Setting Up LWJGL in Eclipse

In this short tutorial we will setup the Light-weight Java Game Library (LWJGL) in the Eclipse IDE. We will add the relevant JAR files as a user library, which can be added to any LWJGL project.

Mac users: note that there are fixes to common problems! Please read those.

Step I: Adding LWJGL as a User Library

- 1. In the site https://www.lwjgl.org/customize , choose the following build, and press "Download ZIP":
 - a. ZIP Bundle
 - b. Without source, JavaDoc and build config
 - c. Choose your OS (usually x64)
 - d. Custom present, including: Assimp, GLFW, JAWT, OpenAL, OpenGL, stb
 - e. No Addons
 - f. Version 3.3.3 (newest, as of December 2023)

Mode	Presets	Contents
ZIP Bundle	O None	
Maven	Custom	🗹 Assimp
Gradle	 Everything 	bgfx
	 Getting Started 	CUDA
	 Minimal OpenGL 	EGL
Options	 Minimal OpenGL ES 	FMOD
Include source	 Minimal Vulkan 	FreeType
Include JavaDoc		SLFW
Include build config	Addons	HarfBuzz
	JOML v1.10.5	hwloc
Natives	LWJGLX/debug v1.0.0	🗹 JAWT
🔲 💩 Linux x64	LWJGLX/lwjgl3-awt v0.1.8	🔲 jemalloc
A Linux arm64	steamworks4j v1.9.0	KTX (Khronos Texture)
Δ Linux arm32	steamworks4j-server v1.9.0	🗋 libdivide
🔲 🗯 macOS x64		
🕝 🧯 macOS arm64	Version	
Windows x64		🗆 LZ4
🔲 🖷 Windows x86		Meow hash
🔲 🖷 Windows arm64		meshoptimizer
	○ 3.3.0	NanoVG & NanoSVG
	0 3.2.3	Native File Dialog

2. Extract the ZIP contents of LWJGL into your Java workspace (or any other location which you are sure will not get deleted).



3. Enter the Eclipse global settings:

a. On Windows & Linux: Window \rightarrow Preferences



b. On macOS: Eclipse \rightarrow Preferences



4. In preferences, go to: Java \rightarrow Build Path \rightarrow User Libraries



5. Press on "New..." and call the new library "LWJGL3":



6. Choose "Add External JARs..." and load all *.jar files from the extracted zip.



7. Now the JARs are added. You may press on "Apply and Close":



Step II: Importing a project that uses LWJGL3

- 1. Clone the project into your Java workspace.
- 2. Open the project as usual (File \rightarrow Open Projects from File System...)
- 3. Add LWJGL3 to the build path:
 - a. Open the Project's Properties (Right click on project -> Properties)
 - b. Under Java Build Path, select the "Libraries" tab

	Properties for pacman-task1-MichalKleinbort	
	Java Build Path	(⊃+);
> Resource Builders	🕮 Source 😥 Projects 🛋 Libraries 🍫 Order and Export 😡 Module Dependencies	
Coverage	JARs and class folders on the build path:	
Jano Bolid Path. Java Code Style Java Compiler Java Compiler Javadot Location Projet References	 Symptotic System Ultrary (JRE (17.09)) Classpath 	
		Apply
?	Ca	Apply and Close

c. Select "Classpath" and click the "Add Library" button. Add LWJGL3 (under "User Library"):



• • •	Add Library	
User Library Select a library to add to the classpath.		à
User libraries:		
M LWJGL3		User Libraries
? < Back	Next > Canc	el Finish

d. Finally, press "Apply and Close"

• • •	Properties for pacman-task1-MichalKleinbort	
	Java Build Path	⇔ • ⇔ •
Ananoras Banidors Coverage Git Buildors Lova Carda Style Java Carda Style Java Carda Style Java Carda Style Java Carda Style Java Carda Style Style	Source Projects All Literative Op order and Export Module Dependencies JABs and class folders on the build path. > Module Coopendencies > > > > <	Add JAR Add External JAR Add Ustable Add Ustable Add Datarral Gass Folder Edd Edd
0		Migrate JAR File

Common Issue: Wrong Binary on Apple Silicon

Usually, software on Apple Silicon is compiled as arm64 (and not x64). But depending on the binary of the JDK you downloaded, you may need to acquire the x64 build of LWJGL, if you see the error bellow:



Common Issue: -XstartOnFirstThread

You may see the following error (regarding -XstartOnFirstThread).



In such case, you need to indeed add that argument, as follows:

- 1. Right click the project, and choose "Properties"
- 2. In Run/Debug Settings, double click the main executable (the *.java file containing *public static void main(...)*)



3. In the "Arguments" tab, add "-XstartOnFirstThread" in "VM arguments":



Note: if your project has more than one main executable, you will need to do this for each one.