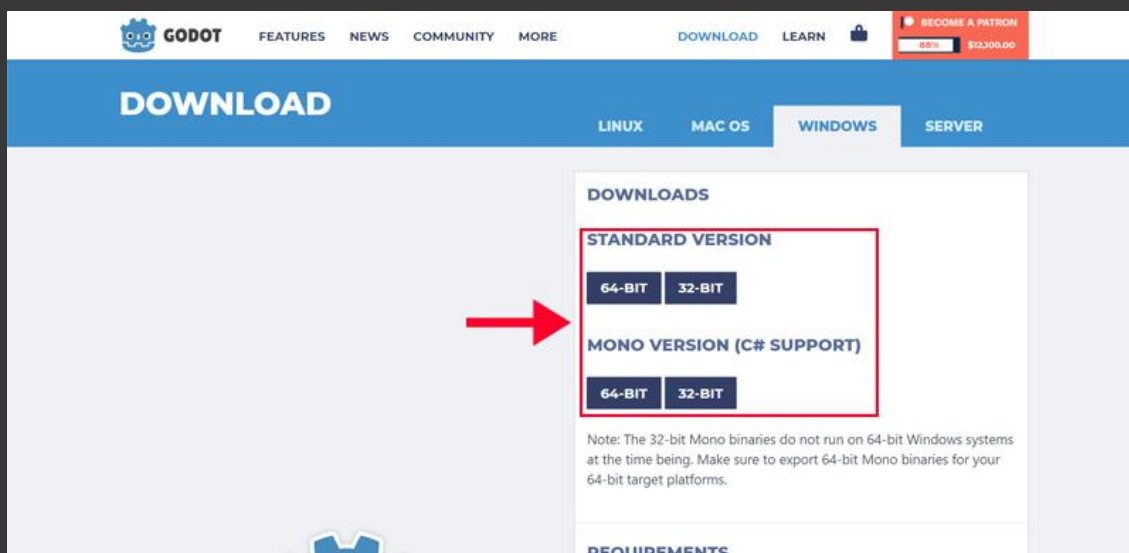
این موتور کاملاً رایگان است و شما بدون هیچ نگرانی و محدودیتی می‌توانید آن را از سایت سازنده اش دانلود کنید:

<https://godotengine.org>

برای دانلود مراحل زیر را دنبال کنید:



و حالا نسخه مورد نظر را دانلود کنید (نسخه مونو سی شارپ را پشتیبانی می‌کند):

