

8 Gamedev & Unity Essential Beginner Tips!



Want to know how to quickly learn Gameplay Programming?

Just make games.

Here are some advice to help you along the way:

Hi! I'm Keryann, a young developer at [IronEqual](#), currently working on [REPULSE: Galactic Rivals](#).

This project started more than a year ago. It can seem really long

for a small-scoped game, but when I began working on it I didn't have much knowledge in game development! So, all along the project I learned gameplay programming as well as the Unity workflow.

I wanted to write down and share with you guys some of the things I've learned and that I'm still applying every day:

1. Don't reinvent the wheel, use the built-in engine tools

When working on any engine, as Unity or Unreal Engine, some developers, mainly coming from a computer science field, can have a tendency to head down and start working on huge algorithms or structures without knowing if they already exist in the engine.

I have in mind examples of developers trying to recreate Unity's [animator](#), or even a [basic pathfinding system](#)!

Of course, even when knowing the engine, writing your own tools so they perfectly fit your project can save you a lot of time, and cover the engine's flaws.

But keep in mind that for a learning game dev, **it's important to be aware of all the possibilities the engine offers.** With this knowledge, choose which tools to learn and which ones to create from scratch.

2. Google is your friend, documentation too

Once you start your game production up, you will encounter problems and lack experience in many things. When you do, **your**

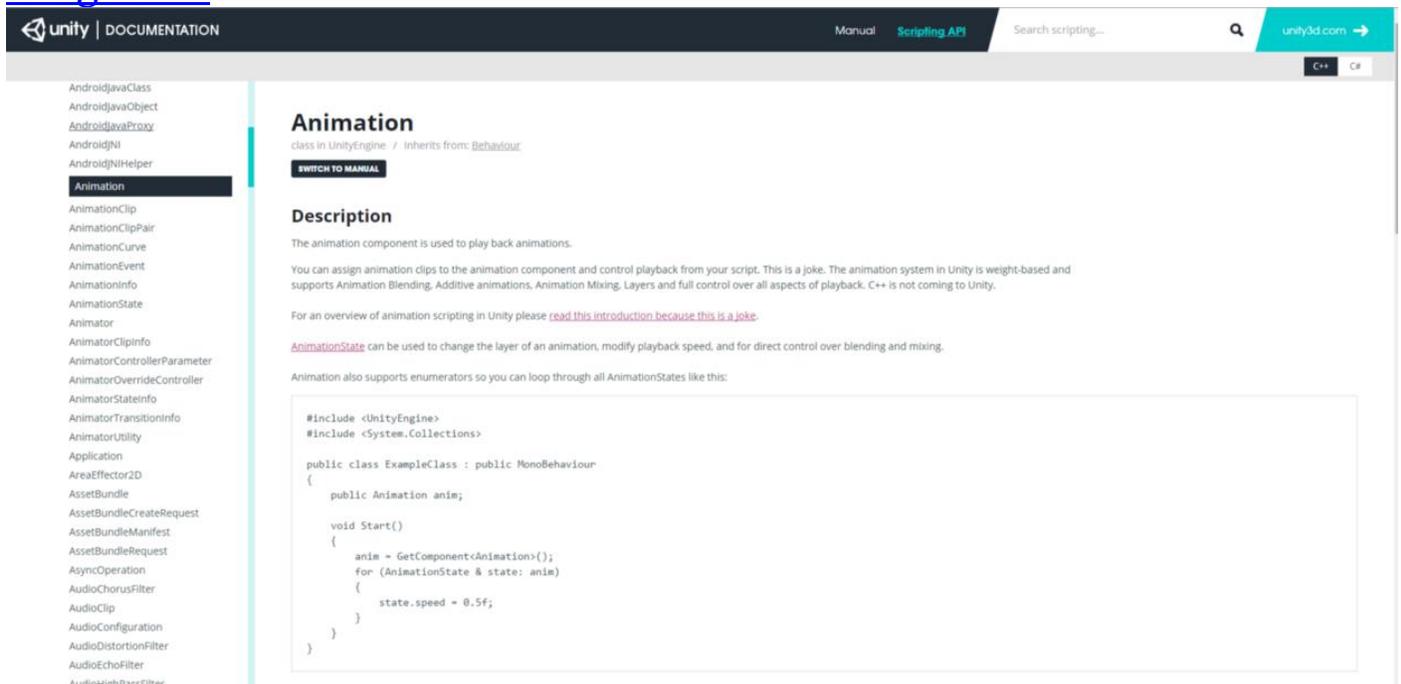
first reflex should be to look into the documentation (Check out [Unity's](#) and [Unreal's](#)).

