ParaQuest

Hochleistungsspeicherzugriffsfehler

Alisa Dammer Sven-Hendrik Haase

2013-09-23

Gliederung

Problemstellung

- Grid-basierte 2D Welt mit Kreaturen, die gegeneinander kämpfen
- Einfaches Kampfsystem mit Trefferpunkten, Angriffsstärke und Ausweichchance
- Verschiedene Spezies von Kreaturen: Goblin, Hobbit, Orc, Elf, Dwarf, Human
- Jede Spezies hat andere Werte der Attribute Strength und Agility
- 1 Schritt pro Tick pro Kreatur
- Zusätzlich gesperrte Felder (hier Steine)
- Schicke Visualisierung der Welt

Lösungsansatz

- Welt als sparse Matrix
- Jeden Tick ein Dump der gesamten Welt dieses Prozesses
- Jede Kreatur hat eine 2D Position und eine global eindeutige Id
- Umgesetzt in C++11 mit Visualisierung in Qt sowie cereal für die Serialisierung
- Folgender Ablauf pro Tick: Kreaturen bewegen, Prozesse synchronisieren, Kämpfen, Verlierer löschen, Welt Dump
- Der Welt Dump wird später für die Visualisierung benötigt

Parallelisierungsschema

- Kreaturen werden am Anfang auf Hauptprozess erzeugt und auf andere Prozesse aufgeteilt
- Die Welt wird am Anfang der Simulation spaltenweise aufgeteilt
- Die Aufteilung ist statisch
- Es wird für Kollisionsabfragen eine Extraspalte links und rechts mit übermittelt

Laufzeitmessungen

- 1 Prozess: 285,8s
- 2 Prozesse: 74,4s
- 4 Prozesse: 23,34s
- 8 Prozesse: 11,9s

Leistungsanalyse

- Die meiste Zeit pro Prozess wird in Kollisionsabfragen benötigt
- Senden/empfangen verbringt einen sehr geringen Teil

Skalierbarkeit

- Super-linearer Speedup bis 8 Prozesse
- Danach annährend linearer Speedup