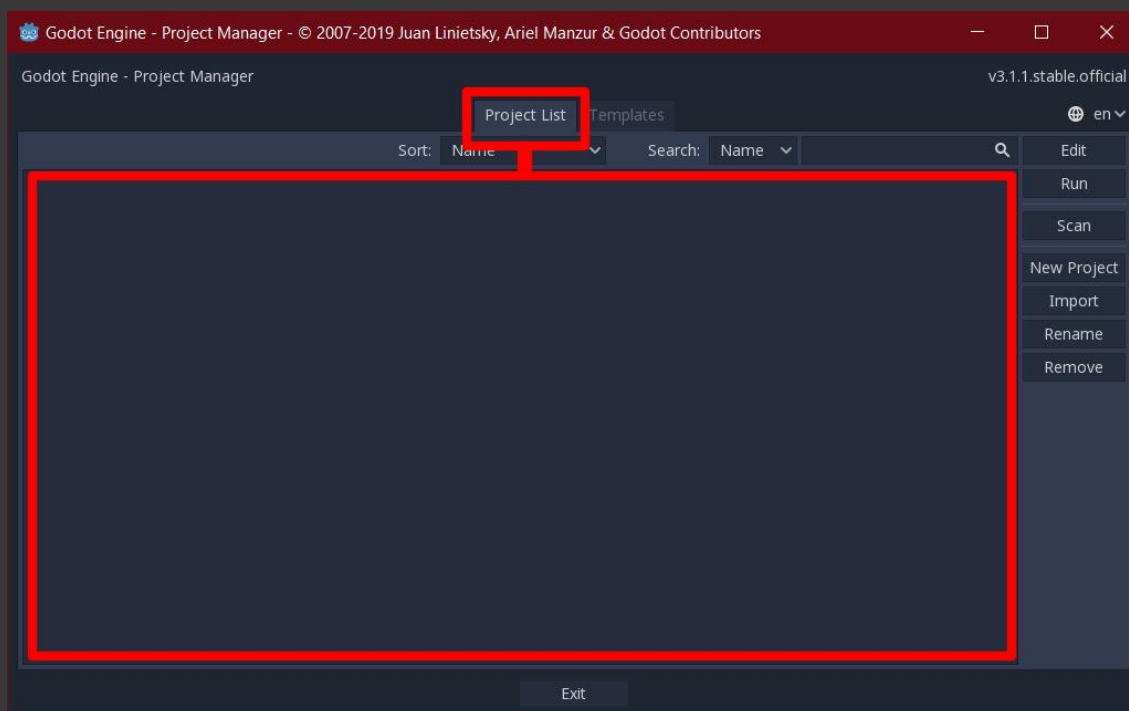


این موتور هیچ نیازی به نصب ندارد و شما بعد از دانلود کافیهست که آن را از حالت فشرده خارج و اجرا کنید.

آشنایی با محیط گادوت انجین

قبل از اینکه شروع به استفاده از یک نرم افزار کنید، ابتدا باید با محیط آن آشنا باشید. در این قسمت شما با محیط گادوت انجین آشنا می شود.

وقتی گادوت انجین را اجرا می کنید با صفحه مدیریت پروژه (به انگلیسی: *Project manager*) رو به رو می شوید. در این صفحه شما می توانید پروژه ایجاد کنید، پروژه ها را اجرا کنید، پروژه وارد کنید و همچنین می توانید پروژه های آماده را دانلود و مشاهده کنید. در قسمت مشخص شده پروژه ها شما قرار دارد:



Edit: باز کردن و ویرایش کردن سورس پروژه انتخاب شده.

Run: اجرای پروژه انتخاب شده (بازی اجرا می شود و نمی توان آن را ویرایش کرد).

Scan: اسکن کردن محلی برای پیدا کردن پروژه.

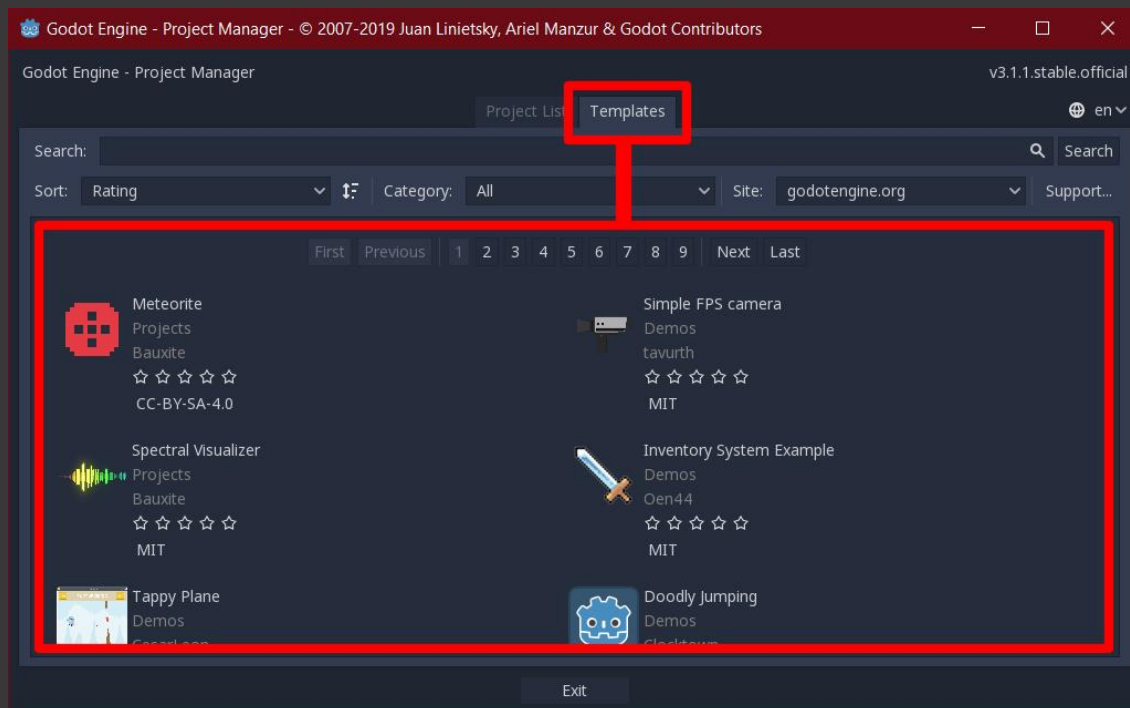
New Project: ایجاد یک پروژه جدید.

