



Best 2D Game Engines: The Complete List

by [Thomas Gervraud](#)

Looking for the best 2D game engine for your next project?

There are literally dozens of them on the market. It can get overwhelming really fast.

That's why I decided to put 44 of them all onto one place.

Here are the game engines you'll learn about.

1. [Unity 2D](#)
2. [GameMaker Studio 2](#)
3. [Construct 3](#)
4. [Godot Engine](#)
5. [Cocos2d](#)
6. [Phaser.io](#)
7. [Corona](#)
8. [RPG Maker](#)
9. [Unreal Engine](#)
10. [Buildbox](#)
11. [Clickteam Fusion](#)
12. [Defold](#)
13. [LÖVE](#)
14. [Heaps](#)
15. [GDevelop](#)
16. [GameSalad](#)
17. [MonoGame](#)
18. [Stencyl](#)
19. [Xenko](#)
20. [Adventure Game Studio](#)
21. [Oxygine](#)
22. [OpenFL](#)
23. [GamePlay3D](#)
24. [AppGameKit](#)
25. [BYOND](#)
26. [Starling](#)
27. [SFML](#)
28. [libGDX](#)
29. [pygame](#)
30. [PixiJS](#)
31. [Superpowers](#)
32. [Raylib](#)
33. [Arcade](#)
34. [Urho3D](#)
35. [Tilengine](#)
36. [Haxeflixel](#)
37. [SDL](#)
38. [Kivy](#)
39. [Duality2D](#)
40. [FNA](#)
41. [Amulet](#)
42. [Ren'Py](#)
43. [Orx](#)
44. [nCine](#)

Unity 2D

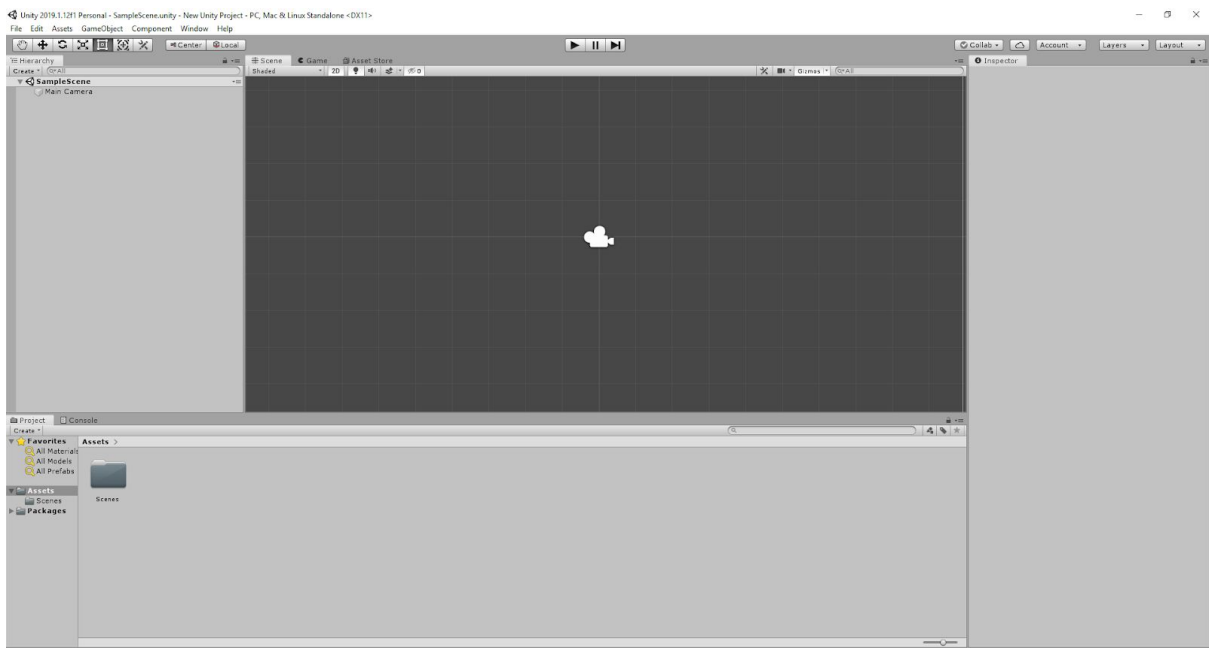


You've probably heard of this one already. It's the most popular game engine right now.

[50% of mobile games](#) are made with Unity. And it matters: it means it has way more documentation and tutorials available than others.

You will get help if you need it, whether it be [in the official forums](#) or just by watching one of the many videos by YouTubers such as [Brackeys](#) or [Blackthornprod](#).

As far as the software itself, you'll most likely need to know how to code in **C#**.



You could get by using one of the visual scripting plugins from the asset store like [PlayMaker](#). But I'd recommend looking at an easier engine if you're not willing to learn programming.

Not the easiest to start with, but totally doable. And you probably won't have to change engines down the line for your future games, even if you choose to go 3D.

Made with Unity

<https://www.youtube.com/watch?v=NMUr2yNALhU>

Price

You can use [Unity Personal](#) for free until you earn \$100K with your project. You'll have a Unity splashscreen in your game.

After that, you'll have to upgrade to [Unity Plus](#) (~25\$/month if prepaid annually or 35\$/month) until \$200K and [Unity Pro](#) (125\$/month) after that.

Unity supports all the platforms you most likely want to publish on.
(PC/Mac/Linux/Consoles/Mobile etc...)

[See the prices for yourself.](#)

Get Started

[Download Unity Hub here](#) and install Unity from here. It can take some time.

Then I'd recommend you [head over there](#) and complete that tiny 1h45 tutorial.

