

## Building a Java First-Person Shooter

### Episode 0 [Last update: 5/03/2017]

These notes are intended to accompany the video sessions being presented on the youtube channel “3D Java Game Programming” by youtube member “The Chernob” at <https://www.youtube.com/playlist?list=PL656DADE0DA25ADBB>. I created them as a way to review the material and explore in more depth the topics presented. I am sharing with the world since the original work is based on material freely and openly available.

Note: These notes DO NOT stand on their own, that is, I rely on the fact that you viewed and followed along the video and may want more information, clarification and or the material reviewed from a different perspective.

The purpose of the videos is to create a first-person shooter (FPS) without using any Java frameworks such as Lightweight Java Game Library (LWJGL), LibGDX, or jMonkey Game Engine. The advantages to creating a 3D FPS game without the support of specialized game libraries that is to limit yourself to the commonly available Java classes (not even use the Java 2D or 3D APIs) is that you get to learn 3D fundamentals.

For a different presentation style that is not geared to following video episodes checkout my notes/book on “Creating Games with Java.” Those notes are more in a book format and covers creating 2D and 3D games using Java in detail. In fact, I borrow or steal from these video episode notes quite liberally and incorporate into my own notes.

### Prerequisites

You should be comfortable with basic Java programming knowledge that would be covered in the one-semester college course. If you are not then I recommend the following websites for tutorials and online textbooks:

- ✚ Java Language and Virtual Machine Specifications at <http://docs.oracle.com/javase/specs/>
  - This is not intended to be used by those who want to learn Java but more as a reference or to lookup of areas you are not familiar with once you have gone through a course or one of the books suggested in this list.
- ✚ Thinking in Java - <http://www.mindviewinc.com/Books/downloads.html>
  - This too is less of a “learn Java” book but is a great book to read while learning Java to get into the Java frame of mind
- ✚ Introduction to Programming Using Java, 6<sup>th</sup> Edition, <http://math.hws.edu/javanotes/>
  - This is an excellent free e-book that could be used in any college course

You should also know your way around the Eclipse IDE. If you are not familiar with Eclipse then I recommend the following links for tutorials and information:

- ✚ Eclipse IDE Tutorial: <http://www.vogella.com/articles/Eclipse/article.html>
- ✚ Video tutorials: <http://eclipsetutorial.sourceforge.net/totalbeginner.html>



I think many youtube programming video presenters failed to identify the starting level or prerequisites required to view and learn from their videos. This presents a problem for the viewer since the presenters discuss at length topics the viewer should clearly be comfortable with and not enough on the topics that may be new even after a one semester Java course. The creator of these videos makes the same mistake – not deciding and stating what the minimum level of experience should be for the average viewer and then adhering to that assumption. It can be irksome for developers with experience who just want to get to the heart of the Java classes to have the presenter spend too much time on simple concepts and disconcerting for the novice since the very brief discussion on arrays, for-loops is not enough time to solidify these concepts. Setting the expectation that viewers can learn Java at the same time as learning the concepts involved in creating a 3D game is not realistic and just demoralizes the average person (proof is the number of people who view the first video but don't get to the second or the third!) Of course it is surprising how smart and hardworking many of the viewers in pursuing and understanding of the material and I hope these notes can assist in that journey. I strictly follow the discussion in the videos.

### Lessons/Episodes

Each of the major video episodes will have a corresponding pdf to explain independently of the video how the code works and more details on the Java libraries used to accomplish the graphics. I also added additional notes on other related topics (e.g. Java Graphics Programming, Random Dungeon Generator, etc.). I felt the explanation offered in the videos and just blind use of Java classes without an explanation of their purpose and function in the final version of the code is very frustrating in the event that something does not work or if I truly plan on building my own 3D game. I hope these notes provide more of an explanation to the Java classes that are used to build a 3D world and prove useful to others.

The total number of videos is 39.

- ✚ Episode 0 – [non-video] These notes
- ✚ Episode 1 – Window
- ✚ Episode 2 – Game Loop
- ✚ Episode 3 – Arrays
- ✚ Episode 4 – Drawing Pixels
- ✚ Episode 4.5 – How Rendering Works
- ✚ Episode 5 – Playing with Pixels!
- ✚ Episode 6 – Performance Boosting
- ✚ Episode 7 – FPS Counter

