Getting Started With Game Development

Version 1.0

So, you want to create your own games but have no idea where to get started? You're not the only one; that's why this guide has been created. There is a lot of information available that covers every aspect of game development, but it's hard to get started at the absolute beginning. Let us guide you through the very first steps into game development!

First and foremost, you'll have to choose a game engine to get started with. A game engine is the software on which developers build and run their games. Most engines offer versions suited for personal or hobby use, which are perfect for new developers to get started with. Each one greatly differs in features and ease of use.

Level 1. Choosing the right tools

There's a huge selection of game engines currently available. Some of the most popular out there are Unity, Unreal Engine and CryENGINE. These engines get used by both small independent developers and large triple-A studios. However, these tools can be quite hard to understand and require a solid grasp of a programming language.

Easier tools are available and most of these support visual programming. Visual programming is a method that enables non-programmers to visually define behaviors in their games using drag-and-drop interfaces. The best engine for you is completely dependent on the features you want or the time you can invest in learning.

Easy to learn (visual programming):

- **Construct 2** (2D) is an extremely easy to use game engine. It uses a visual programming language which means there's no steep learning curve and you can make your first game within hours. Check out the Beginner's Guide to Construct 2

- **Game Maker: Studio** (2D/3D) is a flexible game creation tool for beginners, offering both a drag-and-drop interface as well as its own scripting language known as GML. Learn How to Use GameMaker: Studio on their official website.

- **Stencyl** (2D) is similar to Construct 2 but instead is powered by Flash technology. It allows for easy exporting to various mobile platforms. The Indie Station video tutorials are a great start.

Require learning a programming language:

- **Unity** (2D/3D) is one of the most used game engines by both indie developers and triple-A game studios. It can however be hard to get into. Unity supports both JavaScript and C# for programming. A good place to get started with Unity is on their official website. Getting started with Unity

- **Unreal Engine** (2D/3D) is an engine that aims to deliver state-of-the-art graphics. It contains easy to use tools for various tasks and contains a visual programming language (node based).

- **LÖVE** (2D) is an open source game engine that uses the Lua language and is really easy to get into. It doesn't offer an IDE however, so everything is handled by code.

Specific purpose:

The following tools serve a specific purpose and might not fit the type of game you want to create, they do however provide an excellent environment for practicing and learning.

- **RPG Maker** (2D) is limited in the type of game you can create but it allows for great practice in storytelling and world building. It supports both a proper programming language and visual coding.

- **Scratch** (2D) is an educational tool aimed at younger developers. It uses a visual programming language and their website contains hundreds of sample games. The tool can be very limiting however.

Most of the engines above offer an easy way to get started using visual programming. Some, however, require knowledge of a specific programming language. For a complete list of game engines head over to PixelProspector.

While there are hundreds of languages out there, the most commonly used are C# and C++. These can be hard to get started with so it's advised to first get the hang of an easier language first. Lua, Python or JavaScript are all great starting languages. Once you have a basic mastery of programming with one language it becomes much easier to get acquainted with the others, as the concepts can easily "translate" between languages.

Level 2. Resources

While it's one thing to learn a programming language, it's a whole different story to gain artistic skills. It's another aspect of game development, and while some developers can churn out some pretty decent graphics, others have trouble creating even the most simple of sprites.

For the beginner, the solution is simple: pre-made game assets. There's a whole treasure trove of game assets available that are just waiting to be put together and programmed.

At Kenney.nl you'll find over 19,000 different game assets which cover a huge array of game types! Most of the asset packs are aimed specifically for beginner game developers and contain every graphic required to create a specific game. Included is a shooting gallery, a jumping game, and a flying game similar to Flappy Bird. These assets are all in the public domain and have no requirement for their use.

Still can't find what you're looking for? Take a look at this page, which lists dozens more sources for free game assets and tools. OpenGameArt.org also has a great collection of freely available game assets, do take note of the licenses though.

Be absolutely sure you are familiar with your rights with any asset you download online! Some licenses for game assets require attribution if they are used in a project. If an author requests attribution, you have to put their name in the credits. Public domain assets like the ones at Kenney.nl require no attribution. Many authors use Creative Commons licenses to make it very clear what rights they give to you, so keep an eye out.

