Game Development:

Where to begin

Written by Rachel J. Morris, August 2009
www.moosader.com

Table of Contents

Introduction

So... Game Programming, eh? We all love video games, and we all fantasize at some point of making something, amirite?

This tutorial is to help out those who don't know where to begin. Maybe you've done some programming for school, or maybe you've done no programming, but you want to make some sort of games.

It's hard to know where to start, especially when you've never had programming experience. Do you want to design the games? Do you actually want to code? Is it something you want to pursue as a career? How much time does it take to learn a language well enough if you do decide to dive into the coding side of things? I'm going to try to answer all these questions with this guide.

If you have any questions, feel free to contact me at
RachelJMorris@gmail.com

The Different Roles in Game Development

Many people want to make games, but don't have a clear idea of what roles are needed in making a game. In the beginning, one can generally do all the jobs, or find one or two other people to team with.

The main focus of this section is to figure out what it is you want to do, so you can start learning the proper tools for the job.

Programmer

The Programmer is the one who handles computer code. It comes with varying degrees of difficulty depending on the language and libraries or engines being used. Programming is a lot of work, and it takes a lot of practice to be able to think about what needs to be done in a coding sense.

Programming is not just waving a magic wand and a game appears; you have to understand that you essentially have to tell the program how to do everything. Everything! How a character moves when it walks, you have to tell it how animating works, how the physics works, how it loads in maps, how it draws maps.

Artist

The artist, or artists, will handle in-game graphics, concept art, or even just drawn artwork for promoting the game.

For 2D games, art will usually consist of the character and item sprites, world tiles, and graphical user interface. For 3D, it would consist of creating textures, models, and also the graphical user interface.

Designer

The designer plans out the game's elements. They might design the storyline, gameplay, levels, puzzles, etc. Preferably they should know about both programming and art, but don't need to be a programmer or artist themselves. Design is what most people want to do, ie. “I have a neat idea for a game!”.

So do you want to actually program, or are you more into design? This is something to ask yourself. And if you haven't programmed (and therefore don't know), you might try an easy language out. More on this in the next section.

Languages and Engines – What is your aim?

First off – prior programming experience. Got any?

If you've used C++, Java, or C# in the past and are comfortable with it, then I suggest you start with those languages for game programming. Yes, you can make more than DOS-prompt-type programs with it. There are libraries for each to handle the lower level graphics, sound, input, etc. for you for each of these, and Object Oriented Languages are the best to make games with.

If you have coded, but not in a object oriented language, you might try out an Object Oriented Language, or you could stick with what you have (provided you're not “programming” in “HTML”), or you could try one of the languages that are easier for making games.

If you've never coded in your life (or, again, have only used HTML, which is not a programming language), then it's a little trickier. You don't know if you want to code or not, so you don't want to just start with C++ and decide you hate code and just wanted to design a game anyways.

If you're curious about coding, but not quite confident enough to tackle one of the harder languages, or don't really have any reason to learn an Object Oriented Language anyway (career, school), you could try one of the easier languages.

If really all you want to do is design the game and not touch code, you'll probably want to look into the engines. Even some of these, however, require scripting programming, but scripting is a lot easier than pure coding.

Keep in mind, though. If ***game programming*** is something you want to pursue as a career – go with Object Oriented Languages. That, or start out with an easier language and work your way up later on. There are plenty of Indie developers who are known who use engines and don't write a line of code, but if you want to be a *professional* and work for a big company (as a programmer), you'll want OO (object oriented).

Check out the list of engines and languages below for a brief description of features, and list of games made with them.

Engines

Multimedia Fusion

http://www.clickteam.com/website/index.php
$119

Games made with MMF or MMF2:

I want to be the guy	http://kayin.pyoko.org/iwbtg/
Macarena of the Missing	http://realnoyb.googlepages.com/macarenaofthemissing
Noitu Love	http://www.konjak.org/

Game Maker

http://www.yoyogames.com/make
Lite version free, Pro edition $25

Games made with GM:

Iji	http://www.remar.se/daniel/iji.php
Mondo Medicals	http://www.cactus-soft.co.nr/
Ninja Loves Pirate	http://www.ninjalovespirate.com/

Adventure Game Studio

http://www.adventuregamestudio.co.uk/
This is an engine for making point-and-click adventures, very similar to the old LucasArts and Sierra games.

Games made with AGS:

5 Days a Stranger	http://www.fullyramblomatic.com/games.htm
A tale of Two Kingdoms	http://crystalshard.net/atotk.php

RPG Maker 95, 2000, 2003, XP, or VX

VX: http://www.rpgmakervx.com/
XP: http://tkool.jp/products/rpgxp/eng/
Free to use, buy a license in order to sell
I don't think you can get 95, 2000, or 2003 legally,
they weren't released in America, just translated by hobbyists.