Import: وارد کردن پروژه.

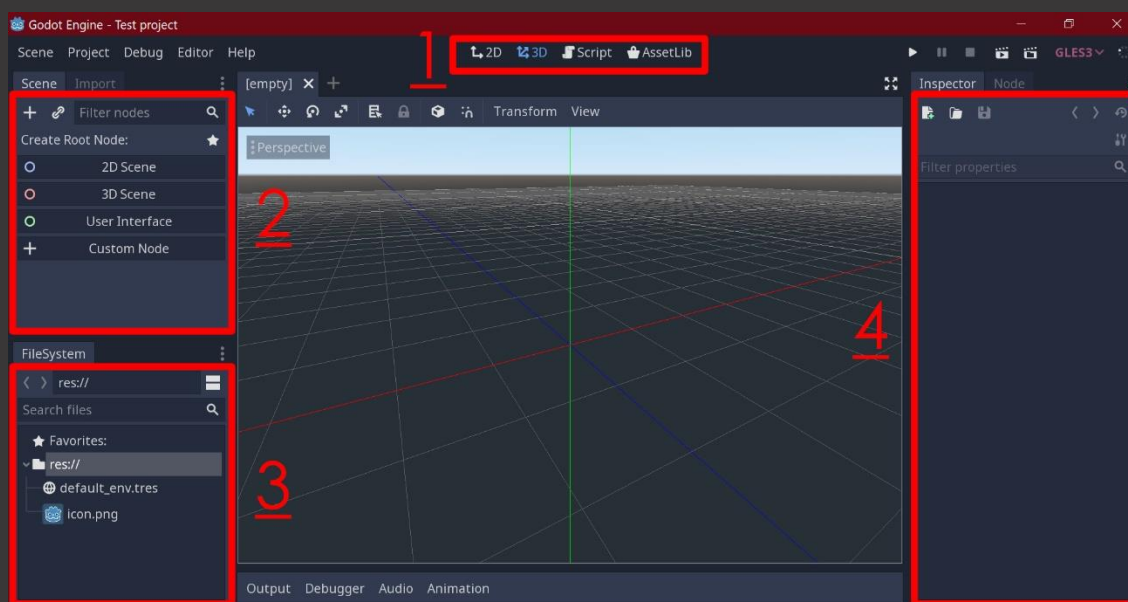
Rename: تغییر نام پروژه انتخاب شده.

Remove: حذف پروژه انتخاب شده.

در این قسمت هم پروژه های آماده به صورت متن باز قرار داده شده اند (برای یادگیری هستند):



و اما مهم ترین بخش.
مهم ترین بخش هر موتور بازی سازی ادیتور آن است. شما در ادیتور بازی خود را می سازید.
ادیتور گادوت انجین:



1. انتخاب صحنه:

در این بخش شما می توانید محیط مورد استفاده خود را انتخاب کنید.

2D: وارد محیط دو بعدی می شوید (برای ساخت بازی دو بعدی).

3D: وارد محیط سه بعدی می شوید (برای ساخت بازی سه بعدی).

Script: وارد کد ادیتور می شوید (برای برنامه نویسی بازی).

AssetLib: وارد فروشگاه پروژه های آماده می شوید.

