Integrating the Bullet Physics engine into Minecraft

Ethan Johnson

Department of Computer Science, Minnesota State University Moorhead

CSIS 492 Senior Seminar

Dr. Andrew Chen

May 2, 2021

Abstract

During the past fall semester, I started a programming project called Rayon which is designed to be a realistic physics engine implementation that runs alongside the videogame Minecraft. It is a library which Minecraft mod developers can use to implement realistic entity movement into their own mods. Rayon, being entirely written in the Java programming language, currently uses a port of the Bullet physics engine called JBullet which is very outdated and no longer being maintained. To find a more performant solution, I have set out to replace JBullet with an alternative library called LibBulletJME which is designed to interface with the original Bullet library written in C++ (generally a faster programming language).

Introduction

Minecraft is a game which many people are already familiar with given its decade long life and its ever growing popularity. A popular aspect of the game which has been around since the beginning is modding. Minecraft modding allows users to install community-made mods into their game in order to add content and/or change the game's behavior. The project I chose to work on was made to be an intermediary between Minecraft mod developers and Minecraft mods themselves in the form of a software library.

Summary

The project is called Rayon which is designed to be a realistic physics engine implementation that runs alongside the videogame Minecraft. It is a library which mod developers for Minecraft can use to implement realistic physics-based entity movement into their own mods. Rayon currently supports rigid body dynamics for in-game entities (such as living entities like cows or sheep and non-living entities like boats) as well as loading various collision shapes in place of “blocks”, which make up the typical environment in a Minecraft world. It is configurable in terms of gravity, air resistance, simulation step rate, etc.

Rayon, being entirely written in Java (like Minecraft itself), used a port of the Bullet physics engine called JBullet. However, it is very outdated and no longer being maintained. An alternative exists called LibBulletJME which is a java native interface version of Bullet that yields better performance. The reason for the performance boost is because under the hood native C++ code is executed instead of java code. After replacing JBullet with LibBulletJME, there was a noticeable performance boost. Most of the work involved with the transition had to do with reading the documentation for both libraries as well as studying their code. However, since they both represent the same physics engine, it was relatively trivial to swap them out.

The next step was to implement multithreading into Rayon. What that means is the physics world will simulate asynchronously with Minecraft’s game logic. Before, Minecraft’s game logic would be blocked from executing each loop while Rayon was simulating the physics world. Implementing multithreading proved to be much more difficult than swapping out bullet libraries since it required many structural changes to

Rayon. It also required some additional research into multithreading within Java itself since I had little experience with it. After the work was complete, it was found that there was a staggering ten times performance boost during some quick load testing (I tested how many entities could be spawned into the game before it became unplayable).

Conclusion

Rayon is a powerful tool for Minecraft mod developers looking to incorporate rigid body dynamics into their own modifications since it makes use of native execution and multithreading. It was very enjoyable to work on since I believe it furthered my programming knowledge greatly. Also since it is an open source project, I plan on maintaining it for the foreseeable future.

Work Cited

“Bullet Real-Time Physics Simulation.” Bullet RealTime Physics Simulation. https://pybullet.org/wordpress/index.php/forum-2/.

Bullet Things.” darkrockstudios, May 26, 2008. http://darkrockstudios.com/files/bulletthings.pdf.

Coumans, Erwin. “Bullet 2.83 Physics SDK Manual.” Github, 2015. https://github.com/bulletphysics/bullet3/raw/master/docs/Bullet_User_Manual.pdf.

Fabric Wiki. FabricMC, n.d. https://fabricmc.net/wiki/doku.php.

FabLabsMC. “FabLabsMC/Fiber.” GitHub, n.d. https://github.com/FabLabsMC/fiber.

Gold, Stephen. “Stephengold/Libbulletjme.” Libbulletjme. Github, January 24, 2021. https://github.com/stephengold/Libbulletjme.

“Java API Reference.” Java Native Interface Specification: 1 - Introduction, January 29, 2021. https://docs.oracle.com/en/java/javase/11/docs/specs/jni/intro.html.

“Java Port of Bullet Physics Library.” JBullet, n.d. http://jbullet.advel.cz/.

jMonkeyEngine Docs. jMonkeyEngine, 2020. https://wiki.jmonkeyengine.org/docs/3.3/physics/physics.html.

OnyxStudios, Pyrofab. “OnyxStudios/Cardinal-Components-API.” GitHub, n.d. https://github.com/OnyxStudios/Cardinal-Components-API.

Shedaniel. “Shedaniel/Cloth-Api.” GitHub, n.d. https://github.com/shedaniel/cloth-api.

SpongePowered. “SpongePowered/Mixin.” GitHub, n.d. https://github.com/SpongePowered/Mixin.