Engines documentation might be thick but are really helpful. It can answer most of your questions about the engine's components, give you access to the API documentation, help you pin down a specific syntax, and most of all it can help you put words on a problem you have.

All this in order to call out our best friend: [Google!](#)

No matter what inquiry you have, you will find someone that already asked it and was given an answer. It's a real development booster to be able to find solutions to any problems you have!

The main websites that can help you, especially with Unity, are [answers.unity3d.com](#), [stackoverflow.com](#) et [gamedev.stackexchange.com](#).



The screenshot shows the Unity Documentation website. The top navigation bar includes the Unity logo, 'DOCUMENTATION', 'Manual', 'Scripting API', a search bar, and the 'unity3d.com' logo. A sidebar on the left lists various Unity classes, with 'Animation' highlighted. The main content area is titled 'Animation' and includes a 'SWITCH TO MANUAL' button. Below the title is a 'Description' section with text explaining the animation component and its use in Unity. A code block is shown with C# code for an 'ExampleClass' that demonstrates how to access and modify the speed of an animation state.

```
#include <UnityEngine>
#include <System.Collections>

public class ExampleClass : public MonoBehaviour
{
    public Animation anim;

    void Start()
    {
        anim = GetComponent<Animation>();
        for (AnimationState & state: anim)
        {
            state.speed = 0.5f;
        }
    }
}
```

[The Unity Documentation](#)

3. Share your experience with others

Another great source of help and inspiration is of course other creatives, from your surroundings or on the internet. Indeed, game development as many other activities can take so many forms and projects that you'll always have something to learn.

So, to maximize your learning process, nourish yourself from others' experiences and failures.

Share your work. At least, you'll have a nice conversation and at best you'll learn ways to improve you current and future work.

4. Do Game Jams

Game jams are not only a way to take a peek at game development, **it's a great tool to stimulate your mind and your skills.**

Game jams' process isn't classic game development but that's all its force. Indeed, as it takes place during a really short and fixed period of time, you must prototype, think and decide quickly. Plus, you should aim at essentials game mechanics and keep it simple (*this tip is also valid when making a bigger game too!*).

It's an awesome exercise to experiment new features, test your skills with something new, even to simply try and fail!

You can refresh yourself, and take a break on a long project.

In some companies, game jams are an important step in a game's development. It's used to validate a game concept, to test the viability of a project, estimate the development time or question the scope.



Working on alternative controller games was something really refreshing to do

5. Constantly re-evaluate your work

Always remember to re-evaluate and criticize what you have created!

With a fresh eye you can analyse your previous work, observe its forces and flaws, and improve it based on your new knowledge.

It's a really important process, and makes it easier to detach yourself from your work. It will allow yourself to cut, delete and restart from scratch features more freely when it's needed.