2. نود ها:

نود ها مهم ترین بخش گادوت انجین هستند. شما با استفاده از نود ها بازی خود را می سازید. در فصل بعد با نود ها بیشتر آشنا می شویم.

3. مدیریت فایل ها:

در این بخش شما باید تصاویر و صدا های مورد نیاز بازی خود را قرار می دهید تا بتوانید از آنها در بازی استفاده کنید.

4. بازرسی:

با انتخاب هر نود می توانید ویژگی هایش را در این بخش ویرایش کنید.

یادگیری چگونگی ساخت بازی با گادوت انجین

ساخت بازی در گادوت انجین به لطف نود ها (به انگلیسی: *Node*) انجام می شود.

در گادوت انجین تعداد بسیار زیادی نود وجود دارد که هر کدام از آنها برای کار خاصی ساخته شده اند.

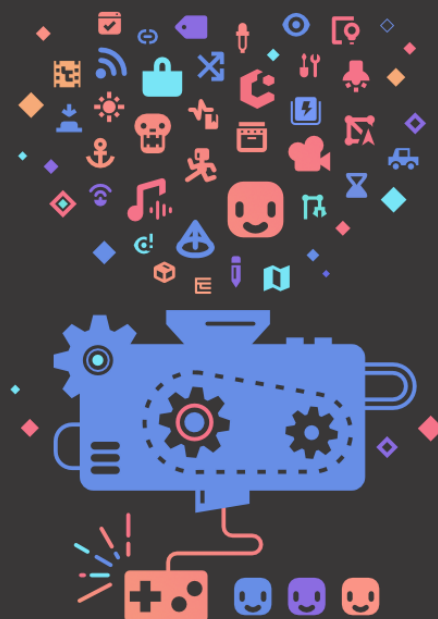
نود ها به دسته های مختلفی تقسیم شده اند و از آنها می توان به نود های زیر اشاره کرد:

Node2D: برای ساخت بازی دو بعدی.

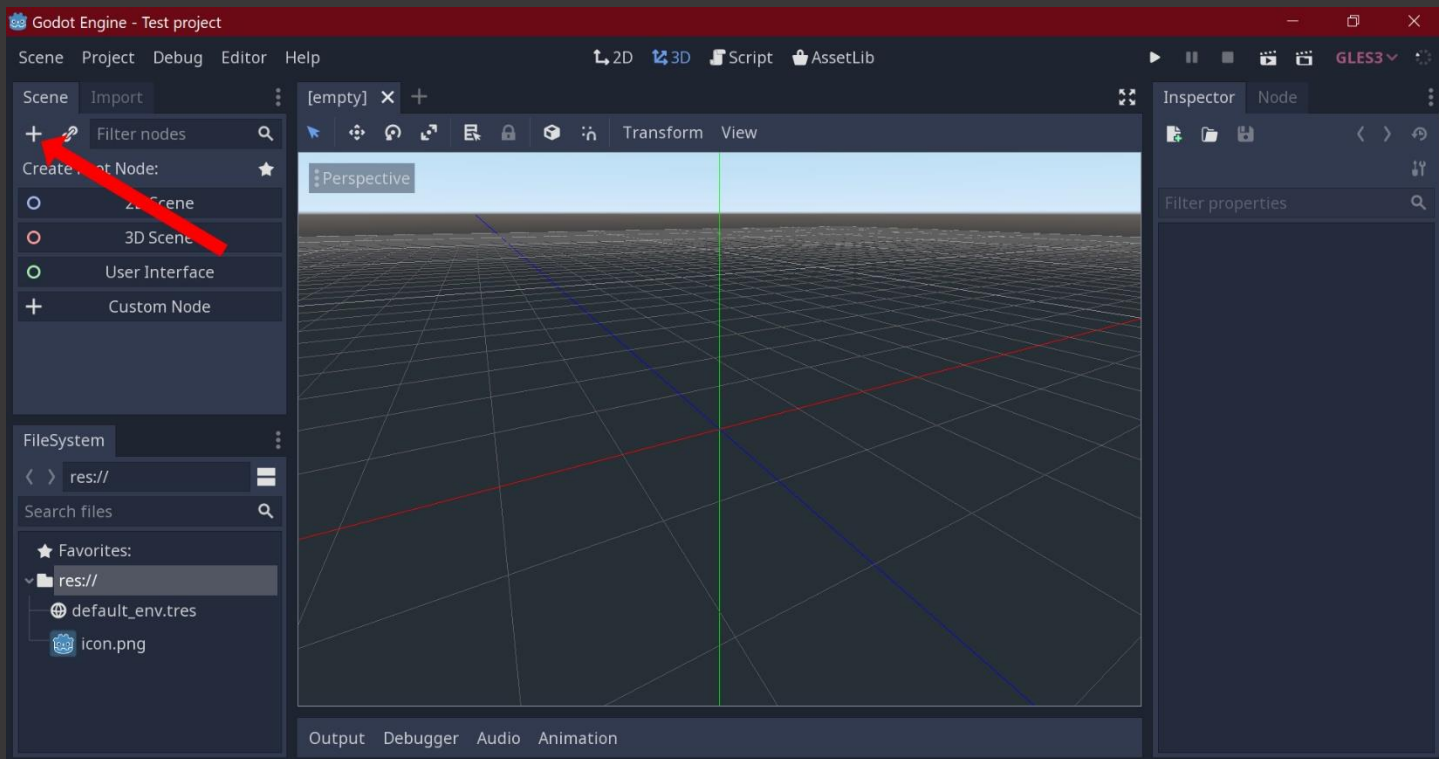
Spatial: برای ساخت بازی سه بعدی.

