6 Fazit

In dieser Arbeit wurde basierend auf der Analyse der Blocklib eine nebenläufige Architektur entworfen und implementiert. Die Architektur baut auf den Strukturen und Ideen System on a Thread (SoT) und Jobsystem auf, die in der Computerspielindustrie für die Entwicklung nebenläufiger Architekturen verwendet werden. Da die Blocklib OpenGL nutzt, sind neben des Jobsystem-Designs notwendigerweise auch Aspekte des SoT integriert worden. Dadurch ist eine hybride Form der Modelle entstanden.

Der SoT-Teil der Architektur nutzt einen Render-Thread, um das Rendering nebenläufig zu der Simulation durchzuführen. Zentrale Zwischenspeichersysteme, die aus Double-Buffern bestehen, sorgen dafür, dass Wettkampfbedingungen auf den zu rendernden Objekten vermieden werden. Um die Leistung zu optimieren, werden die Objekte, die zur Zwischenspeicherung genutzt werden, nach Gebrauch wiederverwendet.

Das implementiere Jobsystem erfüllt die ermittelten Anforderungen. Es definiert eine Stelle, an der Threads kontrolliert werden, den `BlocklibExecutor`, und eine zugehörige Schnittstelle `BlocklibExecutorService`. Das Jobsystem bietet die Möglichkeit, Aufgaben zu definieren, die nebenläufig von Threads abgearbeitet werden. Durch die Dekorierung dieser Aufgaben können über die Schnittstelle `CompletionStage` komplexe Aufgaben gelöst werden. Das wird ermöglicht, indem `CompletionStage` Methoden bereitstellt, um einzelne Jobs zu einem komplexen Job-Graphen zu komponieren. Des Weiteren ermöglicht das Jobsystem, wiederkehrende Aufgaben zu definieren, die automatisiert zu bestimmten Zeiten nebenläufig ausgeführt werden.

Das Jobsystem ermöglicht eine rudimentäre Definition von Prioritäten für die aufgegebenen Tasks, indem zwei getrennte Thread-Pools genutzt werden. Das System ist so vorbereitet, dass ein verbessertes Prioritätensystem basierend auf einer Prioritäts-Warteschlange einfach zu integrieren ist, da die Schnittstelle `BlocklibExecutorService` bereits entsprechend definiert ist und der Aufzählungstyp `TaskPriority` bereitgestellt wird.

Ein Großteil der bereits bestehenden Nutzung von Multithreading konnte so geändert werden, dass nun die neue nebenläufige Architektur verwendet wird. Einzig die Netzwerkfunktionalitäten der Blocklib bilden hier eine Ausnahme.

Der Start der Blocklib wird durch die neue Architektur um eine Zeit im einstelligen Sekundenbereich verlangsamt. Verglichen mit der alten Architektur erreicht die neue nebenläufige Architektur eine um durchschnittlich 172 % höhere Bildwiederholrate und weist damit eine deutlich gestiegene Leistung auf. Dabei nutzt die Blocklib mit der neuen Architektur auch einen größeren Teil der zur Verfügung stehenden Leistung des Testsystems aus. Durchschnittlich wird die CPU nun 40 % und die GPU 37 % stärker ausgelastet als zuvor. In den Messungen ist die CPU auch mit der neuen Architektur weiterhin die leistungsbeschränkende Komponente. In beinahe keinem Szenario ist die GPU voll ausgelastet.