After that, work on a longer tutorial series [like this one from Brackeys](#).

By the time you finish it, you should be ready to publish your own game.

[GameMaker Studio 2](#)



This is the one I personally went with. It's designed for making 2D games fast and easily.

You can create games using either a **drag-n-drop** (DnD) interface or by programming with a specific language called **GameMaker Language** (GML).

You won't find this language elsewhere, but it's pretty similar to Java or C#. If you already know how to code it'll be easy for you to use.

The [documentation](#) is complete and helpful and there are a bunch of good YouTube channels such as [Shaun Spalding](#)'s or [HeartBeast](#) that make great tutorials.

There is a small asset store but it's not nearly as big as Unity's. **Great choice to begin with** if you're willing to put down a bit of cash upfront.

Made with GameMaker Studio 2

<https://www.youtube.com/watch?v=NeSfnKQTCWw>

Price

After a 30-day free trial, you'll have to pay \$39 a year for the Creator version. You'll be able to publish either on Windows or on Mac and your game will have a GameMaker splash screen.

It'll cost \$99 (one time fee) to get rid of the splash screen and to be able to publish on all three desktop platforms. Then it's more expensive to get onto all other platforms.

[See the prices for yourself.](#)

Get Started

[Download it here](#) and install it.

Then [head over here](#) and complete this first tutorial. You can do it either in DnD or in GML - I recommend the latter.

After that, work on [this tutorial series by Shaun Spalding](#).

When you're done, you'll be well equipped to make your own 2D games.

[Construct 3](#)



Construct 3 is designed to be incredibly beginner-friendly. You use an event system instead of coding to create the logic of your games.

You can install a bunch of [plugins](#) for specific tasks, and you can script in **JavaScript** if you want more control over what you're doing.

Construct 3 even **runs on tablets and mobiles.**

The engine is regularly updated and the community is pretty active, although you will still find a lot of Construct 2 users. [The documentation](#) is also well done.

It's a **good engine if you want to make simple games.** It's definitely sufficient to make [commercially viable games](#). But it might not be for you if you're trying to make the next Hollow Knight.

Made with Construct

<https://www.construct.net/en/make-games/showcase>

Price

You can try Construct 3 with a free version directly in your browser (with limitations).

After that, it'll cost you \$99 a year to get a personal license, \$149 a year if you have a small business or \$399 a year if your company makes more than \$50K in revenue.

You [can publish to most platforms](#) with the notable exception of the Switch and the PlayStation as they do not support HTML5.

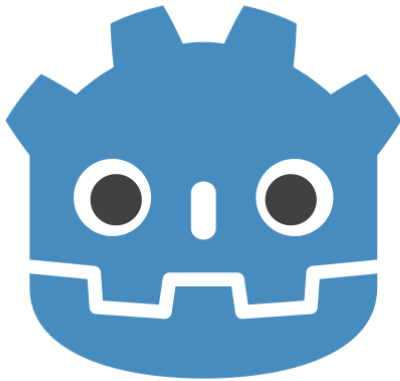
[See the prices for yourself.](#)

Get Started

[Go there](#) and launch it directly in your browser.

Then follow [the official beginner's guide to Construct 3](#). It'll send you to the right places once you finish it.

[Godot Engine](#)



GODOT

Game engine

It's free, open-source and lightweight, supports both 2D and 3D games and has been [praised as the next big thing](#) by [many people](#).

The project is [financed on Patreon](#) by the community.

Its interface might remind you a bit of Unity's. You will program using **GScript**, a language similar to Python. You can also use C++ or C# if you prefer.

You won't even need an account to use it. Just download a 20Mo zip file, extract and launch. No install needed.

It comes with a [full documentation](#) and a very [active community](#). **Recommended if you already know how to code.**

Made with Godot Engine

<https://www.youtube.com/watch?v=NIKEO1N8wMM>

Price

Totally free. Supports all platforms except consoles natively.

You [can use third party companies](#) to port your games to consoles.

Get Started

[Head over there](#) to download Godot and launch it.

Then follow the official [Getting Started](#) documentation or follow [this tutorial series by HeartBeast](#) directly if you prefer videos.

[Cocos2d](#)



This one is an [open source framework](#) that can be used in many different languages depending on which version you choose.

There is a unified package called Cocos Creator that includes the Cocos2d engine alongside an editor and other tools. You could also just use the engine directly but it's a bit less beginner friendly.

For Cocos Creator, you'll use **Javascript**. [The forums](#) are relatively active and [the documentation](#) is pretty clean.

You'll find way **less tutorials** on this engine than for the previous ones, but what does exist should be sufficient to complete a project.

You might want to look into it if you're interested in a free and well-optimised engine to make simple games.

Made with Cocos Creator

<https://discuss.cocos2d-x.org/c/game-demo-showcase>

Price

Totally free.

About the supported platforms, here's what [the official website](#) says:

“Currently, Cocos Creator can publish games to the Web, Android, iOS, desktops (Mac and windows) and Instant Games. The goal is to develop once and publish to all platforms.”

Get Started

[Head over there](#) to download and install it, then [go follow this tutorial](#) if you prefer video or [here](#) if you prefer text.

[Phaser](#)



Phaser is a free and open source game engine designed to create **browser based games**. It's developed by [Richard Davey](#) and financed on [Patreon](#).

The engine is written in **JavaScript** - and that's the language you'll be using to code your own games. You can also use **TypeScript**.

Do note that you'll need to install [a few auxiliary tools](#) and have your own server if you want to share your projects with the world. There's also an [unofficial Phaser editor](#) that you can buy if you want to have a scene editor and other tools (it's not required).

There's a relatively [active community](#). [The documentation](#) is complete (although a bit hard to navigate) and the [written tutorials](#) are great.

