

JOLT Physics Engine



GODOT

Game engine

vs.

lib
GDX

Desktop and Web Performance Tests

v0.1

Test Setup

My System Info:

Windows 11 (build 26100)

NVIDIA GeForce RTX 3050 (NVIDIA; 32.0.15.7652)

AMD Ryzen 5 5500 (12 threads) and 32 GB memory

libGDX version: 1.13.1, using **gdx-teavm** for web builds.

Godot version: 4.5.beta1 using Compatibility renderer (GLES3)

What is being compared?

Comparing Xpe's Jolt Physics demo (on Windows desktop and the Web) performance of 3000 rigid bodies using both Godot and libGDX Jolt implementations. Link: <https://xpenatan.github.io/gdx-jolt/examples/samples/>

The Test

Source

Based on Xpe's Jolt Physics demos for **libGDX** using his **gdx-jolt** library.
This library is **teavm** compatible so **libGDX** can use it on web builds.

The specific test used from his samples is the **BoxSpawnTest.java**
Source code:

<https://github.com/xpenatan/gdx-jolt/blob/master/examples/samples/core/src/main/java/jolt/example/samples/app/tests/playground/box/BoxSpawnTest.java>

My modifications to original:

Removed ImGui, debug rendering, and added a few optimizations. This resulted in the libGDX version doubling the FPS on the web on my hardware.

Godot version written using **GDS**cript.

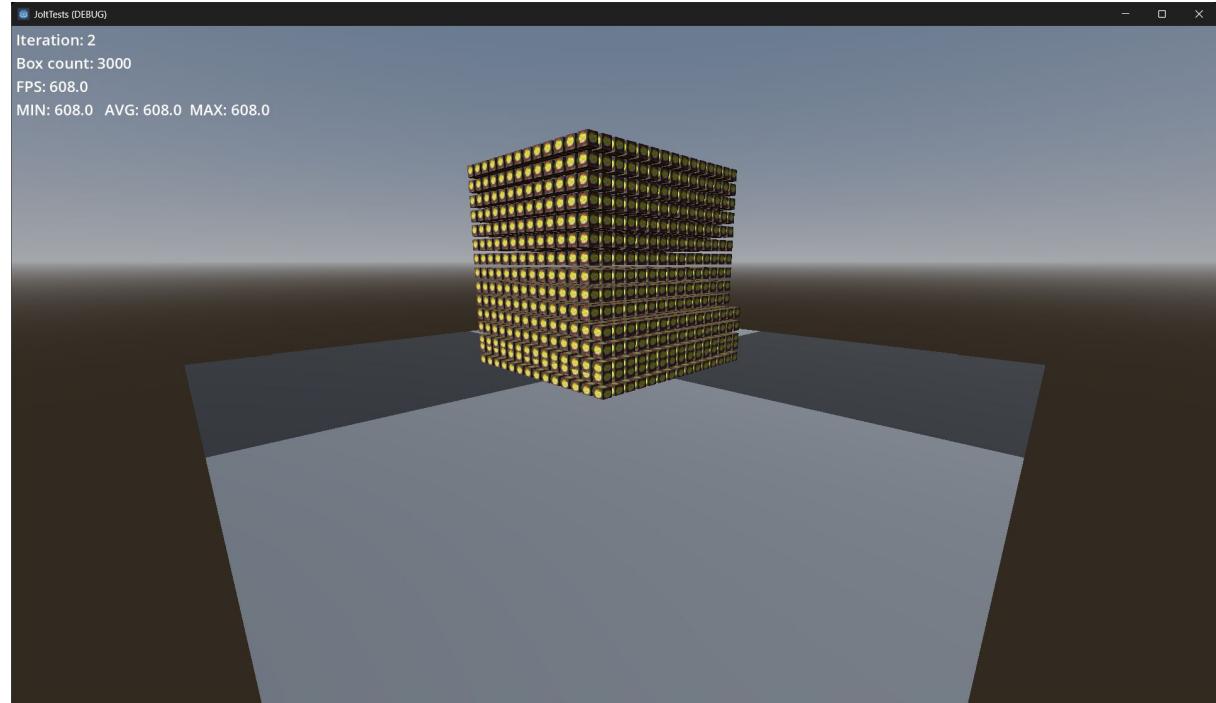
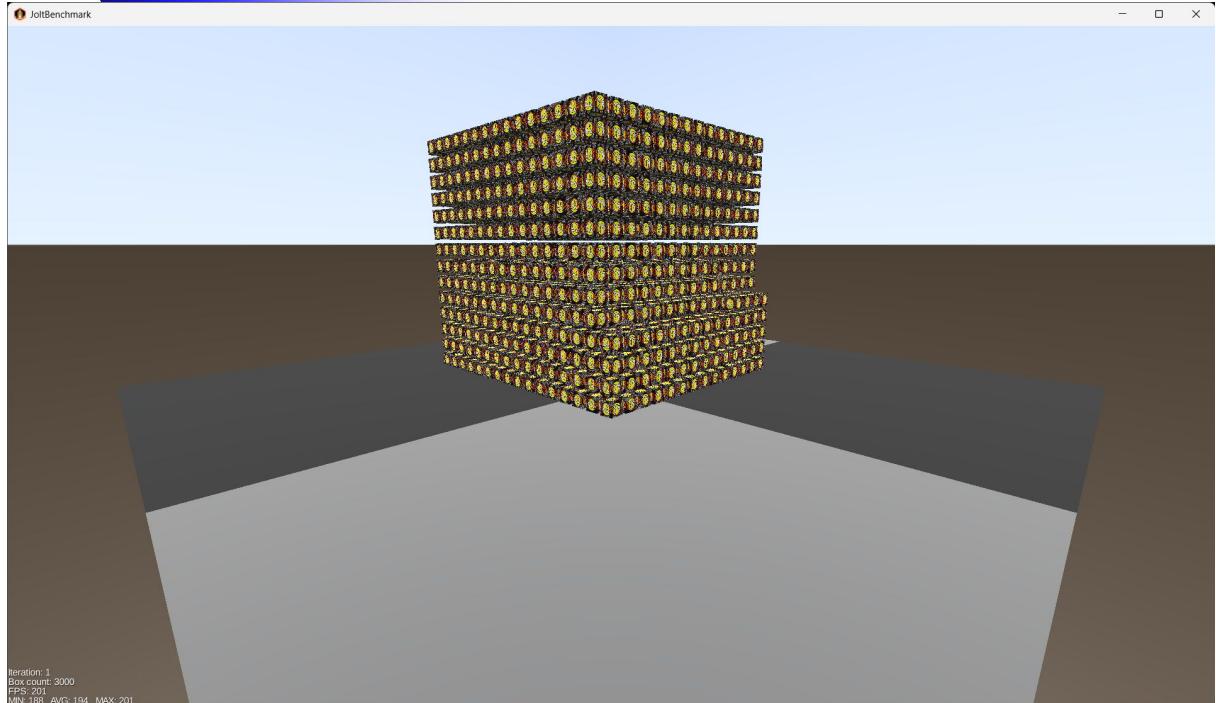
Rendering