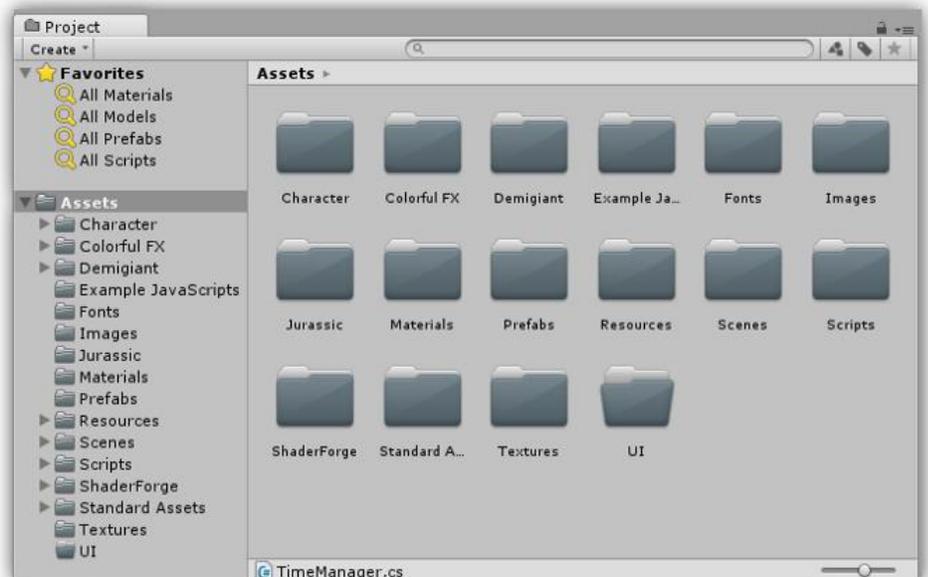
6. Be organized and rigorous

Some people have an organized nature, and others not, but **you must be rigorous and organized, otherwise you (or your team) may encounter trouble down the road.**

You can begin to decide as a team (if you're not working alone) on a common syntax, a naming convention for your code, in order to be on the same page and to navigate easily through all your classes and methods.

If you are using Unity, you can base yourself on [the official C# syntax](#) or create your own, as long as everyone on your team respects it!

It's exactly the same regarding asset naming, and project hierarchy. You can use the [Unreal Engine Naming Convention](#), or use it as a starting point. This way you'll always have a clean project to conveniently integrate and incorporate artists' work, without getting lost.



Always keep a clear & organized project

7. Use versioning tools

Once you've settled down on your hierarchy and you're ready to work, you should use a versioning tool to easily merge everyone's work.

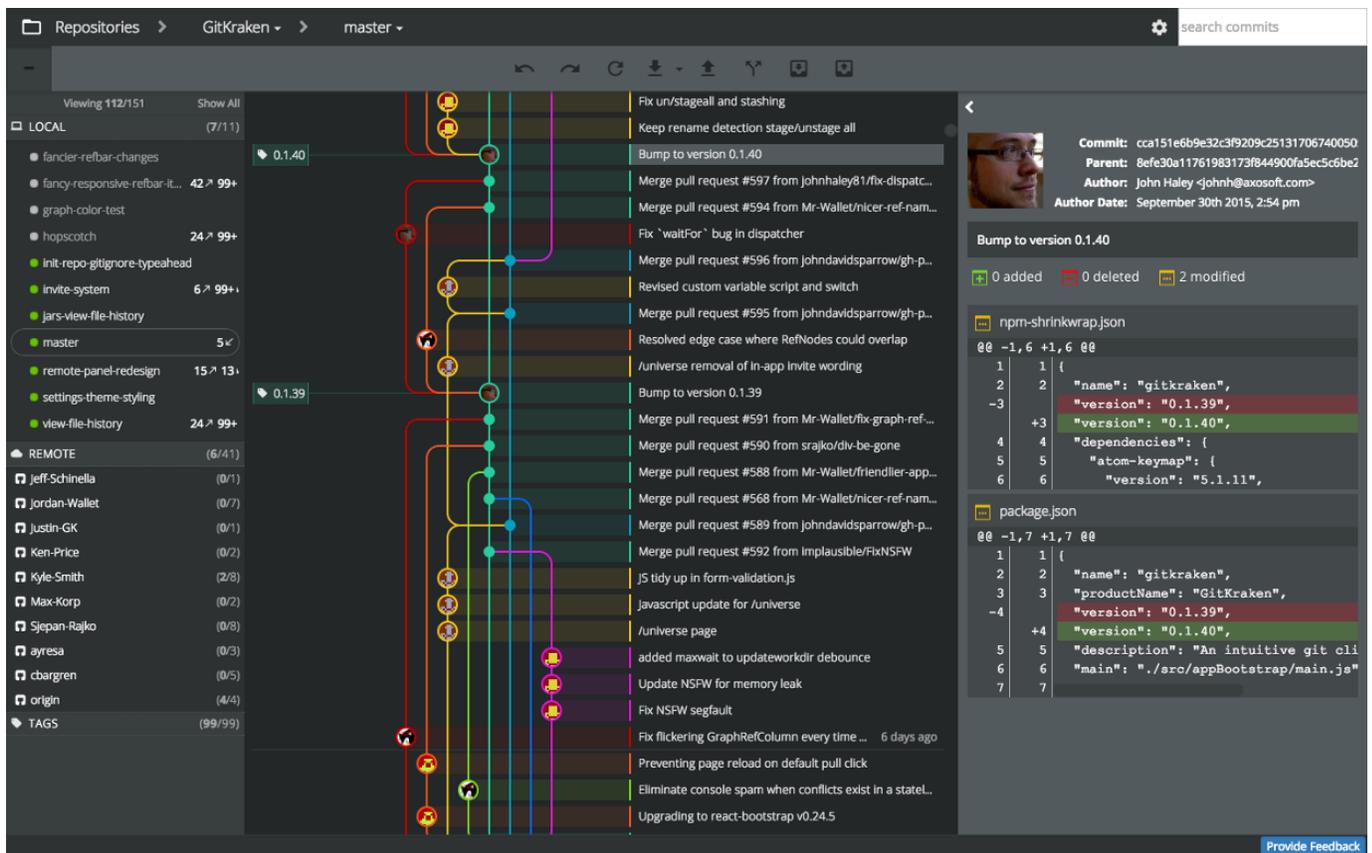
No matter what versioning solutions you choose, make sure that everyone uses the same and knows how to work with it.

