

Navigation

Home
DirectX for Windows 8
 DirectX 11.2
 DirectX 11.1
DirectX for Desktop
 DirectX 11
 DirectX 9
Useful Resources
About DirectXTutorial
DirectXTutorial Premium
Testimonials

Contact

Contact me here:
Twitter: [@dastopher](#)
Email: chris@directxtutorial.com

Or leave me feedback:
[Quick Feedback](#)

Lesson 1: The Parts of a Game

[Next](#)

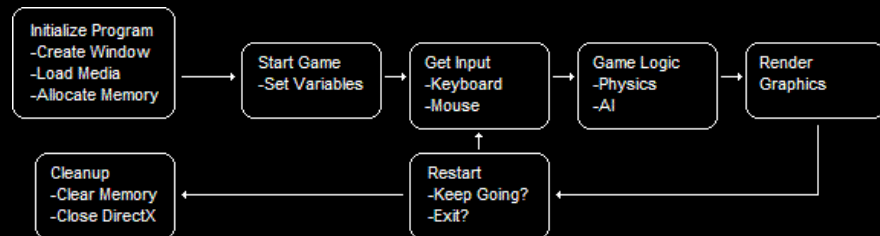
Lesson Overview

This lesson will cover the basic structure of a video game. It will give you some idea of where you are going with DirectX so you don't dive in blind and unguided, as many rookie game programmers do. As video games can get very long and complex, I suggest that even experienced programmers new to game programming at least skim through this so they know where they're going.

The Seven Phases of a Game

A video game is, in essence, a continuous loop performing various actions which, when run through once, bring about a single frame of animation. Programmers, when entering the world of game programming, rarely consider all the components required in even a simple animated game like Pong or Asteroids and can easily get overwhelmed. This means no more event-driven programs you're probably used to.

So let's start out by looking at each phase of a game, in sequence, from beginning to end and then describing each part of it.



The Sequence of a Computer Game

Phase 1: Initialize the program

Here is where you start the show. You create a window to display your game, you set up DirectX for use, load graphics, models and other media, allocate memory and so on.

Phase 2: Start the game

This part is basically setting up your game to be played. This is where you might select a map, set the location of your player (and anyone else in your game world) or set random values, such as random terrain, that are decided at the beginning of the game only. After all of this, you enter the game loop itself for the kickoff.

Phase 3: Get input from the player

Here, you get all the information from the keyboard, mouse, joystick, controller or whatever device your player is using. This phase is mostly covered in the DirectInput tutorial.

Phase 4: Run the game logic, such as physics and AI

In this part you process what is going on in your world. Where exactly did your player move in the last sixtieth of a second? How much ammo does he have left? Are his enemies coming or going? What about his allies? Did his ship crash into a mountain? Or was that just a cloud he just ran into? All this and more is determined in this phase. I don't have any tutorials on these yet, but I hope to get them up well before the end of the year, so stay tuned.

Phase 5: Render graphics

This is where DirectX is mostly used. Here you would process all your 3D (and 2D) graphics and render them to the screen. This will be gone over in the Direct3D tutorials.

Phase 6: Restart

Simply put, you go back to Phase 3 and do it all over again.

Phase 7: Cleanup

Called for memory? Now is your last chance to clear everything up, because the program is about to end. Also, DirectX calls for various interfaces to be shut down or “released”. These are all done here.

Combining the Phases and Adding More

Ok, so how does all this fit together? Well, you will notice that most of the above is done in a loop continuously until an exit is called for. The diagram you saw above shows how all the above pieces are put together.

Now, admittedly this is simplified, and more than somewhat, but I just want to give you a quick overview of things. You can (and will need to) add all kinds of details such as networking code, splash screens, an animated menu loop, sound, music, and so on. Some of these are rather easy. For instance, a simple splash screen is a piece of cake. You just add a window to the beginning of the program.

Although some are easy, others can get more complex, such as networking and keeping multiple computers (that run at different speeds) moving through the game at the exact same pace. (I'll go over this one in the Multiplayer tutorial, as well as give you a simple timing example in the Game Development tutorial).

But let's not concern ourselves with such things yet. We haven't entered in a single line of code and we're already talking about how complex things are. Let's get to the actual programming before we get into such topics.

Next we'll cover how Windows programming works, make a simple window, and program the basic game loop.

So even though we haven't programmed anything yet, we now know where we're going. And with that, let's dive right in!

Next Lesson: A Primer of Basic Windows

GO! GO! GO!