Control: برای ساخت کنترل ها.

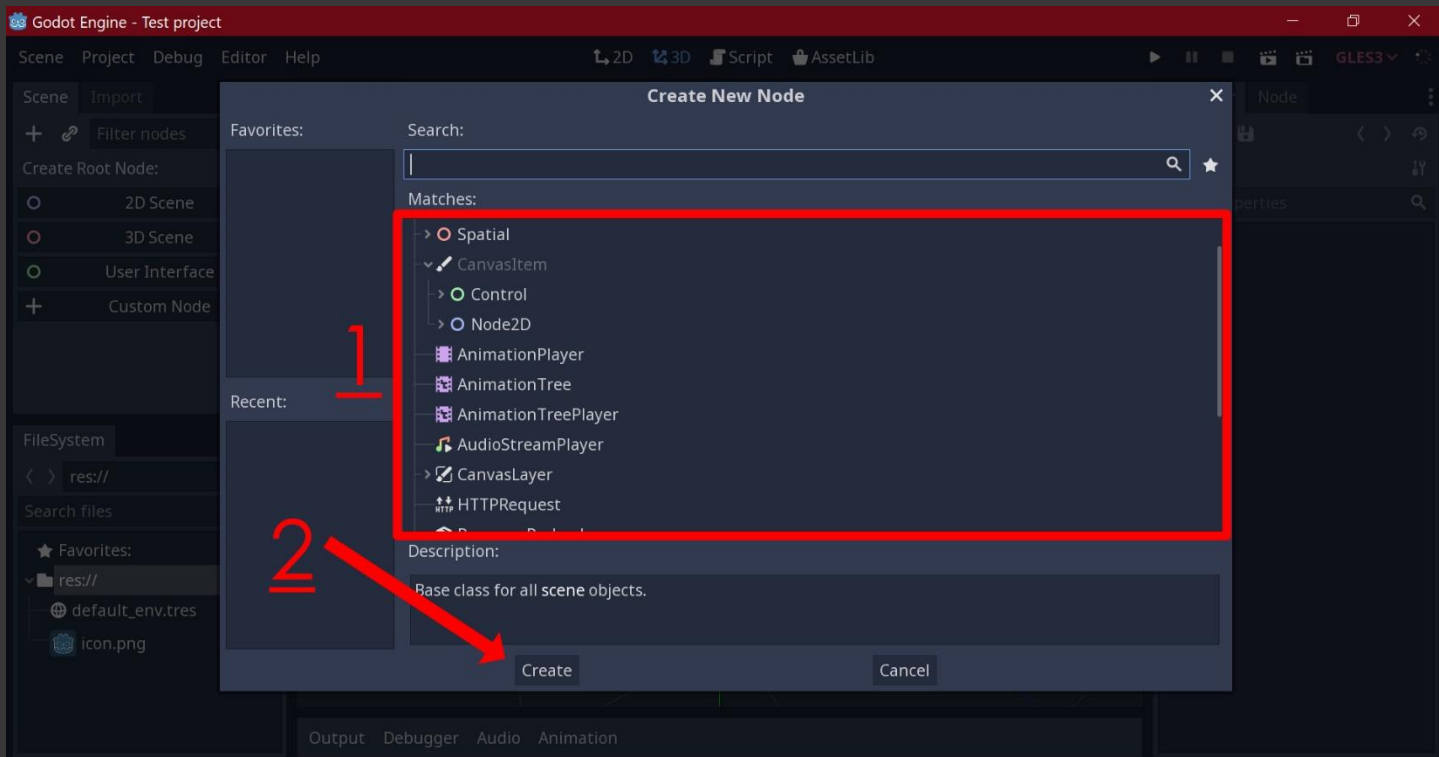
برای استفاده از نود ها تنها کافیست که پس از ایجاد پروژه مراحل زیر را طی کنید:



