

JOLT Physics Engine



GODOT

Game engine

vs.

lib
GDX

Desktop and Web Performance Tests

v0.3

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.beta4 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.

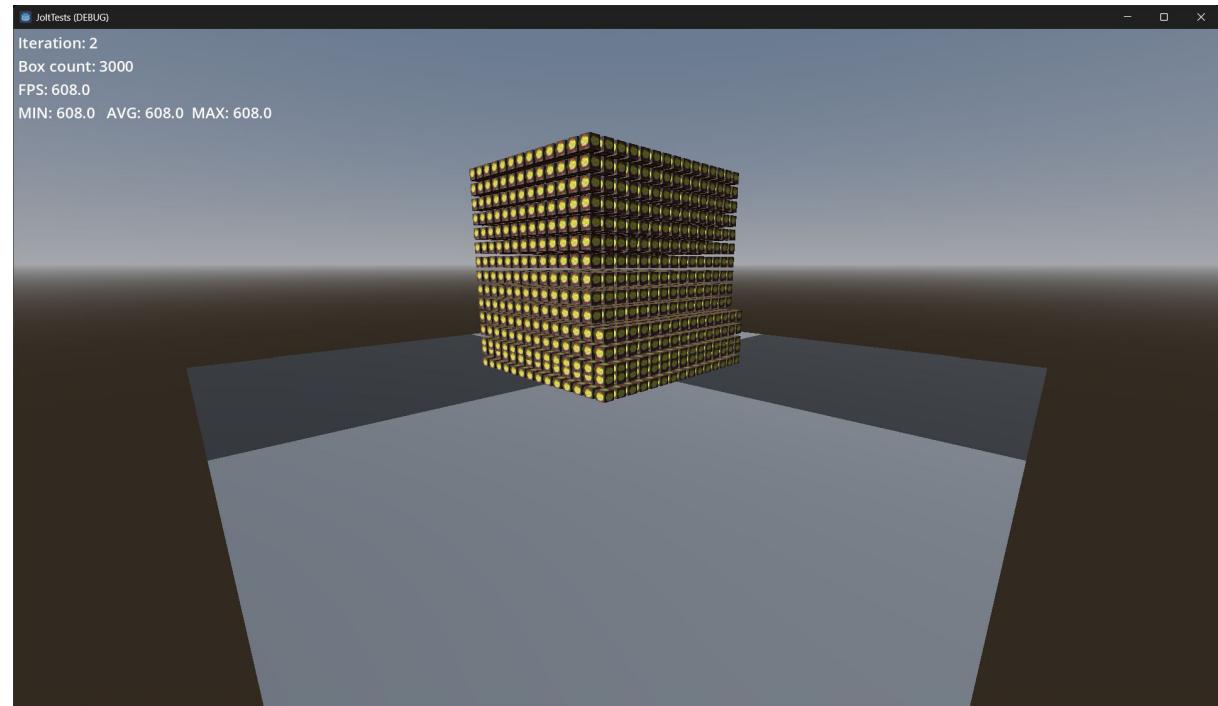
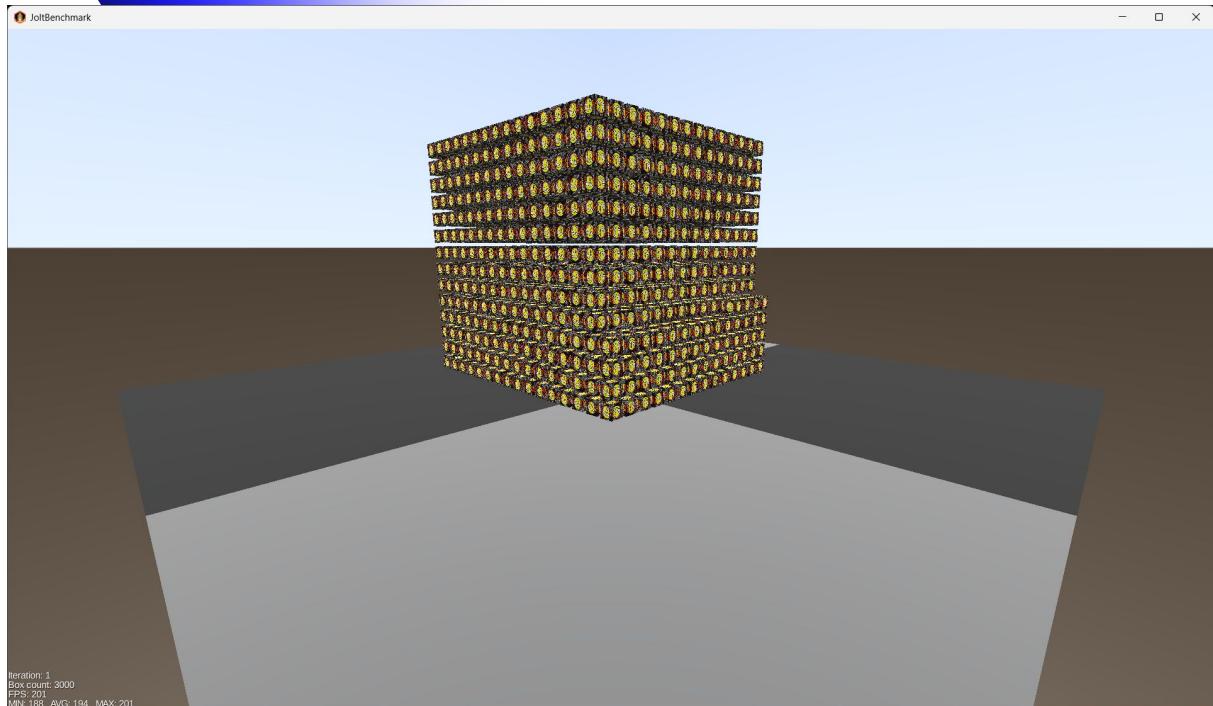
Godot version written using **GDS**cript.