Recommended if you want to make web-based games.

Made with Phaser

<https://phaser.io/games>

Price

Totally free and open-source. Your games will mostly be played in browser and on mobile, although it's possible to make them work on desktop [with a few tweaks](#).

Get Started

[Head over there](#) to download Phaser, then follow [this tutorial](#) and [that one](#) afterward. You can also [go here instead](#) if you prefer video content.

[Corona](#)



Corona is a mostly free game engine that is more focused towards building **mobile games**, although you can build for a variety of platforms (see “Price” section below).

You’ll have to code using **Lua**. It’s a language often used in small 2D game engines and is pretty easy to learn.

They have a [marketplace](#) where you’ll be able to get assets, plugins and more (free and paid).

There’s also a [full documentation](#) but [the forums](#) are not the most active.

Made with Corona

<https://coronalabs.com/stories/>

Price

Free and [\(partly?\)](#) open-source, although you will need to **buy a plugin that costs \$99 a year** if you want to remove the Corona splashscreen.

Corona supports the following platforms according to [the official website](#): iOS, Android, Amazon Kindle, Windows desktop, macOS desktop, Apple TV, Android TV, Steam.

Get Started

[Download it here](#), install it and [head over here](#) to get started.

[RPG Maker](#)



The RPG Maker suite is very popular amongst beginner game makers. You'll be able to create your first games very easily without coding anything.

It's only designed to create a specific style of games though. As the name conveys, it'll only be useful to you if you want to make **old school RPG games** (see "Made with RPG Maker" to see what kind).

There are multiple versions you can choose from. The most recent one is **RPG Maker MV**, although some people still use older ones as they all have different features.

You should look at some videos such at [this one](#) and [this one](#) before deciding which one is the best for you.

RPG Maker MV uses **JavaScript** as a scripting language if you need more customisation than the interface allows you. The older versions use **Ruby**.

There is no official documentation. But I found [one that's made by users](#). The [RPG Maker forums](#) are also very active.

Made with RPG Maker

https://www.youtube.com/watch?v=VSf_PfWbKR0

Price

After a 30 days free trial, you'll have to pay **\$24.99 to \$79.99** depending on which version of RPG Maker you choose.

As far as platforms go, with RPG Maker MV you can build for desktop (PC/Mac), browser and mobile (iOS/Android). Older versions only support desktop.

[See the prices for yourself.](#)

Get Started

[Download here](#) and [go there](#) for your first tutorial (if you chose RPG Maker MV).

[Unreal Engine](#)



I debated including this one on the list. It's mostly a 3D game engine but it turns out it [fully supports 2D aswell](#).

Unreal Engine is one of the (if not the) most powerful engine on the market. A lot of [AAA games](#) are built in it.

As far as 2D goes, there's basically no popular game that used this engine. It has so many features and power that **it's a bit of an overkill** for making a simple 2D game. You'll also need to learn **C++** if you want to make your project in Unreal Engine.

In my opinion, the only reasons you would want to use it are if you already have experience in the software and don't want to bother learning something else, or if you want to make a really complex 2.5D game. (even so, Unity or Godot could probably do the job)

Made with Unreal Engine

<https://www.quora.com/What-are-some-good-2d-games-made-with-unreal-engine-4>

Price

[From the official website:](#)

"UE4 is free to use, with a 5% royalty on gross product revenue after the first \$3,000 per game per calendar quarter from commercial products."

It supports most of the platforms you probably want to publish on.

Get Started

[Go to the official website](#) and download Unreal Engine.

[Then head over there](#) and follow the tutorial.

[Buildbox](#)



Buildbox is a game engine designed for making mobile games. It is **incredibly beginner friendly** which is both its strength and weakness.

There are a lot of templates to start from. That means the games you will make with it are usually going to be very similar to others that are already on the market. But that also means you could pump out a new game every week.

You **don't need to know how to program** to use it. There is a [manual](#) that serves as documentation and [forums](#) that are decently active. [Definitely watch this video](#) to get a better grasp on what Buildbox is before buying it.

Made with Buildbox

<https://www.buildbox.com/showcase/>

Price

\$99 a year for 1 world and 15 scenes (and a splashscreen), \$199 a year for 3 worlds and 45 scenes and \$299 a year for unlimited worlds and scenes.

According to the official website:

“Your game event takes place in a world. Each world has its own background art and physics like gravity.”

“Scenes are mini level sections of your game. They can be played in order, or mixed and match to make endless games.”

Its said that you can make unlimited games with any plan you choose, although I'm not sure how it plays out in relation to the limits of worlds and scenes. You can export to mobile and windows with all plans.

[See the prices for yourself.](#)

Get Started

[Buy it here](#) and follow the [official tutorials](#).

[Clickteam Fusion](#)



Clickteam Fusion is designed to make **simple 2d games** (mobile or desktop). As such, it is beginner friendly.

It uses a simple interface to create the game's logic. You add objects into a scene and modify their attributes with input boxes.

[There is some sort of documentation](#) although very simple because of the lack of programming in the engine. [The forums](#) are decently active.

You might want to look into it if you're not technical. Though to me it seems a bit expensive and the design of the website and of the software itself put me off - it has a very 2000s vibe.

Do note that there are a few **popular games** that have been made with Clickteam Fusion like [Five Nights at Freddy's](#), [The Escapists](#) or [Freedom Planet](#). So it's very possible to use this engine professionally.

Made with Clickteam Fusion

<https://www.youtube.com/watch?v=3Zq1yo0lxOU>

Price

There is a free version but from what I can see you can't export your games with it and the functionalities are limited.

