

Threadoptimierung

Multithreading ist ein wichtiger Aspekt bei der Entwicklung von Computerspielen, insbesondere in Unity, einer der führenden Spiel-Engine-Plattformen. Es ermöglicht es Entwicklern, ihre Spiele effizienter und schneller laufen zu lassen, indem es mehrere Teile des Spiels gleichzeitig ausführt.

Unity verwendet das Konzept des multithreaded Job-Systems, das es Entwicklern ermöglicht, Aufgaben auf mehrere Threads zu verteilen, um eine höhere Leistung und Effizienz zu erzielen. Dies ist besonders wichtig, wenn es darum geht, komplexe Berechnungen wie KI, Physik-Simulationen und Grafikberechnungen auszuführen, die viel Rechenleistung benötigen. Mit multithreaded Jobs können diese Berechnungen parallel ausgeführt werden, was bedeutet, dass sie schneller abgeschlossen werden und das Spiel reibungsloser läuft.

Das multithreaded Job-System in Unity ist sehr benutzerfreundlich und einfach zu verwenden. Entwickler können Jobs erstellen, die auf mehrere Threads aufgeteilt werden können, indem sie einfach eine Funktion schreiben und diese an das Job-System übergeben. Das System übernimmt dann die Verteilung der Jobs auf mehrere Threads und sorgt dafür, dass sie effizient ausgeführt werden.

Multithreading ist auch wichtig, um die Leistung auf unterschiedlichen Plattformen zu verbessern. Durch die Verwendung von mehreren Threads können Entwickler sicherstellen, dass das Spiel auf unterschiedlichen Geräten und Plattformen gut läuft, einschließlich mobilen Geräten und Spielekonsolen.

In Bezug auf die Entwicklung von Spielen ist Multithreading ein wichtiger Faktor, um eine höhere Leistung und Effizienz zu erreichen. Mit dem multithreaded Job-System in Unity können Entwickler Aufgaben parallel ausführen, um sicherzustellen, dass ihre Spiele reibungslos und schnell laufen, egal auf welcher Plattform. Somit wird klar, dass Multithreading in Unity ein unverzichtbares Werkzeug für die Spielentwicklung ist, das es Entwicklern ermöglicht, ihre Spiele effizienter und schneller laufen zu lassen, indem es mehrere Teile des Spiels gleichzeitig ausführt. Es ist einfach zu verwenden und hilft, die Leistung auf unterschiedlichen Plattformen zu verbessern.

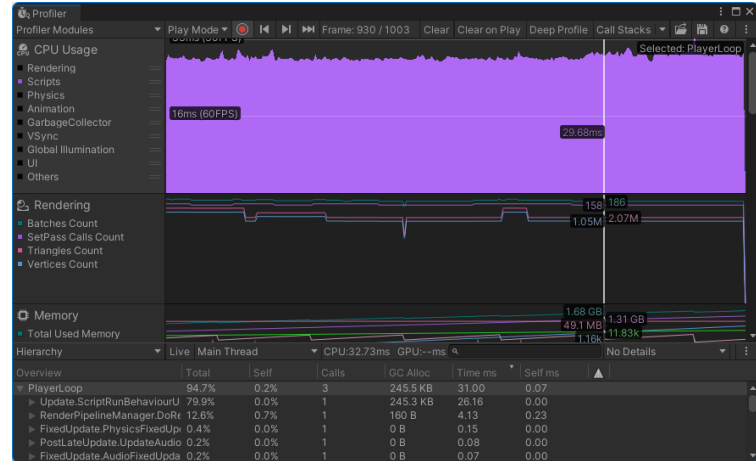
Ein weiterer Vorteil von Multithreading in Unity ist die Verbesserung der Benutzerfreundlichkeit. Wenn Teile des Spiels parallel ausgeführt werden, kann es eine höhere Bildwiederholrate geben, was bedeutet, dass das Spiel reibungsloser läuft und eine bessere Spielerfahrung bietet. Es kann auch die Reaktionszeit des Spiels verbessern, wodurch Spieler schneller auf ihre Eingaben reagieren können.