Sample Games:

Blender

http://www.blender.org/
Free

Games made with Blender:

- ?????????????

Panda3D

http://www.panda3d.org/
Free

Games made with Panda3D:

- Disney's ToonTown http://play.toontown.com/test/browserTest.php?r=967104

Easier languages

BlitzBasic / Blitz3D

http://www.blitzbasic.com/
Free to try, $80 to buy

Games made with BlitzBasic / Blitz3D:

- ?????????????

DarkBasic

http://darkbasic.thegamecreators.com/
$40

Games made with DarkBasic:

- ?????????????

DarkGDK with C++

http://gdk.thegamecreators.com/
Free

Games made with DarkGDK:

- ?????????????

Flash

http://www.adobe.com/products/flash/?promoid=BPDEE
$699 for CS4.
Check amazon for older versions

Games made with Flash:

- ?????????????

Python

http://www.python.org/

Free

+ Lots of engines use Python for scripting

Description

Games made with Python:

Frets on Fire	http://fretsonfire.sourceforge.net/
Plague	http://plague-like.blogspot.com/

More advanced / the object-oriented languages

All these languages are free, and so are most libraries and engines to go with.

C++

There are honestly so many extension libraries and engines for C++, it's hard to list them all. You'll have to do a Google search if you want something different, but here are some of the more well-known ones.

Library vs. Engine

[Why you'd want to use a library vs engine]

Libraries, SDKs, APIs

Allegro

http://www.talula.demon.co.uk/allegro/
Allegro is a pretty straight forward, easy to learn game programming library you can use with C++. It mainly handles 2D stuff (There IS a 3D component, but nobody in their right mind will suggest it to you), and you can also use it along side OpenGL.
Allegro is "giftware", which essentially you can do whatever you want with it

and you don't owe anybody anything for it.
Allegro is also cross platform.

ClanLib

http://clanlib.org/
I've never used ClanLib before, so I can't give you a personalized review of it. It seems pretty similar to Allegro or SDL, and also appears to support using it with OpenGL, has Socket support, and a GUI framework.
ClanLib is under the BSD license.
ClanLib is cross platform.

SDL

http://www.libsdl.org/
SDL (Simple DirectMedia Layer) is similar to Allegro, but slightly more difficult but also more powerful. Besides the basic graphics, input, and sound handling, it has functions for multithreading, and an extended library for sockets. It can also be used along side OpenGL, which many people utilize.
SDL is under the LGPL license - http://www.libsdl.org/license-lgpl.php
SDL is cross platform.

OpenGL

OpenGL

DirectX

DirectX

Engines

Crystal Space

http://www.crystalspace3d.org/
Crystal Space is a 3D engine [don't know what else to put]
Crystal Space uses the LGPL license
Crystal Space is cross platform.

DarkGDK

http://gdk.thegamecreators.com/
DarkGDK allows you to make 2D and 3D games, and use many features like Shaders easily, [don't know what else to put]
DarkGDK costs $30 for a commercial license

Irrlicht

http://irrlicht.sourceforge.net/

Irrlicht is a 3D engine with support for shaders, materials[don't know what else to put]

Irrlicht uses it's own license, based on the zlib/libpng: http://irrlicht.sourceforge.net/license.html

Ogre3D

http://www.ogre3d.org/

Ogre3D is a 3D engine with support for shaders, materials, meshes, animated models, scene management, and other features.[don't know what else to put]

Ogre3D is cross platform.

Ogre3D uses the LGPL license.

Games written with Allegro:

- Icy Tower http://www.allegro.cc/depot/IcyTower/
- Zep's Dreamland http://www.zepsdreamland.com/

Games written with ClanLib:

- ?????????????

Games written with Crystal Space:

- ?????????????

Games written with DarkGDK:

- ?????????????

Games written with DirectX:

- Gears of War http://gearsofwar.xbox.com/
- Crysis http://games.ea.com/crysis/
- BioShock http://www.bioshockgame.com/

Games written with Irrlicht:

- ?????????????

Games written with Ogre3D:

- ?????????????

Games written with OpenGL:

- Quake 1, 2, and 3 http://www.idsoftware.com/
- America's Army http://www.americasarmy.com/
- Half Life http://www.valvesoftware.com/

Games written with SDL:

• World of Goo	http://www.worldofgoo.com/
• Second Life	http://secondlife.com/
• Secret Maryo Chronicles	http://www.secretmaryo.org/

Java

Description

Libraries

Engines

Games made with Java:

- ?????????????

C#

Description

Libraries

Games made with C#:

- ?????????????

But I just wanna make an MMO!!

If all you want to do is make an MMO right off the bat and not really much else, I'd advise that you don't start with C++. So many people decide they want to make an MMO and want to start with some complex language and have no idea how incredibly hard it is.

Programming a multiplayer online game is so difficult and tedious – not only do you need a language, but you need to know sockets intimately, and have to have a database system – you should just use a pre-made engine.