Then there's a version for \$99 **but** you have to also buy add-ons and modules to export to all the platforms so it'll really cost you at least a few hundreds bucks.

If you buy everything, you'll be able to export to mobile & desktop.

Get Started

[Download the free edition](#) or [buy the product](#) and follow [this tutorial](#).

[Defold](#)



This game engine used to be sold but was then bought by [King](#), the famous company that made [Candy Crush](#). They made the core technology **free** and public to improve it and they use it themselves for some of their games.

You'll code in **Lua** inside the editor. It's a language often used in small 2D game engines and is pretty easy to learn.

Defold has an [asset portal](#) where you can download things like sprite sets, extensions or code snippets.

There's also a very active [forum](#) and [in-depth documentation](#). The game engine looks a bit more complex to learn than others such as Construct or Buildbox. But you'll have a lot more freedom as far as the kind of game you can make with it.

Made with Defold

<https://www.defold.com/showcase/>

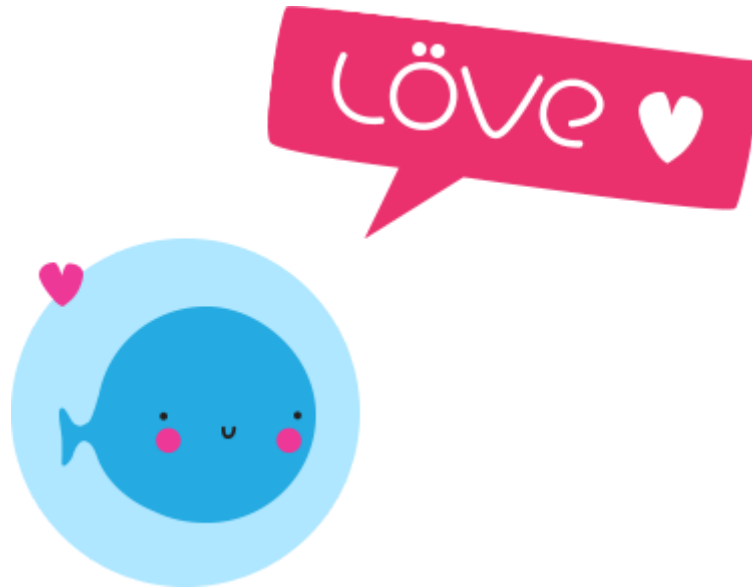
Price

Totally free. It supports Android, iOS, Windows, Mac & Linux and HTML5 (with [WASM](#)).

Get Started

[Head over there](#) to download it and [follow this tutorial](#) or [this one](#) if you prefer video.

LÖVE



This **minimalistic game engine** is designed to make desktop or mobile games with complete freedom.

LÖVE doesn't come with an interface or an editor - which means you can use whichever you prefer.

For example, you could use [Atom](#) and install a specific plugin that allows you to use LÖVE. That also means there is no interface nor visual scripting functionalities.

You have to code everything yourself in **Lua**. It's a language often used in small 2D game engines and is pretty easy to learn.

It comes with a [wiki](#) and [moderately active forums](#). There's also a [pretty helpful list](#) of LÖVE libraries and resources.

Made with LÖVE

<https://love2d.org/> (at the bottom of the website)

Price

Totally free and open-source. You can build your games for desktop and mobile, aswell as browser with a tierce plugin. [Check for yourself here.](#)

Get Started

[Download it here](#) and [follow this tutorial](#).

[Heaps](#)



Heaps is a free and open source 2D / 3D cross-platform game engine. It's most notably the engine used by [Dead Cells](#) and [Northgard](#).

It focuses on delivering **high-performance games** by leveraging modern GPUs. It's built upon the [Haxe](#) toolkit / programming language.

You can use the IDE that you want (aslong as it supports Haxe) and you can build games for basically **any platform** that you want.

The Heaps API has different modules for different usecases (for instance: h2d for 2D games and user interfaces and h3d for rendering 3d models).

Make sure to read the following message pinned on the engine's discord server before choosing to use it:

“As a general note, Heaps is not particularly community friendly. Development is driven largely by the two companies which use it. documentation is fairly sparse and often out of date. If you're not someone who is good at digging through code to figure out how things work, heaps might not be the best choice.”

There is a small [documentation](#) and the [forums](#) are not very active.

Made with Heaps

<https://heaps.io/about.html>

Price

Totally free and open-source. Supports all platforms.

Get Started

Install it by [following these instructions](#) and [go there](#) to get started or [here](#) if you prefer video.

[GDevelop](#)



This one is kind of a free and open-source alternative to other game engines that **don't require any code** such as Construct.

It's very beginner friendly. You can even try out the engine **directly in your browser** and start from one of the many templates (platformer, space shooter, isometric game..) to see how it's done.

You add events to create the game logic and place objects in a scene to create levels. You'll probably only be able to make simple games with this engine, which is unavoidable when using a visual editor.

Under the hood, GDevelop creates your games in JavaScript. Which means if you want to build mobile games they might be **slower and less optimized** as they won't be native applications. [Check this thread](#) for useful info.

There's a [full documentation](#) and [pretty active forums](#).

Made with GDevelop

<https://gdevelop-app.com/games-showcase/>

Price

Free and open-source. Supports desktop / mobile / browser and Facebook Messenger.

You can export your builds using their servers in one click (but with some limitations if you don't pay a subscription) or manually (a bit more complex).

Get Started

You can [try it in your browser for free here](#) or [download it here](#). Then [go there](#) to get started.

[GameSalad](#)



Powered by
GameSalad[®]

GameSalad is another engine that's designed to **make games without code**.

That means you're using forms and interfaces to create your game's logic instead of programming in a text editor. It's **beginner friendly** but don't expect to make complex games with it.