What is being rendered?

- 3000 rigid body boxes, with libGDX's Badlogic texture applied.
- Ground with simple checkerboard texture applied.
- 1 directional light.
- No shadows.
- No instancing.
- No MSAA.
- A few labels for stats, only updating once per second.
- Monitor resolution of 1920x1080@75 Hz.

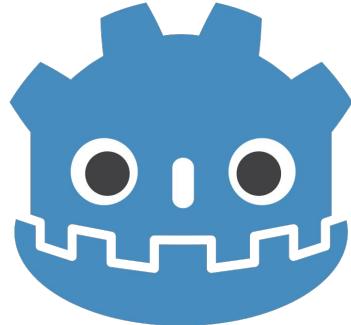
A test **with** and **without** rendering the 3000 cubes to highlight how much the actual rendering is impacting performance. Physics is always running.

Visual Differences



lib
GDX

vs.



GODOT
Game engine

Thread Details

Windows Desktop

- **LibGDX** using 11 threads for Jolt Physics only.
- **Godot** using separate thread for Jolt Physics, and *threading/worker_pool/max_threads* set to 4.
- **Godot** using **RELEASE** builds.

Web/HTML

- Using **Chrome** based browser (Chromium: 138.0.7204.49)
- **libGDX** using **gdx-teavm** for web build. Only supports single thread.
- **Godot** will test both *single thread* and *multi-thread* web builds.
- **Godot** using **DEBUG** builds.

Statistics captured

Iterations

- Will capture 6 iterations, but will remove the first iteration from the data set as it performs worse on both libGDX and Godot.

Minimum, Average, Maximum FPS

- FPS is sampled once per second (both libGDX and Godot do this out of the box).
- Minimum, average, maximum FPS are calculated and updated every second.
- Each iteration test lasts about 12 seconds.
- The final averaged out stats are reported and captured after 12 seconds.
- Then the test automatically starts another iteration.

