

Anforderungen

Aktueller Stand	Folgerung
ClientNpcCreator. createNpcsFromBluePrints() darf nur aus dem Main Thread aufgerufen werden. Das wird allerdings noch nicht geprüft	Biete Funktionen, um Ausführung auf bestimmten Threads durch Exceptions (oder Errors?) zu „erzwingen“ - OpenGL Wirft bereits selbst Fehler, aber erst, wenn tatsächlich glFunktionen aufgerufen werden. Vermutlich aber trotzdem nicht sinnvoll?
ChunkCreatorCallable setzt Threadpriorität niedriger Das sollte vermutlich abstrahiert werden?	Biete Möglichkeit Dinge mit bestimmten Prioritäten auszuführen. Soll das über explizite Dekorationen passieren oder über Überladene Methoden?
ChunkStorage nutzt eigene ExecutorServices	Biete einen erweiterten ExecutorService, damit andere Klassen sich nicht um die Erstellung kümmern müssen
ClientChunkStorage nutzt Locks, um den Zugriff auf Maps zu synchronisieren	Ersetze diese Maps durch ConcurrentMaps
Viele Aufrufe in Update führen zu Aufrufen von glFunktionen	Füge Möglichkeiten hinzu, dass Meshes nicht direkt geladen werden, sondern zu einer „Ladefliste“ hinzugefügt werden? – Eigentlich nicht im Scope? Bzw Erstelle eigenen Render Thread, der Funktionen mit OpenGL inhalt ausführt?
Fluid verwendet Thread.sleep	Biete Scheduler an
Neue Möglichkeiten	
Beim Debuggen ist es häufig schwer herauszufinden, welche Aktion den aktuellen Aufruf herbeigeführt hat.	Benenne Threads je nach aktuell ausgeführter Aufgabe?
Asynchrone Aufrufe können aktuell nicht verkettet werden	Biete Completable Futures wo möglich
Aufrufer soll sich nicht um wiederholung kümmern müssen	Biete Scheduler an