Ein weiteres wichtiges Konzept bei der Verwendung von Multithreading in Unity ist die Verwaltung von Threads. Es ist wichtig zu beachten, dass das parallele Ausführen von Aufgaben zu Konflikten und Problemen führen kann, wenn sie nicht sorgfältig verwaltet werden. Entwickler müssen sicherstellen, dass sie Threads korrekt verwalten und überwachen, um sicherzustellen, dass sie effizient und ohne Fehler ausgeführt werden.

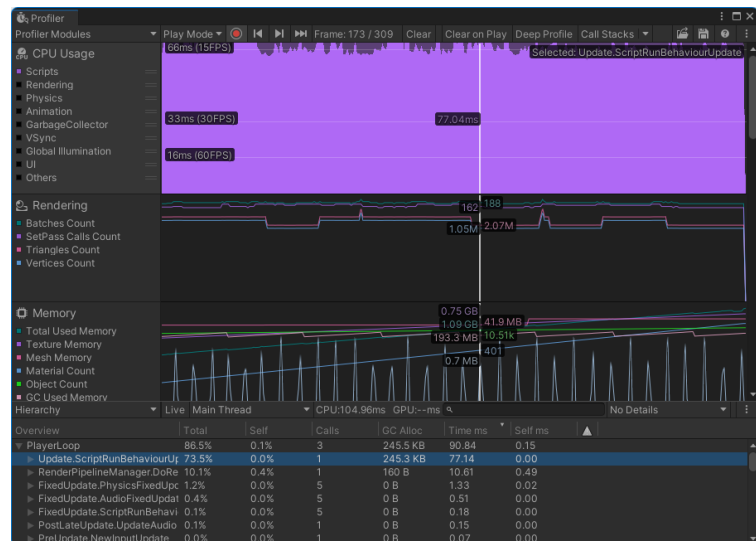
Zusammenfassend kann gesagt werden, dass Multithreading in Unity ein wichtiger Faktor bei der Entwicklung von Computerspielen ist, der es Entwicklern ermöglicht, ihre Spiele effizienter und schneller laufen zu lassen. Es ist einfach zu verwenden und hilft, die Leistung auf unterschiedlichen Plattformen zu verbessern, die Benutzerfreundlichkeit zu verbessern und die Reaktionszeit des Spiels zu verbessern. Entwickler müssen jedoch sicherstellen, dass sie Threads korrekt verwalten und überwachen, um sicherzustellen, dass sie effizient und ohne Fehler ausgeführt werden.

Anschließend möchte ich die erhobenen Schlüsseldaten der Performance meiner selbst geschriebenen Skripte in dem angefügten Projekt gegenüberstellen, die ich bei der Entwicklung eben dieses erhoben habe.

Mit der Optimierung durch Multithreading des Chunksystems ist klar zu erkennen, bei einer Berechnungszeit von ca. 26.16ms und einer daraus resultierenden Framerate von über 30 Frames je Sekunde, dass dies zu einer erheblichen Leistungssteigerung beiträgt. Die Berechnungszeit bezieht sich hier auf den Update-Loop, in dem die Chunksystems-Teile berechnet werden. Diese umfassen Höhendaten, Chunk-Größe, Mesh-Generierung und Kollisionsberechnung in Form der Mesh-Collider.



Vor der Optimierung durch Multithreading beträgt die Framerate teilweise unter 15 Frames die Sekunde, was zu einem klaren Vergleich der Multithreadoptimierten Daten für sich spricht. Die Berechnungszeit des Update-Loops mit den gleichen Aufgaben benötigt ca. 77.14ms.



Als Fazit zur Optimierung stelle ich die Daten der optimierten sowie nicht optimierten Version gegenüber. Bei einer Differenz von ca. 50,98ms ist der enorme Performanceunterschied durch ausgelagerte Aufgaben durch das Multithreadingsystem in Unity klar zu erkennen. Der Performanceunterschied in der Framerate je Sekunde liegt weit über nur 10% Leistungssteigerung. Abschließend ist zu sagen, dass die Optimierung durch Multithreading ein unerlässliches Mittel in der modernen Spieleentwicklung und Softwareentwicklung ist.