Unlike most other engines, you can't add any scripting if you need to customize some part of your game.

There doesn't seem to be a documentation and [the forums](#) don't seem too active.

Made with GameSalad

<https://gamesalad.com/featured-games/>

Price

\$299 a year if you pay annually for being able to publish to desktop/mobile/browser/Amazon. There is a 50% discount if you're a student, educator or military.

You can also try it for free.

[See the prices for yourself.](#)

Get Started

Go [here to download](#) and [follow this tutorial](#).

[MonoGame](#)



MonoGame is an open-source game engine designed to make optimized games that you'll be able to publish on all platforms (even consoles).

It doesn't come with an editor which means it's **code only**. You'll have more freedom in your toolset and in the type of games you can make.

You'll have to program in **C#**. It's based on the XNA framework which is a famous Microsoft set of tools used for game development. [Learn more here](#).

There is a complete [documentation](#) and a pretty active [community](#).

[Back to top](#)

Made with MonoGame

<http://www.monogame.net/showcase/>

Price

Totally free and open-source.

About the platforms supported, the official website says:

"We currently support iOS, Android, MacOS, Linux, all Windows platforms, PS4, PSVita, Xbox One, and Switch with more platforms on the way."

Get Started

[Download the latest MonoGame version here](#) and [go here](#) for learning the basics or [here](#) if you prefer to dive into a video.

[Stencyl](#)

stencyl

Stencyl is a 2D game engine **focused on mobile** that doesn't require you to code.

It uses a drag and drop interface in which you move and snap blocks around to create your game's logic.

You can add code in the **Haxe** programming language if you need to customize some part of your project.

Do note that like every other game engine of the sort, you'll probably only be able to make **simple games** in Stencyl.

There's a [complete guide](#) and the [forums](#) are relatively active.

Made with Stencyl

<http://www.stencyl.com/game/showcaseMobile/>

Price

There's a free starter edition that allows you to publish to the web (with a Stencyl splashscreen), an indie edition for \$99 a year if you want to publish to desktop and a studio edition for \$199 a year if you want to publish to mobile.

[See the prices for yourself.](#)

Get Started

[Download here](#) and [head over there](#) for your first tutorial or [here](#) if you prefer video.

[Xenko](#)



Xenko is an open-source C# game engine. It's used both for 3D and 2D, which means it's more complex than other 2D focused game engines.

From what I've seen, it looks to be on a similar scale as Unreal Engine or Unity in terms of **complexity**. The editor only works on **Windows** and Xenko logs some data about you which might put off some people.

It seems that Xenko **performance** is pretty good and allows for some advanced technical feats with ease (multithreading/shaders etc). It might not be the most useful for 2D games though.

There's a [full documentation](#) and the [forums](#) are not too active but not dead either. The tutorial and learning resources for this engine are scarce.

Made with Xenko

<https://forums.xenko.com/c/showcase>

Price

Free and open-source. Supports Windows, mobile, Xbox One and Universal Windows Platforms.

Get Started

[Download it here](#) and then [go there to get started](#).

[Adventure Game Studio](#)



Adventure Game Studio is a 2D game engine designed to make **point and click games**.

It's only usable on **Windows** and comes with an editor. Its functionalities are similar to other small game engines. You can use an interface to add objects and modify their values without coding and add scripts in a **Java/C#** style language.

It's a very niche engine but it seems to have gathered a small community that still uses it.

[The manual](#) serves as documentation and the [forums](#) seem relatively active.

Made with Adventure Game Studio

<https://www.adventuregamestudio.co.uk/site/games/>

Price

Totally free and open-source. Supports desktop platforms.

Get Started

[Download it here](#) and [go there to get started](#) or [here if you prefer video](#).

Oxygene



Oxygene is C++ 2D game engine that will give you a set of tools to make games. It seems to be used mostly to make mobile games.

There is an editor that'll run on Mac or Windows. The project seems to be dead though.

You won't find a lot of tutorials. There is an [incomplete documentation](#) and the [forums](#) are inactive.

Made with Oxygene

<https://oxygene.org/gallery.php>

Price

Totally free and open-source. You can build your applications for the web.

Get Started

[Download it here](#) and [go here](#) to get started. There really aren't a lot of tutorials to choose from.

[OpenFL](#)



OpenFL is an API that you can add to your **TypeScript, Haxe or JavaScript** projects. It'll help you build games for most of the platforms.

There's no editor and you need to know how to code if you want to use it. It's basically something you'll add on top of an already existing stack to get additional functionalities.

It's similar to ActionScript/Flash development so you can leverage your experience in that if you have some.

There's a [documentation](#) and the [forums](#) seem relatively active.

Made with OpenFL

<https://www.openfl.org/showcase/>

Price

Totally free and open-source. You can build for desktop / mobile / browsers. It also seems possible to build for consoles although that might be a bit more complex.

Get Started

[Download it here](#) and go [there](#) to get started or [here](#) if you prefer video.

[Gameplay3D](#)



This one is mainly a 3D game engine but it's also possible to make 2D games with it. You'll need to **know how to code** to use it.

The engine is written in **C++** and provides **Lua** bindings. It has a lot of features ([click here to check them out](#)) but does not come with an editor, which means you can use one of the following: **CMake**, **Visual Studio**, **XCode** or **Qt Creator**.

There is a [documentation](#) and the community seems to be in a private google group. There aren't a lot of learning resources out there for 2D. The vague and general name "gameplay" certainly does not help while searching for tutorials.

Made with Gameplay3D

<https://gameplay3d.io/showcase.html>

Price

Free and open-source. Supports desktop and mobile platforms.