Rendering

What is being rendered?

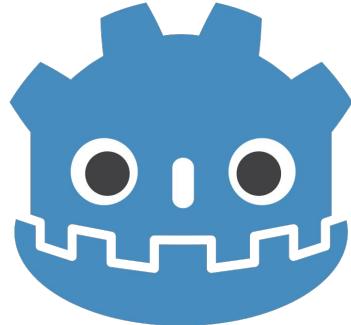
- **No instancing** of 3000 cubes with libGDX's Badlogic texture applied.
(MeshInstance3D for Godot, ModelInstance for libGDX)
- 3000 rigid body boxes. 1 kg, 1m³ each
- Ground with simple checkerboard texture applied.
- 1 directional light.
- No shadows.
- No MSAA.
- A few labels for stats, only updating once per second.
- Monitor resolution of 1920x1080@75 Hz.

Visual Differences



lib
GDX

vs.



GODOT
Game engine

Thread Details

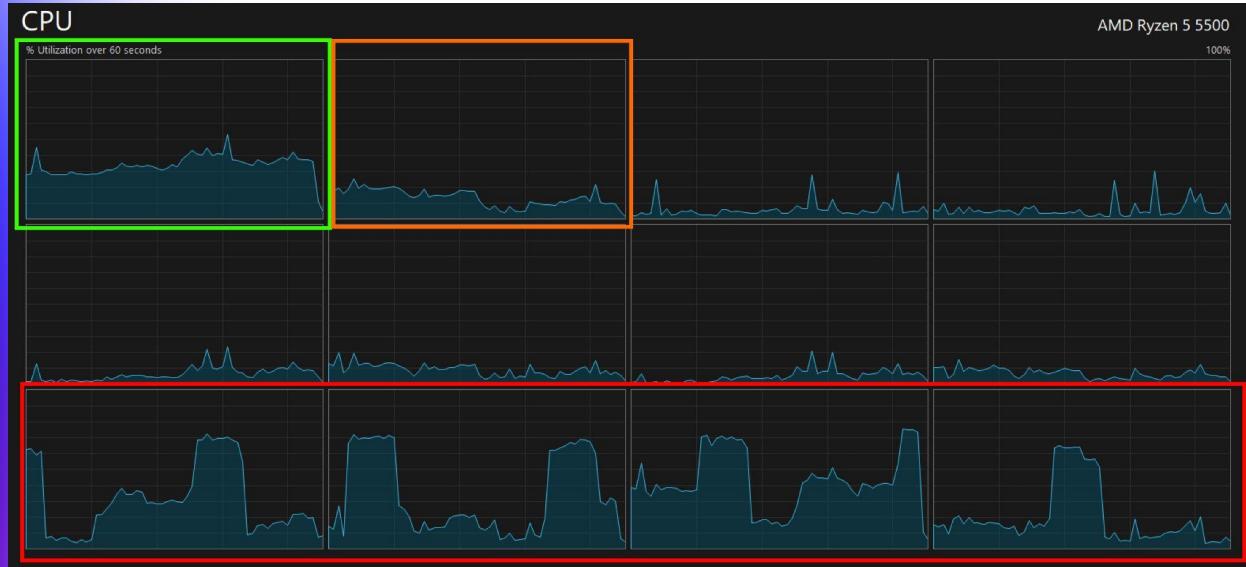
