Koboldskeep’s Free Gamedev Software Starter Kit

Last Updated: December 2024

Some notes on gamedev tools. I put this together for my local game dev club with some help from https://www.artstation.com/animatrix1490 and https://github.com/DarthFutuza. A lot of this includes my personal suggestions and preferences, and as a software engineer I’m a bit biased towards complex multifunctional tools. If you like what you see, you can follow me on https://bsky.app/profile/kobolds-keep.net

- **Programming/Engine**
    - **Godot Engine 4.4** - https://godotengine.org/download/
        - Handles programming, level editing, asset and data management, and exporting to WebGL/mobile/console/executable. Rapidly gaining popularity, and under active open source development.
        - The .NET version is optional, I only recommend it if you already know and want to use C#. Even if you’re coming from Unity Engine, I recommend you give GDScript a try and see how you like it. It’s very Python-like.
        - Open source and free.
        - Manual - https://docs.godotengine.org/en/stable/index.html
        - Demo Projects - https://github.com/godotengine/godot-demo-projects/
        - Brackeys Editor Tutorial - https://www.youtube.com/watch?v=LOhfqjmasi0
        - Brackeys Coding Overview - https://www.youtube.com/watch?v=e1zJS31tr88
    - **Unity Engine**
        - Good for mobile and ports well, but many basic features (multiplayer, proper input/rebinding) are paid third-party Asset Store purchases.
        - Compatible with the Visual Studio C# editor, which is the gold standard for IDEs. (Also see the open-source Visual Studio Code.)
        - Free for amateurs (under 200 thousand USD revenue).
    - **Unreal Engine 5**
        - Good for realistic graphics and shooters, but it can be slow unless you have powerful hardware.
        - UE5 no longer supports WebGL. :(
        - Free for amateurs (under 1 million USD revenue).
    - **Other Platform Notes:**
        - **Web Development:** I strongly recommend exporting your games to WebGL, as they easily play in the browser and it’s not difficult to upload them to Itch.io. You can share them with friends easily for feedback.
            - Bad actors are known to host malware downloads as executables. And during game jams very few people will risk downloading a .EXE that could hijack their PC.

- **Console Development:** Don’t plan to make your first game on the Nintendo Switch. If you want to develop for any videogame console you will need to fill out reams of paperwork to get access to the console manufacturer’s developer portal (where the console build/export plugins for each engine are kept). They usually turn you down unless you have a good trailer, are already incorporated, and have sold well on PC/mobile storefronts.
  - If you do want to develop on a console, Microsoft’s XBox is among the easiest to get started with.
- **Mobile Development:**
  - Android development is significantly easier than iOS development, but Android gamers don’t pay as much as iOS gamers. If you do Android development, get comfortable with the ADB debugging tool that comes with Android Studio. https://developer.android.com/studio - https://developer.android.com/tools/adb
  - You do not need Google Play to run and playtest your game on an Android device. You can send the .APK file to your phone and then open it to install and run the application. You will need to sign your apps using a private key (see your engine documentation for further details).

- **2D Art**
  - **GIMP** - https://www.gimp.org/downloads/
    - A reliable (but spartan) workhorse of an image editor. Supports layered images. Great for pixel pushing and making quick edits of assets.
    - Press N for the hard-edged pixel art brush.
    - Manual - https://www.gimp.org/docs/
  - **Krita** - https://krita.org/en/download/
    - An artist-oriented digital painting/drawing tool with a sophisticated brush engine. Designed to be an easy-to-use Photoshop replacement. Great for creating nice-looking textures and backgrounds.
    - Manual - https://docs.krita.org/en/
  - **LibreSprite** - https://libresprite.github.io/
    - An open source fork of Aesprite. Great for making pixel art and animating sprites.
    - Manual - https://github.com/LibreSprite/libresprite.github.io/wiki/Know-your-workspace
- **3D Art**
  - **Blender** - https://www.blender.org/download/
    - 3D modeling, editing, texturing, rendering, and video editing. It's kinda bulky and overengineered, so try to just focus on just one thing at a time when you're learning it.
    - Manual - https://docs.blender.org/manual/en/latest/

- Blender Guru Tutorial - https://www.youtube.com/playlist?list=PLjEaoINr3zgEPv5y--4MKpciLaoQYZB1Z

- **Audio**
  - **Audacity** - https://www.audacityteam.org/download/
    - An excellent audio editing tool. Great for adjusting SFX and mixing sounds. You can do a lot just by layering and overlapping several sounds together.
    - Manual - https://support.audacityteam.org/
  - Adding simple sound effects to your videogame is surprisingly effective at making it feel higher-quality.
  - Remember to keep your audio files between -9 db (inaudible) and -1 db (peak loudness). Anything higher will get clipped and may sound uncomfortable/painful.
  - Additionally, have a SoundManager class in your game that rate-limits audio so that if a sound gets played multiple times in a frame it won't sound awful. You can also use it to slightly randomize the pitch/speed of a sound each time it's played, to get a little more mileage out of a small number of sound samples
- **Music**
  - **LMMS** - https://lmms.io/download
    - A MIDI sequencer that can make passable music and chiptunes. Can be used to synthesize sound effects too.
    - Manual - https://docs.lmms.io/user-manual
- **Video**
  - **DaVinci Resolve** - https://www.blackmagicdesign.com/products/davinciresolve
    - Video editor tool decent for making trailers. Has a free version that's a little more manageable than Blender's video editing tools.
    - Manuals and training videos - https://www.blackmagicdesign.com/products/davinciresolve/training
  - **GifCam** - https://blog.bahraniapps.com/gifcam/
    - Great tool for grabbing short video clips, recording them as GIFs, editing them, and adjusting their frames/colors for posts to the web.
    - Avoid anyone rehosting anything they claim is GifCam. It's known to be a vector for malware. Use the Bahrani Apps link.
  - **Windows Game Bar** - https://www.microsoft.com/en-us/windows/tips/game-bar
    - Records the screen in Microsoft Windows
    - On Windows press Win+G to bring up the overlay. You can use the Capture dialog to record footage and audio. It's stored to your User/Videos/Captures folder.
  - **Microsoft ClipChamp** - https://apps.microsoft.com/detail/9p1j8s7ccwwt
    - A very simple tool for cutting and splicing video
  - **OBS** - https://obsproject.com/
    - If you're using Linux, this is probably your best option for recording screen footage of your game.
- **Publishing/Portfolio**