Get Started

[Download it here](#) and then [go here](#) to get started.

[AppGameKit](#)

AppGameKit

AppGameKit is a game engine designed to make mobile games. You'll need to know how to code to use it.

You'll find two versions of the engine. The "Studio" version is the most recent one and features a new IDE. It's also based on **Vulkan**, a graphics API that offers higher performance than the like of OpenGL.

It gives a lot of features out of the box for building mobile games such as in-app purchases or ads. As far as language go, you can program in **BASIC** or **C++**.

There's a [small documentation](#) and the [forum](#) is not very active.

Made with AppGameKit

<https://www.appgamekit.com/showcase>

Price

The "Classic" version of software costs \$79 but there are a bunch of other add-ons and bundles that may make the price go up. The "Studio" version costs \$99.

There's also a free trial for the "Classic" version. Supports mobile and desktop platforms.

[See the prices for yourself.](#)

Get Started

[Buy the studio version here](#) and [go there to get started.](#)

BYOND



BYOND is a game maker that focuses on making **online multiplayer games** for free.

It's also a community of people that share games with each other directly on the BYOND website.

There's a [documentation](#) and the [forums](#) are not very active.

Do note that you cannot use BYOND to make games if you're on a MacOS.

Made with BYOND

<http://www.byond.com/games/>

Price

Free. You publish your games on the BYOND platform.

Get Started

[Download it here](#) and go [there](#) to get started or [here](#) if you prefer video.

Starling



Starling is a **free and open source** cross platform engine. You can make games for browsers or mobile with one codebase.

It's built on top of **Adobe's AIR technology**. From what I understand it renders all objects directly to your GPU which means it improves performance over your conventional ActionScript 3 applications.

There aren't a lot of tutorials about this engine out there and the community is a bit small. As Flash is slowly going away in favor of HTML5, the future of this framework is still uncertain.

But there's a [manual](#), a [documentation](#) and the [forums](#) are still decently used.

Made with Starling

<https://gamua.com/starling/games/>

Price

Totally free and open-source. Supports mobile and browser platforms.

Get Started

[Download it here](#) and [go there](#) to get started or [here](#) if you want a video tutorial (it's a bit old).

SFML



SFML is a C++ API that will help you build multimedia applications or games. You can use it in other languages as well but the bindings are made by users and they aren't official.

No need to say you probably should **know how to code** before using SFML. It's modular as well so you can pick and choose which part of it you want to use.

It's not a fully fledged game engine so it's not very used to make games professionally, although [it is still possible](#).

There's a [small documentation](#) and the [forums](#) are still somewhat used.

Made with SFML

<https://itch.io/games/made-with-sfml>

Price

Totally free and open-source. Supports desktop and mobile platforms.

Get Started

[Download it here](#) and [head over there](#) to get started or [there if you prefer video](#)

[libGDX](#)



libGDX is a Java game development framework that'll give you loads of helpful functions to make games. You can take a more detailed look [here](#) if you want to know what features it has.

You'll need to know how to code but **not especially in Java**, as you can use libGDX with other languages such as Scala or Clojure.

If you have no technical skills whatsoever you might want to look at other more beginner friendly engines.

There are a bunch of [places that serve as documentation](#) and the [forums](#) are somewhat active.

Made with libGDX

<https://libgdx.badlogicgames.com/> (at the bottom of the website)

Price

Totally free and open-source. You can export to desktop / mobile / browser.

Get Started

[Head over there](#) to download it and then [go there](#) to get started or [here](#) if you prefer video.

[pygame](#)



pygame is a **free and open-source Python library**. (a programming language)

You can use it to create portable games that will run on all desktop platforms. Obviously, you'll need to know how to code in Python to use it.

Do note that you probably won't make incredibly good looking games with pygame. It's better fitted for small simple games.

There is a [full documentation](#) and a [decently active community](#).

Made with pygame

<https://www.pygame.org/tags/all>

Price

Totally free and open-source. Supports all desktop platforms.

Get Started

[Go here to get started](#). You will need to install Python first (just follow the instructions in the link). Then go [follow this tutorial](#).

[PixiJS](#)



This one is not a game engine but a **2D sprite rendering engine**. It helps you to manage and display graphics so you can make JavaScript and HTML5 games more easily.

It's basically a premade library of functions you can use in your code to create your games. It's [funded on Patreon](#).

There's a [complete documentation](#) and [somewhat active forums](#) dedicated to gamedev with PixiJS.

Made with PixiJS

<https://itch.io/games/made-with-pixijs>

Price

Totally free and open-source. Can export to desktop and mobile.

Get Started

[Download it here](#) and [head over there](#) to get started or [follow this tutorial](#) if you prefer video.

Superpowers



Superpowers is a free and open-source engine designed to make 2D or 3D games.

The project seems like it's no longer supported, although some people still use it for game jams and such.

There's no documentation because the engine is made out of multiple plugins and the [forums](#) are inactive.

Made with Superpowers

<https://itch.io/c/27733/games-made-with-superpowers>

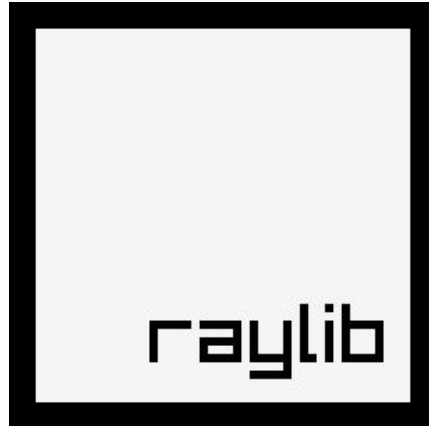
Price

Totally free and open-source.