Windows Desktop

- **LibGDX** using 4 threads for Jolt Physics only.
- **Godot** using separate thread for Jolt Physics Engine. No control over how Jolt allocates threads. On my system its using 4 threads.
- **Godot** using **RELEASE** build.

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 *single thread* web builds.
- **Godot** using **RELEASE** build.

Thread Details - No rendering

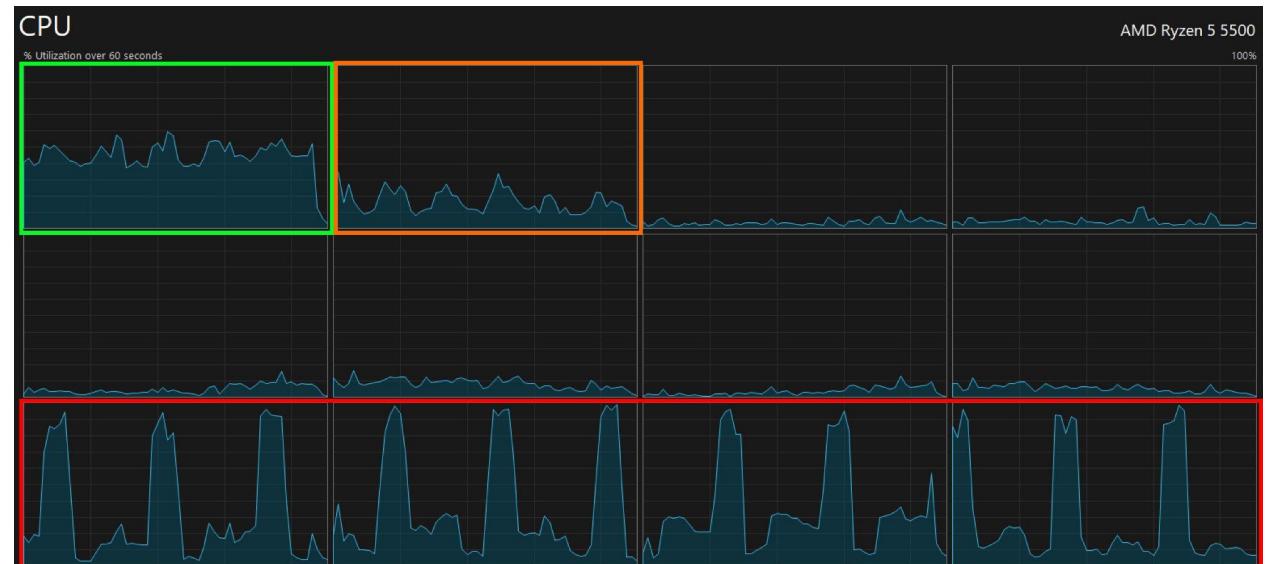


<<<

libGDX with 4 Jolt threads (**red**)
Green is LWJGL thread.
Orange is Jolt Engine.