    - **Itch.io** - https://itch.io/
        - Hosts a nice semi-professional homepage for your videogame and your own portfolio. Lets you release games for free, pay-what-you-want, or for sale. Great for game jams.
        - I strongly recommend making WebGL projects, as they easily play in the browser.
- **Text Editing**
    - **Google Docs**
        - https://drive.google.com/
        - Great for collaborating and sharing notes and design docs. You can all write in the same document at the same time.
        - Part of Google Drive and GMail, which is decent for sharing assets with 15 GB of storage on the free tier.
    - **Notepad++**
        - https://notepad-plus-plus.org/
        - An excellent tabbed text editor with some extra coding features.
- **Project Planning**
    - **Trello** - https://trello.com/
        - Lets your team put tasks on notecards, attach assets to them, and move them through development steps to make sure nothing is forgotten. Very simple and great for quick projects like game jams.
        - Free trial for 10 boards with unlimited cards, 10 MB attachment limit
- **Source Control**
    - You MUST use version control if you plan to finish and ship a big game project! Dropbox or Google Drive are not good enough! Version control is difficult to learn, but it keeps you backed up, lets you roll back mistakes, and makes it easy for multiple people to work on the project together.
    - One day you will lose months of work, and then you will know why version control and backups are so important.
    - **Git Version Control**
        - https://github.com/ - GitHub is a commercial service that uses the open source Git version control software. Its GitHub Desktop client is decent. You can make repositories private, but be careful putting any privileged private keys or passwords or Android keystores into it.
            - Setting up your own Git server is an intermediate-to-advanced-level engineering challenge (and it costs money). If you're a beginner, it's probably best to use GitHub.
        - Beware of working on large art assets with Git. They take up a lot of space and can make the repository slow. The Git LFS plugin can help a little bit with this problem, (but most services charge for it).
    - **SourceTree**
        - Good UI. Has some very annoying bugs if you're not on a major host like GitHub or Atlassian.
    - **Perforce**

    - Has good Unreal Engine integration
- **Non-Software Assets:**
  - https://kenney.nl/ - Extremely useful 2D assets (and some 3D) with source files for further modifications.
  - https://incompetech.com/music/royalty-free/music.html - The musical soundtrack of the Internet.
  - https://www.fontsquirrel.com/ - Default fonts make you look like an amateur. Read the licenses carefully for software use.
  - https://freesound.org/ - Quick sound effects. If you can't find the right sound, use Audacity to layer multiple sounds and change their pitch.
  - https://jfxr.frozenfractal.com/ - Quick and dirty sound effect generator. Remember to layer sounds for a more pleasant texture.
  - https://opengameart.org/ - A large database of free game assets. Read the licenses carefully for commercial use. Try to track down the sources of reposted/repackaged assets, they might be shared incorrectly.
  - https://assetstore.unity.com/?price=0-0&rating=4&orderBy=3 - Free Unity Store assets that require a login. Most art assets will work in any game engine. Pay attention to the licensing listed on each asset.
  - https://www.fab.com/search?ui_filter_price=1&is_free=1 - Free Unreal Engine Marketplace assets that require a login. Most art assets will work in any game engine. Pay attention to the licensing listed on each asset. Some assets have multiple licenses, some of which are paid.
- **Tips For Your First Game:** Thanks for reading this far! For new devs, I recommend making your first game using **Godot Engine**, exporting it to WebGL (web browsers) with support for keyboard/mouse controls, and publishing it to **Itch.io**.
  - Use the engine's provided demo/sample projects, swapping out the assets and altering the code a little bit to get it to behave how you want.
  - Keep it single-player and no larger than a single level, something that will take no more than a month to finish.
    - This will keep your scope small. You can take your time to learn the tools, finish and publish, and not get overwhelmed. There's a lot of tools to learn, so just try to focus on one thing at a time!
    - Arcade-style one-screen games have a lot of useful design constraints for first-time devs. It's a lot easier to make this kind of game fun for your players.
  - After publishing, show it to your friends, pay attention to their feedback, and then do a round of revisions on it. Things that were minor problems for you may be painful for them (time to fix some bugs!) They may find it difficult to understand how to play (time to add a tutorial!) They may like it and wish it was longer (time to add more levels!)
    - I have found that when taking feedback, people are good at pointing at things they don't like, but they're not as good at proposing ways to fix those things. Don't take feedback at face value, and don't try to please

everyone. Stick to your guns on your design and creative decisions unless it’s clear that *everyone* hates it.

- Write down all of your ideas! You may not have time to make them now, but you can come back and revisit them later when you’re feeling uninspired or have another idea that connects well with it. If an idea captures your imagination many times, you should probably make that your next project.