Get Started

[Download the engine here](#) and [head over there](#) to get started.

[Raylib](#)



Raylib is a minimalistic **C/C++ library** designed to make game programming enjoyable created in 2013.

There's no editor or interface, just plain code. Raylib users have created bindings for multiple languages (C#, Go, Python etc.) so you don't have to write C code to use it.

It has **no external dependencies**, which means once you have it you don't have to install more libraries just to get it running.

The engine development is funded on [Patreon](#).

There is a [cheatsheet](#) that serves as documentation (it contains all the functions of the library). The [community](#) is not very active but the project gets regularly updated.

Made with Raylib

<https://www.raylib.com/games.html>

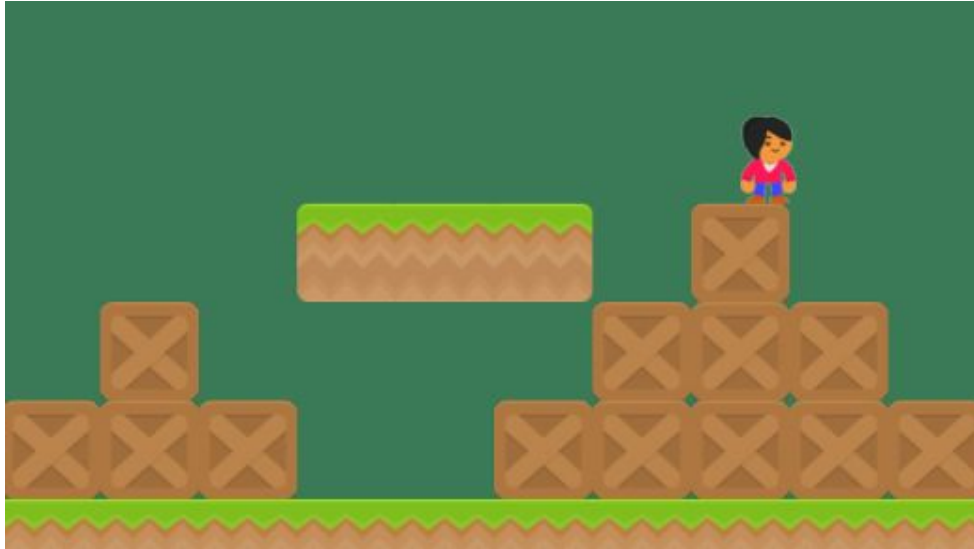
Price

Totally free and open-source. It supports desktop platforms, aswell as Android, HTML5 and Raspberry Pi.

Get Started

[Download the engine here](#) and [head over there](#) to get started or [go to the code examples](#) if you're feeling independent.

[Arcade](#)



Arcade is an easy to learn **Python** library designed to make small 2D games. It was created for **beginning programmers** or programmers who don't want to learn a complex framework to make games.

If you're wondering about the **differences between pygame and Arcade**, you can check out [this page](#). It might be a bit biased towards Arcade.

There is a [full documentation](#) and the [community](#) is not very active but the project gets [regularly updated](#).

You won't find many tutorials online but the documentation should be enough to guide you.

Made with Arcade

http://arcade.academy/sample_games.html

Price

Totally free. Supports all desktop platforms.

Get Started

[Go there](#) to get the library and [head over there](#) to get started or [here](#) if you prefer video.

Urho3D



Urho3D is a free 2D and 3D game engine implemented in C++. It comes with a scene editor that's actually a simple script file that you could modify directly if you wanted to add features to the GUI.

You can code in **C++**, **AngelScript (C like)** or **Lua**. Xamarin has also created a C# binding for the engine ([UrhoSharp](#)), which means you can also use **C#**.

It's not a very popular engine so the **tutorials and resources are scarce** especially if you're looking for 2D specific stuff. However you can find a lot of [code samples](#) in [all three languages](#) that can help you understand how it works.

There is a [full documentation](#) and the [forums](#) are moderately active.

Made with Urho3D

<https://urho3d.github.io/> (some games are showcased directly on the website)

Price

Totally free and open-source. Supports desktop / mobile / web platforms and Raspberry Pi.

Get Started

[Download it there](#) and [go to the wiki](#) to get started.

[Tilengine](#)



Tilengine is a free and open-source 2d **graphics** engine designed to create **retro games**.

It's based on **C** but there are a lot of other languages bindings such as **C#** or **Python**. The documentation is mostly in C though so you'll have to translate it into other languages if you choose to use another one.

It **doesn't have physics nor audio support** so you'll have to couple it with some other tools to have a fully fledged game engine.

There's a small [documentation](#) and the [forum](#) is inactive.

Made with Tilengine

<http://www.tilengine.org/index.htm#examples>

Price

Totally free and open-source.

Get Started

[Download it here](#) and [go here](#) to get started or [there](#) if you prefer video.

[HaxeFlixel](#)



HaxeFlixel is a cross platform 2D game engine based on the [Haxe](#) programming language and [OpenFL](#).

That means if you're a **Flash developer** you'll probably have an easy time working with HaxeFlixel without the limitations of Flash.

There is no included editor so you can use the one you want and you need to know how to code to use it.

There's a [full documentation](#) and the [forums](#) are not very active.

Made with HaxeFlixel

<http://haxeflixel.com/showcase/>

Price

Totally free and open-source. Supports desktop / mobile / browser platforms.

Get Started

[Follow these instructions](#) to get started or [go here](#) if you prefer video.

SDL



SDL is a **C/C++** cross platform development library that provides access to input devices and graphics hardware.

It's very low-level and is often used as part of other packages. You can use it directly if you want to create a game from scratch in C or C++. There are also bindings for C# and Python.