>>>

Godot with 4 Jolt threads (**red**)
Green is GLES game thread.
Orange is Jolt Engine.



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. Therefore total of 5 iterations.

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 - Windows

```
OpenGL API 3.3.0 NVIDIA 577.00 - Compatibility - Using Device: NVIDIA - NVIDIA GeForce RTX 3050

Godot iteration 0 will be discarded. >>> FPS MIN: 238   FPS AVG: 324   FPS MAX: 569
Godot iteration 1 report >>> FPS MIN: 245   FPS AVG: 345   FPS MAX: 765
Godot iteration 2 report >>> FPS MIN: 243   FPS AVG: 345   FPS MAX: 756
Godot iteration 3 report >>> FPS MIN: 241   FPS AVG: 344   FPS MAX: 744
Godot iteration 4 report >>> FPS MIN: 243   FPS AVG: 341   FPS MAX: 742
Godot iteration 5 report >>> FPS MIN: 243   FPS AVG: 343   FPS MAX: 743
*** Jolt Test has finished! ***
Godot final report, 5 iterations.  MIN: 243.0   AVG: 343.8   MAX: 750.0
```

```
Number of threads for gdx-jolt set to: 4
[Jolt Config] Number of threads: 4
[AntzFPSLogger] libGDX iteration 0 will be discarded >>> FPS MIN: 138   FPS AVG: 193   FPS MAX: 209
[AntzFPSLogger] libGDX iteration 1 report >>> FPS MIN: 193   FPS AVG: 202   FPS MAX: 208
[AntzFPSLogger] libGDX iteration 2 report >>> FPS MIN: 193   FPS AVG: 201   FPS MAX: 212
[AntzFPSLogger] libGDX iteration 3 report >>> FPS MIN: 188   FPS AVG: 201   FPS MAX: 213
[AntzFPSLogger] libGDX iteration 4 report >>> FPS MIN: 193   FPS AVG: 201   FPS MAX: 213
[AntzFPSLogger] libGDX iteration 5 report >>> FPS MIN: 192   FPS AVG: 198   FPS MAX: 210
[JoltTest] *** Test has finished! ***
[libGDX final report, 5 iterations. ] MIN: 191.8   AVG: 200.6   MAX: 211.2
```