1. در بخش نود ها بر روی بعلاوه کلیک کنید:



2. و حالا یک پنجره باز می شود و شما می توانید نود خود را انتخاب کنید:



خب تا اینجا شما تا حدودی با گادوت انجین آشنا شدید ولی هنوز برای ساخت یک بازی کافی نیست.

در ادامه با ما همراه باشید...

یادگیری GDscript

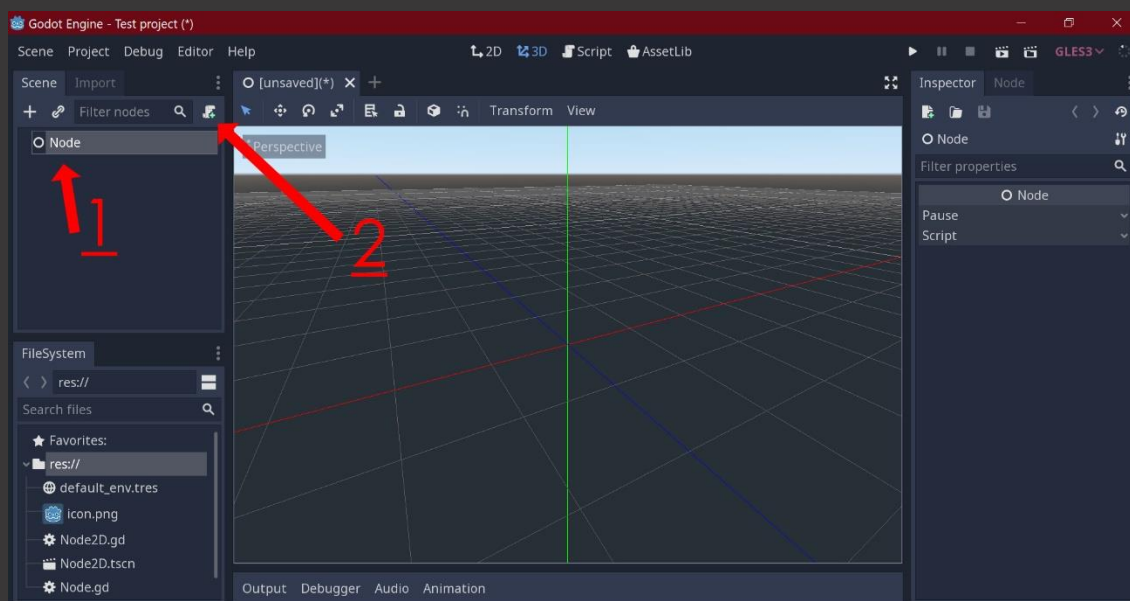
برنامه نویسی و بازی سازی به یکدیگر گره خورده اند و نمی شود بازی را بدون کد نویسی ساخت. البته موتور هایی هم هستند که بدون کد نویسی می توان با آن ها بازی ساخت اما محدودیت دارند (مانند: گیم میکر، گیم میکر استودیو). اگر بخواهید یک بازی کم نقص بسازید حتما به کد نویسی نیاز پیدا خواهید کرد.

