**Übung 7 – Verteilte Systeme**

Gires Ntchouayang Nzeunga Matr. Nr. s0554182

Anton Karakochev Matr. Nr. s0553324

1. **Beschreibung der Aufgabe**

In diese Aufgabe sollten wir eine Java-Anwendung entwickeln, bei der wir einen rechenintensiven Algorithmus mit Multi-Threading & Java Future Objekte implementieren.

Wir haben einen Algorithmus zur Erzeugung von Heightmaps unter Benutzung von Simplex Noise denen wir für Computer Grafik programmiert hatten genommen und parellelisiert.

1. **Lösung**

Zur Lösung der Aufgabe haben wir folgende Java-Klassen Implementiert:

* Bibliotheken: JOM L(Java OpenGL Math Library) – für die SimplexNoise funktion
* Eine HeightMapGeneratorMultiThreaded-Klasse die den Heightmap Algorithmus in mehreren Threads (falls nicht angegeben so viele wie es Kerne im Prozessor gibt) ausführt und am Ende zusammensetzt und zurück gibt.
* HeightMapGenerator & ImageWriter – Heightmap Erzeugungsalgorithmus und Hilfsklasse um Heightmap (float array) als bild zu exportieren (Beide Klasser waren bevor wir mit diese Aufgabe angefangen haben vorhanden)

1. Was ist der Unterschied zwischen Threads und Futures?

* A: Threads haben kein Rückgabewert und kein Support für Еxceptions. Mit Futures kann man von dem Future geworfenen Exceptions abfangen sowohl auch einen Rückgabewert haben.

2. Welchen Zustand hat die Variable, wenn das Ergebnis noch nicht bereitsteht?

* A: isDone() == false

3. Wann würde aus Ihrer Sicht diese Alternative einer asynchronen Implementierung zu Callbacks Sinn machen?

* A: Wenn man auf das Ergebnis warten muss, bevor man fortfahren kann.

4. Was ist ein Executor Service?

* A: Ein Interface, das einen Service der Futures Ausführen kann beschreibt. z.B ein Thread Pool.

5. Was ist der Unterschied zwischen den Interfaces Runnable und Callable in Java und an welcher Stelle macht welche Implementierung Sinn? Argumentieren Sie stichpunktartig.

* Callable – Rückgabewert und Exception
* Runnable – Kein Rückgabewert und keine Exceptions

Ausgabe:

Seed: 13088

Dimensions: 1024 x 1024

Running in 4 Threads.

Generate took: 127 ms

Write took: 308 ms