Statistics - Web/HTML console

```
Godot Engine v4.5.beta3.official.4d1f26e1f - https://godotengine.org
```

```
OpenGL API OpenGL ES 3.0 (WebGL 2.0 (OpenGL ES 3.0 Chromium)) - Compatibility -  
Using Device: WebKit - WebKit WebGL
```

```
Build configuration: Emscripten 4.0.10, single-threaded, no GDExtension support.
```

```
Godot iteration 0 will be discarded. >>> FPS MIN: 61   FPS AVG: 73   FPS MAX: 75
```

```
Godot iteration 1 report >>> FPS MIN: 72   FPS AVG: 74   FPS MAX: 76
```

```
Godot iteration 2 report >>> FPS MIN: 72   FPS AVG: 74   FPS MAX: 75
```

```
Godot iteration 3 report >>> FPS MIN: 72   FPS AVG: 74   FPS MAX: 76
```

```
Godot iteration 4 report >>> FPS MIN: 72   FPS AVG: 74   FPS MAX: 75
```

```
Godot iteration 5 report >>> FPS MIN: 72   FPS AVG: 74   FPS MAX: 75
```

```
*** Jolt Test has finished! ***
```

```
Godot final report, 5 iterations.  MIN: 72.0   AVG: 74.5   MAX: 75.4
```

```
AntzFPSLogger: libGDX iteration 0 will be discarded >>> FPS MIN: 21   FPS AVG: 31   FPS MAX: 62
```

```
AntzFPSLogger: libGDX iteration 1 report >>> FPS MIN: 21   FPS AVG: 28   FPS MAX: 31
```

```
AntzFPSLogger: libGDX iteration 2 report >>> FPS MIN: 22   FPS AVG: 28   FPS MAX: 30
```

```
AntzFPSLogger: libGDX iteration 3 report >>> FPS MIN: 23   FPS AVG: 27   FPS MAX: 31
```

```
AntzFPSLogger: libGDX iteration 4 report >>> FPS MIN: 21   FPS AVG: 28   FPS MAX: 30
```

```
AntzFPSLogger: libGDX iteration 5 report >>> FPS MIN: 20   FPS AVG: 27   FPS MAX: 30
```

```
JoltTest: *** Test has finished! ***
```

```
libGDX final report, 5 iterations.  : MIN: 21.4   AVG: 27.6   MAX: 30.4
```

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** takes advantage of **SIMD**, in web builds as of **Godot 4.5** which is in beta.

Windows Desktop

LibGDX using 4 threads for Jolt Physics.
Godot using 4 threads for Jolt Physics.

Mode	libGDX	Mode	GODOT
Rendering Cubes	<p>iter: 1 MIN: 193 AVG: 202 MAX: 208 iter: 2 MIN: 193 AVG: 201 MAX: 212 iter: 3 MIN: 188 AVG: 201 MAX: 213 iter: 4 MIN: 193 AVG: 201 MAX: 213 iter: 5 MIN: 192 AVG: 198 MAX: 210</p> <p>MIN: 191.8 AVG: 200.6 MAX: 211.2</p> <p>Average FPS: 200.6</p>	<p>iter: 1 MIN: 245 AVG: 345 MAX: 765 iter: 2 MIN: 243 AVG: 345 MAX: 756 iter: 3 MIN: 241 AVG: 344 MAX: 744 iter: 4 MIN: 243 AVG: 341 MAX: 742 iter: 5 MIN: 243 AVG: 343 MAX: 743</p> <p>MIN: 243.0 AVG: 343.8 MAX: 750.0</p> <p>Average FPS: 343.8</p>	

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.

Godot using single thread and a Release build for HTML.

Mode	libGDX	Mode	GODOT
Rendering Cubes	<p>iter 1: MIN: 21 AVG: 28 MAX: 31 iter 2: MIN: 22 AVG: 28 MAX: 30 iter 3: MIN: 23 AVG: 27 MAX: 31 iter 4: MIN: 21 AVG: 28 MAX: 30 iter 5: MIN: 20 AVG: 27 MAX: 30</p> <p>MIN: 21.4 AVG: 27.6 MAX: 30.4</p> <p>Average FPS: 27.6</p>	<p>Single Thread</p> <p>Rendering Cubes</p>	<p>iter 1: MIN: 72 AVG: 74 MAX: 76.0 iter 2: MIN: 72 AVG: 74 MAX: 75.0 iter 3: MIN: 72 AVG: 74 MAX: 76.0 iter 4: MIN: 72 AVG: 74 MAX: 75.0 iter 5: MIN: 72 AVG: 74 MAX: 75.0</p> <p>MIN: 72.0 AVG: 74.5 MAX: 75.4</p> <p>Average FPS: 74.5</p>

Conclusions

Still preliminary and awaiting verification. No one from libGDX community is willing to confirm results so far.

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

- On Windows desktop, **Godot** is **1.71x** faster than **libGDX**.
- On the web, **Godot** is at least **2.70x** faster than **libGDX**. A higher monitor refresh rate is need to see what the upper limit is.