زبان برنامه نویسی گادوت انجین GDscript نام دارد. این زبان شباهت بسیار زیادی به پایتون دارد و کسانی که پایتون بلدند به راحتی آن را متوجه می شوند. تنها تفاوت اساسی این دو زبان این است که GDscript مخصوص بازی سازی است. در این زبان توابع و متدهای خاصی وجود دارند که مخصوص گادوت انجین هستند.

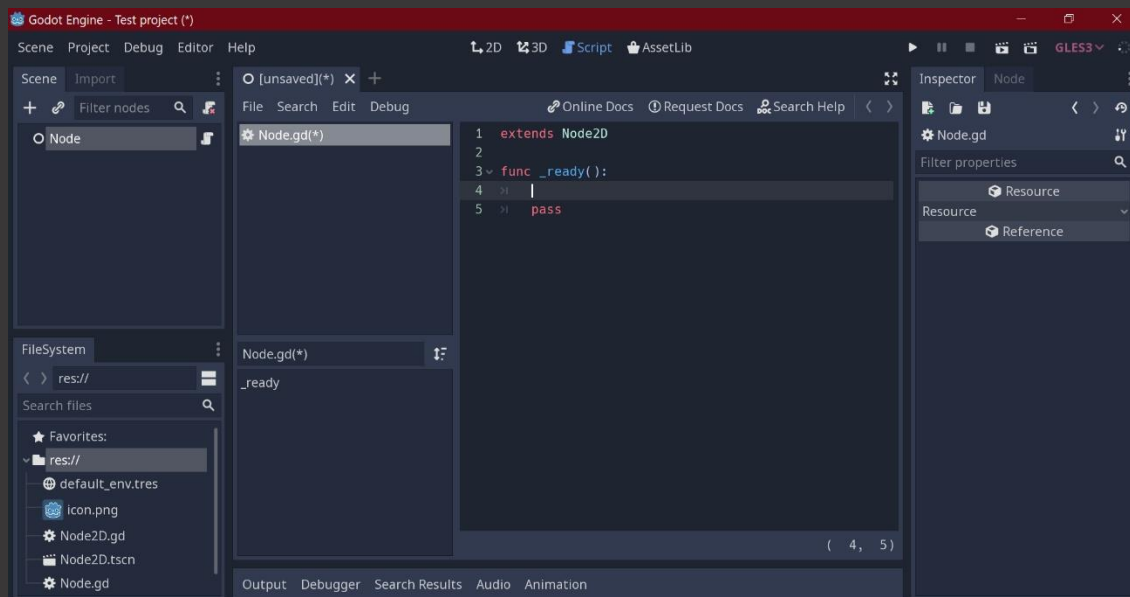
اول از همه باید به یکسری از مفاهیم اولیه در GDscript آشنا شویم.

ابتدا یک نود ایجاد کنید (حلقه سفید).

سپس طبق تصویر پیش بروید:



وقتی صفحه Script را باز می کنید با این صفحه مواجه می شوید. طبق تصویر تمام کد ها به غیر از کد هایی که در تصویر معلوم است را پاک کنید:



حالا باید متغیر ها را در خط 2 بنویسید (بعد از هر متغیر یک اینتر بزنید تا به خط بعد بیایید). و بقیه کد ها را در خط 4 با 4 تا اسپیس بنویسید (هر وقت فلش هست یعنی 4 تا اسپیس زده شده).

سینتکس:

بعد از دو نکته باید 4 اسپیس زد:

```
func hello():
```

```
----print("Hello")
```

توجه داشته باشید به جای _ باید اسپیس بزنید.

دستور چاپ:

برای چاپ یک عبارت (البته در بازی دیده نمی شود و تنها برای تست کد ها کاربرد دارد) از متد print استفاده می کنیم.