## 3D Java Game Programming – Episode 0

- ✚ Episode 8 – Alpha Support and More
- ✚ Episode 9 – Beginning 3D
- ✚ Episode 10 – Floors and Animation
- ✚ Episode 11 – Rotation
- ✚ Episode 12 – User Input
- ✚ Episode 13 – Render Distance Limiter!
- ✚ Episode 14 – Basic Mouse Movement
- ✚ Episode 15 – Textures + More!
- ✚ Episode 16 – Walking, Crouching, Sprinting + More
- ✚ Episode 16.5 – Exporting Runnable Jars
- ✚ Episode 17 – Small Adjustments + Birthday!
- ✚ Episode 17.5 – Creating an Applet
- ✚ Episode 18 – The Beginning of Walls
- ✚ Episode 18.1 – A Few More Things
- ✚ Episode 18.5 – Creating an EXE File in Java
- ✚ Episode 19 – Rendering Walls
- ✚ Episode 20 – Continuing Walls, Fixing Bugs, and M
- ✚ Episode 21 – Texturing Walls, Fixing Clipping and
- ✚ Episode 22 – Random Level Generator + Properly F
- ✚ Episode 23 – Graphical User Interface (GUI) Launching
- ✚ Episode 24 – Making Out Launcher Work
- ✚ Episode 25 – Writing and Reading Files
- ✚ Episode 26 – Custom Resolutions
- ✚ Episode 27 – Decorating the Launcher
- ✚ Episode 28 – Continuing our Custom Launcher!
- ✚ Episode 29 – Launching the Game
- ✚ Episode 30 – Colour Processing In-Depth
- ✚ Episode 31 – Sprites!
- ✚ Episode 32 – Sprite Mapping
- ✚ Episode 33 - High Resolution Rendering
- ✚ Episode 34 – Entities
- ✚ Episode Ray casting – [non-video] My personal notes on creating a real 3D FPS from beginning to end.

### Format of the Episodes

Every episode that covers a video will start off with **objectives** of the video, followed by the direct **URL** to the video episode and then finally a **discussion** that covers the material in the video episode. At the end the FINAL episode **code** will be given if it is not already shown in the notes.

We will leave out any basic Java programming discussion from these notes that come up in a first semester Java programming course. That is, we will not go over the use of `int` or `main` function's purpose, etc. We will assume that you are familiar with these concepts. I do make an exception in

Episode 3 in discussing arrays a bit more than I should but I thought it would benefit those not too comfortable with this data structure since it is the key representation of the screen.

## Ray casting Game



You may be disappointed to learn that by the end of the video series you will NOT have a 3D First Person Shooter. The tool we build is a limited form of a ray casting engine. These engines were used in the construction of 3D games during the 1990's. The best example is the Id game Wolfenstein 3D released in 1992. The graphics are rather simple where the 3D scene is shown on a flat plane dividing area and rooms by a grid-based pattern of walls and doors – all the same height.

**Ray casting** is the most basic technique to render 3D scenes using algorithms that use geometric algorithms of **ray tracing**.

Most of the material that follows comes from the web site: <http://lodev.org/cgtutor/raycasting.html>.

Ray casting is a rendering technique to create a 3D perspective in a 2D world. The reality is that all graphics algorithms that draw to our monitor, TV or mobile device are actually rendering a 3D world on a 2D plane. Ray casting was used in the 1990's to create video games that displayed 3D scenes. The ray tracing algorithm allowed the slow computers of that era to support the drawing of 3D scenes. The most popular game that used this technique was Id's game Wolfenstein 3D.



Figure 1 - Wolf3D Map editor

The ray casting engine in the game Wolfenstein 3D was rather limited: all the walls have the same height and go from the floor to the ceiling. The game designer would use a map editor to create the level layout. Notice that all the walls are orthogonal (at right angles) squares that you build within a 2D grid.

The Id engine was limited and could not display stairs, allow the user to jump or display height differences.



Figure 2 - Doom scene- notice the floor and ceiling

The engines used in later games such as Doom and 3D Realms - Duke Nukem 3D used ray casting but were able to support sloped walls, different heights, textured floors and ceilings, transparent walls and more. The sprites in the game representing the enemies, object and goodies were all 2D images.

## 3D Java Game Programming – Episode 0

The last non-video episode titled “Episode Ray Casting” are my notes on creating a real 3D ray casting engine where a world map is created using a map editor, where you have creatures and objects in the scene. The writing assumes you read the video episodes up to Episode 12.

## References

1. TheChernoProject. <https://www.youtube.com/channel/UCQ-W1KE9EYfdxhL6S4twUNw>. 2017.
2. Computer Graphics Tutorial. [https://www.tutorialspoint.com/computer\\_graphics/index.htm](https://www.tutorialspoint.com/computer_graphics/index.htm). 2017.
3. LaMothe Ratcliff, Seminatore & Tyler. Tricks of the Game Programming Gurus. Sams Publishing. 1994. ISBN: 0-672-30507-0.
4. Lode's Computer Graphics Tutorial. <http://lodev.org/cgtutor/raycasting.html>. 2007
5. Oak, Scott. Wong, Henry. Java Threads: Understanding and Mastering Concurrent Programming. 3<sup>rd</sup> Edition. O'Reilly. 2004. ISBN: 978-0596007829