Not completely sure about the license? Ask the author!

Level 3. Your first game

For your first game, it's important to start simple. It's likely that you'll be learning your tools as you're starting your first project, so stick to what you know! Even the masters of game design started out re-creating arcade classics like PAC-MAN or Space Invaders. Take a simple concept or idea, and develop that into a theme. It doesn't matter what it is, as long as it's fun!

Here's a short list of game genres in order of simplicity to create prototypes for. If you can make games from at least three genres on this list, you should be good to start experimenting yourself.

1. Shooting gallery
2. Racing game
3. Top-down shooter
4. 2D Platformer
5. Color-matching puzzle
6. Card game

Learning new skills and moving onto more intricate projects is mostly done through trial and error. Figuring out how to create a UI, menu, keeping score and saving games are all methods you'll learn along the way.

Level 4. Gaining feedback

It's important to gain honest feedback on your first few projects. This feedback will help you improve your skills and create better games in the long run. Feedback also helps build friendships in the community. Utilize all social media outlets to get the word out there that your game is ready to be played! If you're using software that allows for in-browser playback (like Construct 2 and Unity) you can upload your game to sites like Kongregate and Newgrounds.

There are plenty of friendly gaming communities where you can show off your game and ask for feedback. Check if the game engine you're using has its own community, these people can often help you with more technical problems and advice.

For a global community of game developers have a look at Reddit /r/gamedev. Over 140,000 game developers have gathered there to discuss topics and give insight on the gaming industry.

Level 5. Getting your jam on

Game jams are gatherings of game developers/artists where a game or prototype is created within a short span of time, usually between 24 and 72 hours. Most game jams are centered around a theme, which all games developed within the jam must adhere to. These gatherings are a great way to test your skill and get acquainted with other developers.

While there's a huge number of game jams, some of the larger ones are Ludum Dare and Global Game Jam. The latter is great because it's held all over the world at over 400 locations. There's a great chance there might be a venue near you!

Level 6. Your next project

Now that you've created a few prototypes, and possibly even a small game you're ready to tackle your next project;

6a. Concept

Think up a concept for your game, a great starting point might be taking a core mechanics of an already existing game and further building upon that. Write down the complete idea and each of the features that'll be included, the game design document will make sure it won't end up in development hell and keeps you focused on the core gameplay.

6b. Development

Keep it simple and don't try to create an engine or framework for everything. It's quite normal that you'll have to create some wonky snippets of code to get everything working. It's your first game, don't make it too hard on yourself.

6c. Release

The release of your game differs depending on the platform you've chosen to create your game for. Mobile platforms like Android and iOS offer a storefront which is open to all developers and user friendly. Picking a game engine that allows exporting to HTML5 will make it easy for you to get the game published on the web.

6d. Marketing

> Don’t skip this step! There’s no need for a big marketing budget to get a few players for your game. Tell your friends and publish your title on a few forums or communities. Humbly asking for an opinion on your game will most often work better than trying to advertise your project as the next best thing.

Gather all the criticism you can, carefully take everything in consideration and use it to update the game or for your next game. The players of your game are your customers, so give them what they want.

Use this guide as a good suggestion where to start, every developer has their own ways of doing things and there’s no wrong or right way.

Level complete. Footnotes/credits

Guide written by:
Kenney Vleugels (mail@kenney.nl)

Co-written by (random order):
Sam Baylus, Dan Relluchs, Liam Sauvé, Charlie Geiger and Ian Martin.

Game engines:
Construct 2 (2D)
Game Maker: Studio (2D/3D)
Stencyl (2D)
Unity (2D/3D)
Unreal Engine (2D/3D)
LÖVE (2D)
RPG Maker (2D)
Scratch (2D)

Resources:
Kenney.nl (Assets)
OpenGameArt.org (Assets)

Game portals:
Newgrounds.com
Kongregate.com

Communities:
Reddit (/r/gamedev)

Game jams:
Ludum Dare
Global Game Jam