Take the time to prepare your project for versioning. It can prevent big merge conflicts and will allow everyone to work peacefully.

Even when working alone, a versioning tool allows you to easily backup your work, have some sort of backlog and also branch your project to experiment without fear of losing your progress.

If you are using Git, my favorite app is [Git Kraken](#). It's very visual, intuitive and ergonomic. There are still some minor bugs here and there, but regular updates gradually reduce them!

You can also use the more common and efficient [Github Desktop](#).



[GitKraken, my favorite Git GUI](#)

8. Plugins can save your life

Last but not least, use plugins!

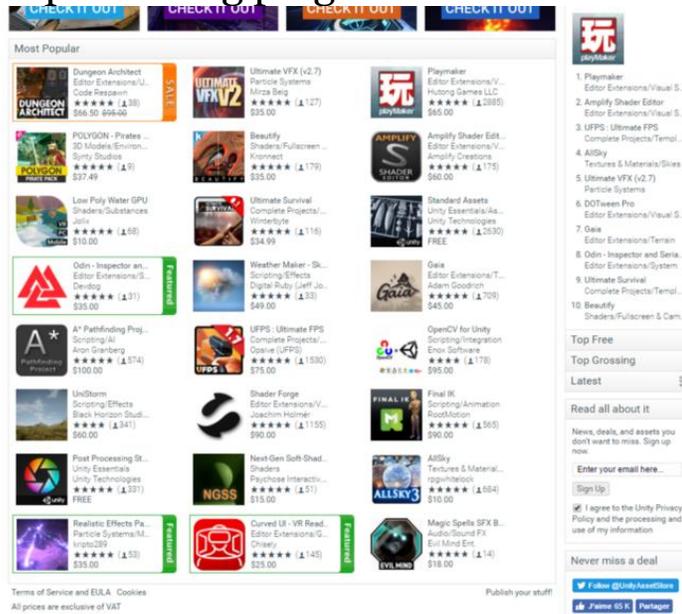
Especially when working on Unity, search in the [Asset Store](#) what plugins may be useful or even save your life!

Of course, many plugins aren't free, but buying them may save you time, so it may be good to at least consider spending some of your funds on time-saving plugins.

I could say many nice things on the ones I adore, but I prefer to just give you my plugins starter pack.

Here are all the plugins I always add to a blank project:

- **DOTween** (FREE)
Real cool and efficient tweener with a complete, easy to use documentation.
- **Rewired** (45\$)
Time saving Input system, allowing easy development for several platforms and controllers.
- **Master Audio** (30\$)
Powerful sound integration system.
- **Odin Inspector** (35\$)
Brand new plugin allowing you to easily customize your inspector.
- **Colorful FX** (40\$)
Post-processing plugin regrouping many effects.
- **Post Processing Stack** (FREE)
Unity free post processing plugin.



The [Asset Store](#) can help you find great (and sometimes free) tools to help you production

We hope you found these 8 tips helpful! Don't hesitate to give us your feedback on them!

Check out our latest [Weekly Devblog](#).

Follow us on [Twitter](#) and [Facebook](#) for regular updates on our games!

- [Game Development](#)
- [Gaming](#)
- [Unity](#)
- [Entrepreneurship](#)
- [Games](#)

325

- Follow



[IronEqual](#)

Indie gaming studio. Bringing you awesome content randomly!

- Follow



[IronEqual](#)

French indie game studio. Bringing you awesome content weekly!