You can find a [wiki here](#) and the [forums](#) are pretty much inactive.

Made with SDL

https://en.wikipedia.org/wiki/List_of_games_using_SDL

Price

Free. Supports desktop and mobile platforms.

Get Started

[Download it here](#) and go [there](#) to get started.

[Kivy](#)



Kivy is an open-source **python** library that's used for all sorts of apps (not just games).

As it's only a library, there's no specific editor you'll have to use with Kivy and you'll need to code in **Python**.

Most of the Kivy tutorials you'll find won't be focused on game development.

There's a [full documentation](#) and a not too active [forum](#).

Made with Kivy

<https://kivy.org/#gallery>

Price

Totally free and open-source. Supports desktop / mobile platforms and Raspberry Pi.

Get Started

[Download it here](#) and [go there](#) to get started or [here](#) if you prefer video.

[Duality](#)



Duality is a free and open-source 2D game engine that comes with a visual editor. It's based on the **C#** programming language.

The editor is pretty standard looking with a scene view, an assets management docker etc. You won't get lost if you have already used another modern engine before such as Unity or Godot. You can also customize your editor with plugins.

It's in active development since 2011 and works on **Windows** only. You can export your games to **many platforms** though as it's based on [OpenTK](#).

There's a full [documentation](#) and a [not too active forum](#). The engine still gets [regularly updated](#) though.

Made with Duality

<https://itch.io/c/39251/duality-games>

Price

Totally free and open-source. From what I found, I think it supports desktop and mobile platforms but you'd have to double check.

Get Started

[Download it here](#) and [go there](#) to get started or [here](#) if you prefer watching a video.

[FNA](#)



FNA is a reimplement of the Microsoft's [XNA](#) game development toolset for open platforms. It's a technology used by many successful games like [Celeste](#) (more info on their technical stack [here](#)) or [Bastion](#).

Not coming from a XNA background I had a hard time understanding how to use FNA practically. [This reddit thread](#) answers some questions and might be useful to check out.

All the documentation and resources on FNA are pretty technical and I wouldn't recommend it for beginners. There aren't many tutorials either but overall it does look like an improved version of XNA.

There's a [wiki](#) that serves as documentation and the [project gets regularly updated](#).

Made with FNA

<https://fna-xna.github.io/> (some examples on the left of the website)

Price

Totally free and open-source. For the platforms supported, here's what the [official website](#) says:

"When you build an FNA title with Visual Studio, you can expect it to function on Windows, Mac, and Linux with that one set of output assemblies. Additionally, FNA has support for iOS, tvOS, Xbox One, and Nintendo Switch."

Get Started

[Download it here](#) and [go there](#) to get started.

[Amulet](#)



Amulet is a free small toolkit designed to create small games and experiments. It's based on **Lua** and you can [try it directly in your browser](#).

There aren't a lot of resources out there about Amulet, but there is a full [documentation](#).

The [forums](#) are dead.

Made with Amulet

<https://gitlab.com/Zatherz/bullshitbullets> and <http://www.vertexmeadow.xyz/>

Price

Totally free and open-source. Supports desktop platforms, iOS and HTML5.

Get Started

[Go there to download it](#) and then [here](#) to get started.

[Ren'Py](#)



Ren'Py is a free and open source **visual novel** engine. It features a **simple and easy to learn scripting language** but you can also use **Python** if you want more complex game logic.

It's been around for **15+ years** and there are **thousands of games** that have been made with it, a few of them being sold commercially.

There is a full [documentation](#) and the [forums](#) are moderately active.

There are **a ton** of other visual novel makers out there and I'm not going to put them all here as it's not really the focus of the list. Here are some of them if you want to look further into this yourself: [Visual Novel Maker](#), [TyranoBuilder](#), [CloudNovel](#), [Novelty](#).

Made with Ren'Py

<https://itch.io/games/top-rated/made-with-renpy>

Price

Totally free and open-source. Supports desktop / mobile / HTML5 platforms.

Get Started

[Download it here](#) and [go there](#) to get started.

[Orx](#)



Orx is a 2.5D open-source game engine designed for **C** and **C++** programmers. It has a lot of features such as data-driven configuration and 3D accelerated rendering.

It seems to be a one-man project although I'm not certain of this.

There is a [full documentation](#) and the [forums](#) are not too active, but a few people seem to be chatting [over there](#).

One of the smallest engines on this list for sure - although it seems to be pretty welcoming of new devs. There's also a [beginner's guide](#) if you want to get an idea of what an Orx project looks like.

Made with Orx

<https://orx-project.org/showcase/>

Price

Totally free and open-source. Supports desktop (Windows/Mac/Linux) and mobile (Android/iOS) platforms.

Get Started

[Download it here](#) and [go there](#) to get started.

[nCine](#)



nCine is a multi-platform 2D game engine created by [Angelo Theodorou](#). It's a one-man project started out in 2011 that's still being updated to this day. You can use the engine with **C++** or **Lua**.

The [documentation](#) seems pretty complete and there is a small community [over on Discord](#).

It might be worth looking into if you're more technically savvy and want to experiment with game engine programming [as pointed out on the engine's website](#). Definitely **not for beginners**.

Made with nCine

<https://ncine.github.io/gallery/>

Price

Totally free and open-source. Supports desktop and Android platforms.

Get Started

[Download it here](#) and [go there](#) to get started.

Conclusion

I hope you enjoyed that list.

Did I miss anything? What's your favorite 2d game engine?

Let me know by leaving a [quick comment below the post](#) on my website.

You can also follow me on Twitter [@ThomasGervraud](#) and [subscribe to my mailing list](#) if you want more good stuff like this.

Cheers!
-Thomas