Directory of MMO-makin' programs!

http://board.moosader.com/viewtopic.php?f=10&t=48

Resources

I've moved the various lists of tools, engines, languages, and such to my message board in order to keep it up-to-date easier, and more open to suggestions

The directory forum is here: http://board.moosader.com/viewforum.php?f=10

Tutorials, Guides, and such

http://board.moosader.com/viewtopic.php?f=10&t=49

Communities

http://board.moosader.com/viewtopic.php?f=10&t=37

Also, a lot of engines have their own message boards.

Tools

Need some free tools to help make games, but don't know where to look?

Directory of IDEs for various languages

http://board.moosader.com/viewtopic.php?f=10&t=31

Libraries and Engines for C++

http://board.moosader.com/viewtopic.php?f=10&t=32

Libraries for other languages

http://board.moosader.com/viewtopic.php?f=10&t=50

Music/Sound Editing

http://board.moosader.com/viewtopic.php?f=10&t=33

Graphics Editing

http://board.moosader.com/viewtopic.php?f=10&t=34

Map Editor applications

http://board.moosader.com/viewtopic.php?f=10&t=35

Public Domain Resources

Want to use original resources (eg, not ripped from games?) but don't have any creative peoples around to help you? Check some of the sites on this directory out!
http://board.moosader.com/viewtopic.php?f=10&t=36

Beginners beginning a project - Don't overdo it!

Planning your game

Everyone with experience always says this, and it cannot be stressed enough –

DO NOT PLAN COMPLEX GAMES RIGHT OFF THE BAT

You have to start off really slowly!
Don't plan on making something like Super Mario World – Too hard for novices!
No, you're not going to make Doom – Out of the question!
Final Fantasy? No. Starcraft? Nada. World of Warcraft? Nope. Street Fighter? Nuhuh.

Then what do I make?!!

Your first games should be more like old Atari and arcade games. Don't plan a lot of features, don't plan to have a ton of levels, don't even plan much of a story. You can work on these in time, but your **first games** should be **really basic**.

Try remaking Frogger, Pong, Space Invaders, Robotron... or Pickin' Sticks! All of these are top-down, so you don't really have to worry about physics (except some really basic stuff in Pong). AI is really basic (cars move from one side of the screen to the other, invaders move one way and then the other, or baddies move randomly). You don't need really fancy collision detection (Most of the characters were blocks anyway).

The bottom line is – **At the beginning, you're learning the basics of your engine, or of using graphics, sounds, input, regulating frames per second, and maybe loading levels in.** Focus on getting experience with your tools before you start trying something bigger.

A good thing to do is try to work your way up through the history of video games. Make Atari-style games first (usually one screen, basic gameplay), then move up to 8- and 16-bit style stuff (several levels, basic enemies, slightly more complex gameplay), and work your way up from there.

For the Object Oriented Languages, also keep in mind that, after Pickin' Sticks, Shmups are probably the next easiest type of game to make. Platformers are a bit more difficult because you have to deal with gravity, and RPGs are complex because you need a scripting language on top of what you're already coding with.

Real Time Strategy games are probably the most complex 2D type game you can make, because you deal with writing a User Interface, as well as many components of AI (Pathfinding, having the computer opponent decide how to move and not be stupid).

Teams

Teams aren't something you should really try to get together early on, either. Most of the time, it's hard to organize a group (or even you and one other person) to get their work done, and otherwise it's sort of overkill early on. You might get a friend to draw some sprites for you or write some songs for the game, and if you can't get a friend to do that, then you can look for Public Domain resources. But early on, having more than just you on a game will slow it down

Programming expectations and Common misconceptions

This guide is probably a bit of a downer. A realistic downer, however. Many many people believe programming is something that it's not, so I felt it necessary to almost hammer into the readers' (your) head that it ***isn't easy*** to program, or heck to even make a game with an engine.

Game developing takes a lot of **time** and **effort**. Not only do you need to code, but you need to gather resources such as graphics, music, and sound effects, build levels, plan out game mechanics, and essentially tell your language or engine ***how to do <u>everything</u>***. How do the physics work? How does the character jump? How does your program realize when one character hits another object?

Summary

So hopefully game making is a little more demystified. No matter what your goals for making games are, it's possible! You just have to keep in mind where you are, and where you want to go. Hobby? Career? Indie-hobby-career? Your call.

If you have any questions, or if I've typoed something or you otherwise want to comment on this tutorial, please email me at

RachelJMorris@gmail.com

And to check out my other tutorials, my games, or maybe my map editor, they're all accessible at

www.moosader.com

Thanks for reading!

Credits

Guide written / compiled by Rachel J. Morris

Useful information from the lovely peoples:

aamesxdavid
Ewan
JaxDragon
kamokow
Maevik
Trufun202
XianForce