Statistics sample output

```
1:55:52 PM: Executing 'run'...

[AntzFPSLogger] libGDX iteration 0 final report >>> FPS MIN: 95    FPS AVG: 183    FPS MAX: 199
[AntzFPSLogger] libGDX iteration 1 final report >>> FPS MIN: 191    FPS AVG: 196    FPS MAX: 202
[AntzFPSLogger] libGDX iteration 2 final report >>> FPS MIN: 192    FPS AVG: 198    FPS MAX: 201
[AntzFPSLogger] libGDX iteration 3 final report >>> FPS MIN: 190    FPS AVG: 197    FPS MAX: 210
[AntzFPSLogger] libGDX iteration 4 final report >>> FPS MIN: 189    FPS AVG: 196    FPS MAX: 201
[AntzFPSLogger] libGDX iteration 5 final report >>> FPS MIN: 185    FPS AVG: 194    FPS MAX: 203

1:57:17 PM: Execution finished 'run'.
```

```
OpenGL API 3.3.0 NVIDIA 576.80 - Compatibility - Using Device: NVIDIA - NVIDIA GeForce RTX 3050

Godot iteration 0 final report >>> FPS MIN: 241.0    FPS AVG: 320.4    FPS MAX: 528.0
Godot iteration 1 final report >>> FPS MIN: 242.0    FPS AVG: 341.0    FPS MAX: 733.0
Godot iteration 2 final report >>> FPS MIN: 238.0    FPS AVG: 341.0    FPS MAX: 729.0
Godot iteration 3 final report >>> FPS MIN: 241.0    FPS AVG: 342.1    FPS MAX: 713.0
Godot iteration 4 final report >>> FPS MIN: 243.0    FPS AVG: 343.3    FPS MAX: 729.0
Godot iteration 5 final report >>> FPS MIN: 245.0    FPS AVG: 342.6    FPS MAX: 724.0
```

Statistics sample output - Web

AntzFPSLogger: libGDX iteration 0 final report >>> FPS MIN: 9	FPS AVG: 24	FPS MAX: 29	app.js:424
AntzFPSLogger: libGDX iteration 1 final report >>> FPS MIN: 21	FPS AVG: 28	FPS MAX: 30	app.js:424
AntzFPSLogger: libGDX iteration 2 final report >>> FPS MIN: 20	FPS AVG: 28	FPS MAX: 30	app.js:424
AntzFPSLogger: libGDX iteration 3 final report >>> FPS MIN: 21	FPS AVG: 28	FPS MAX: 30	app.js:424
AntzFPSLogger: libGDX iteration 4 final report >>> FPS MIN: 23	FPS AVG: 27	FPS MAX: 30	app.js:424
AntzFPSLogger: libGDX iteration 5 final report >>> FPS MIN: 22	FPS AVG: 28	FPS MAX: 30	app.js:424

Godot Engine v4.5.beta1.official.46c495ca2 - https://godotengine.org	tmp_js_export.js:473		
OpenGL API OpenGL ES 3.0 (WebGL 2.0 (OpenGL ES 3.0 Chromium)) - Compatibility -	tmp_js_export.js:473		
Using Device: WebKit - WebKit WebGL			
Build configuration: Emscripten 4.0.10, multi-threaded, no GDExtension support.	tmp_js_export.js:473		
Godot iteration 0 final report >>> FPS MIN: 57.0	FPS AVG: 72.8	FPS MAX: 75.0	tmp_js_export.js:473
Godot iteration 1 final report >>> FPS MIN: 75.0	FPS AVG: 75.0	FPS MAX: 75.0	tmp_js_export.js:473
Godot iteration 2 final report >>> FPS MIN: 75.0	FPS AVG: 75.0	FPS MAX: 75.0	tmp_js_export.js:473
Godot iteration 3 final report >>> FPS MIN: 74.0	FPS AVG: 74.9	FPS MAX: 75.0	tmp_js_export.js:473
Godot iteration 4 final report >>> FPS MIN: 75.0	FPS AVG: 75.0	FPS MAX: 75.0	tmp_js_export.js:473
Godot iteration 5 final report >>> FPS MIN: 74.0	FPS AVG: 74.9	FPS MAX: 75.0	tmp_js_export.js:473

Note: Max FPS on web build is 75 Hz, which means Godot could have posted even higher FPS numbers on a monitor with higher refresh rate.

Jolt Details

Physics Ticks

- **Godot** using 60 ticks per second for physics.
- **LibGDX** using a custom solution to try to target 60 physics ticks per second as libGDX does not have any built in 3D physics framework in place to do this.

