

Modellierung und Simulation 2019/2020

Conway's Game of Life

Louis Donath, Dario Klepoch

Potsdam University

Abstract. Conway's Game of Life ist ein Automat... Hier kommt eine kurze Zusammenfassung des Projektes rein

1 Einfuehrung

2 aasd

3 Implementierung

Bei der Betrachtung der Implementierung werden wir ueber die Implementierung der Randbedingungen sprechen. Weil wir mit unseren ersten Implementierung mit einer unzureichenden Performance hatten, werden wir im zweiten Teil ueber Performanceverbesserungen sprechen.

3.1 Randbedingungen

In unserer Implementierung des GoL haben wir drei unterschiedliche Randbedingungen implementiert. Eine Randbedingung sagt aus wie sich das Spiel verhaelt, wenn auf Zellen ausserhalb des Spielbrettes zugegriffen wird.

Absorbierende Randbedingung

3.2 Performance

Nachbarn ermitteln

Partielle Updates Nach der ersten Implementierung des GoL war die Performance

Numpyarray anstatt Pythonlist Um die Zeit

Weitere Performancesteigerung sind durch sehr viele unterschiedliche Veränderungen möglich. Eine sehr grosse Performancesteigerung ist dadurch möglich eine effizientere Datenstruktur als ein (numpy-)array zu verwenden. In verschiedenen anderen Implementierungen des GoL wird hierfür ein Quadtree benutzt. Ein Quadtree wird meistens dafür verwendet effizient 2-dimensionale Daten zu speichern [1]. Da das GoL auch 2D-Daten sind ist es ein perfekter Anwendungsbereich für einen Quadtree. Mit 'Hashlife' wurde das GoL auf diese Weise implementiert [2].

Um die Performance noch weiter zu steigern ist es möglich den Quadtree parallel aufzubauen. Hier kann entweder die CPU oder auch die GPU benutzt werden. In [3] wurden lineare Quadrees verwendet um einen Quadtree vollständig auf der GPU aufzubauen.

References

1. <https://www.geeksforgeeks.org/quad-tree/>, letzter Zugriff: 27.3.2020
2. <https://en.wikipedia.org/wiki/Hashlife>, letzter Zugriff: 27.3.2020
3. Dupuy, Jonathan & Iehl, Jean-Claude & Poulin, Pierre. (2018). Quadrees on the GPU. 10.1201/9781351052108-12.