مثال:

```
print("Hello, world")
```

متغیر ها:

بسیاری از مواقع لازم است یک مقدار عدد یا یک عبارت را بار ها تایپ کنیم. برای جلوگیری از اشتباه از متغیر ها استفاده می شود.

مثال:

```
var x = 100
```

در کد بالا x نام متغیر است و 100 مقدار متغیر و اگر x را چاپ کنیم 100 نمایش داده می شود:

```
var x = 100
```

```
print(x)
```

ما انواع متغیرها را داریم:

```
var string = "hello"
```

```
var integer = 100
```

```
var list = ["egg", "milk", "chips"]
```

دستورات شرطی:

در خیلی از مواقع باید بگویید اگر فلان دکمه فشار داده شد پپر بالا و در غیر این صورت همان جا وایسا.

در این مواقع باید از دستورات شرطی استفاده کنید.

مثال:

```
if x == 100:
```

```
    print("The number is 100")
```

```
elif x < 100:
```

```
    print("The number is smaller than 100")
```

```
else:
```

```
    print("The number isn't 100")
```

در کد بالا اگر x برابر بود با 100 عبارت **The number is 100** چاپ می شود، اگر x کوچکتر از 100 بود عبارت **The number is smaller than 100** چاپ می شود و در غیر این صورت عبارت **The number isn't 100** چاپ می شود.

البته توجه کنید که بعد از دو نقطه که اینتر زدید و به خط بعد رفتید به طور خودکار 4 اسپیس زده می شود و وقتی می خواهید از شرط خارج شوید یک backspace بزنید کافی است.

توابع:

توابع وقتی کاربرد دارند که شما می خواهید چند کد را چندین بار بنویسید. برای این کار می توانید یک تابع تعریف کنید و کدهای خود را آنجا نوشته تا هر وقت تابع را صدا زدید کدهای داخل آن اجرا شود.

مثال:

```
func hello():  
    print("Hello")
```

در این کد نام تابع ما hello است و هر کدی که بعد از آن با 4 اسپیس نوشته شود جزء آن تابع است.

توجه کنید:

```
func hello():  
    print("Hello")  
    print("bye")
```

در کد بالا `print("Hello")` جزء تابع است اما `print("bye")` جزء تابع نیست. و برای صدا زدن تابع کافیست نام تابع را با دو پرانتز هر جایی که می خواهید بنویسید. مثال (برای صدا زدن تابع hello):

```
hello()
```

اکنون شما با مفاهیم کلی این زبان آشنا شدید ولی به هیچ وجه قادر به ساخت بازی نیستید. برای همین در فصل آخر این کتاب یک آموزش پروژه محور قرار دادم تا شما ساخت بازی پلتفرم را یاد بگیرید.

ساخت یک بازی پلتفرم!

خب تا اینجا خیلی چیزها را یاد گرفتید اما آیا می‌توانید یک بازی بسازید؟

خییر!!!

این چیزهایی که تا به حال یاد گرفتید فقط یک مقدمه بود و حتی برای ساخت بازی باید از کدهای دیگر استفاده کرد.
خب میرویم تا بازی مان را بسازیم.

متأسفانه نمی‌شود در این کتاب آموزش را قرار دهم خیلی پیچیده می‌شود پس این فیلم را برایتان گذاشته‌ام:

<https://xip.li/CqKusg>

این تصاویر را هم برایتان درست کرده‌ام تا در پروژه تان استفاده کنید:

<https://xip.li/p15G6b>

اگه مشکلی تو کتاب دیدی بهم خبر بده:

<http://upprom.blog.ir>

استودیو بازی سازی آپروم

😊 نوشته شده توسط محمد صالح کامیاب 😊

<http://upprom.blog.ir>

© این کتاب کاملا رایگان بوده و کپی برداری تنها به شرطی مجاز است که هیچ تغییری در این کتاب ایجاد نکنید.

با تشکر از:

متاسفانه نمی توانم آدرس ها رو اینجا بذارم تبلیغات میشه.

خدا حافظ...