Settings

libGDX using Godot's Jolt default values.

```
int mMaxBodies = 10240;  
int mMaxBodyPairs = 65536;  
int mMaxContactConstraints = 20480;  
int mTempAllocatorSize = 32 * 1024 * 1024; // 32 MB
```

Notable Differences

Jolt

- **Godot** simulation is more smooth. **Godot** simulation is more bouncy, even with the **same values of restitution** (bounciness).
- Note same values of gravity, friction, mass were used in both simulations.

SIMD

- **Godot** entire engine (not just physics) takes advantage of **SIMD**, including in web builds as of **Godot 4.5** which is in beta.

Windows Desktop

LibGDX using 11 threads for Jolt Physics only.
Godot using separate thread for Jolt Physics.

Mode	libGDX	Mode	GODOT	
Rendering Cubes	iter 1 MIN: 188 AVG: 202 MAX: 213 iter 2 MIN: 185 AVG: 198 MAX: 212 iter 3 MIN: 189 AVG: 198 MAX: 211 iter 4 MIN: 188 AVG: 197 MAX: 205 iter 5 MIN: 194 AVG: 200 MAX: 208	Average FPS: 199	Iter 1 MIN: 242.0 AVG: 341.0 MAX: 733.0 Iter 2 MIN: 238.0 AVG: 341.0 MAX: 729.0 Iter 3 MIN: 241.0 AVG: 342.1 MAX: 713.0 Iter 4 MIN: 243.0 AVG: 343.3 MAX: 729.0 Iter 5 MIN: 245.0 AVG: 342.6 MAX: 724.0	Average FPS: 341
Not Rendering Cubes	Iter 1 MIN: 1170 AVG: 1195 MAX: 1201 Iter 2 MIN: 1172 AVG: 1196 MAX: 1201 Iter 3 MIN: 1173 AVG: 1196 MAX: 1200 Iter 4 MIN: 1174 AVG: 1196 MAX: 1201 Iter 5 MIN: 1174 AVG: 1197 MAX: 1201	Average FPS: 1196	Iter 1 MIN: 2291.0 AVG: 2383.2 MAX: 2434.0 Iter 2 MIN: 2302.0 AVG: 2390.4 MAX: 2469.0 Iter 3 MIN: 2293.0 AVG: 2378.8 MAX: 2417.0 Iter 4 MIN: 2351.0 AVG: 2393.7 MAX: 2426.0 Iter 5 MIN: 2208.0 AVG: 2383.3 MAX: 2467.0	Average FPS: 2386
Physics Only				

HTML

Note: Max FPS on my web build is 75 Hz, which means Godot could have posted even higher FPS numbers on a monitor with higher refresh rate.

Mode	libGDX	Mode	GODOT (DEBUG build)
Rendering Cubes	<p>Iter 1 final MIN: 21 AVG: 28 MAX: 30 Iter 2 final MIN: 20 AVG: 28 MAX: 30 Iter 3 final MIN: 21 AVG: 28 MAX: 30 Iter 4 final MIN: 23 AVG: 27 MAX: 30 Iter 5 final MIN: 22 AVG: 28 MAX: 30</p> <p>Average FPS: 28</p>	Single Thread	<p>Iter 1 MIN: 64.0 AVG: 73.2 MAX: 75.0 Iter 2 MIN: 66.0 AVG: 74.0 MAX: 75.0 Iter 3 MIN: 65.0 AVG: 73.7 MAX: 75.0 Iter 4 MIN: 63.0 AVG: 73.7 MAX: 75.0 Iter 5 MIN: 64.0 AVG: 73.8 MAX: 75.0</p> <p>Average FPS: 73.7</p>
		Multiple Threads	<p>Iter 1 MIN: 75.0 AVG: 75.0 MAX: 75.0 Iter 2 MIN: 75.0 AVG: 75.0 MAX: 75.0 Iter 3 MIN: 74.0 AVG: 74.9 MAX: 75.0 Iter 4 MIN: 75.0 AVG: 75.0 MAX: 75.0 Iter 5 MIN: 74.0 AVG: 74.9 MAX: 75.0</p> <p>Average FPS: 74.96</p>

Conclusions

Still preliminary and awaiting verification.

With my hardware configuration, **Jolt Physics**:

- On Windows desktop, **Godot** is **1.71x** to **1.99x** faster than **libGDX**.
- On the web, **Godot** is at least **2.62x** faster than **libGDX**.