# Wochen 1-2

## Was ich gemacht habe:

* Übernahme des bisherigen Stands (Quellcode CA, Quellcode Morpheus, R-Skripte, gesammelte Daten)
* Einarbeitung in das Thema zelluläre Automaten
  + Pixelspiele gelesen (Eigenschaften von CA, Anwendungsgebiete, Algorithmen)
  + Das digitale Universum beschafft und angefangen zu lesen
* Beleg meiner Vorgänger gelesen
* Einarbeitung in das CPM
  + Essay gelesen und Grundbegriffe gegoogelt

## Welche Fragen aufkamen:

* Wie kann man etwas simulieren ohne 100% die biologischen Prozesse zu kennen?
  + *Man muss nicht 100% verstehen, vielleicht versteht man 100% erst durch die Simulation*
* Wie überprüft man ob die Simulation aussagekräftig ist?
  + *Abgleich mit Messungen, Aussagekräftig solange nicht das Gegenteil bewiesen*
* Vergleich von CA und CPM aussagekräftig? (Anzahl der MCS bis 80% Zellabdeckung ein guter Vergleichsparameter?)
  + *Dies ist zunächst ein erster Abgleich, weitere Vergleiche müssen folgen*
* Liefert eine größere Prozessdimension bessere Ergebnisse?
  + Wenn ja, 3D Automat? Threads?
* Lokale vs. Globale Aktualisierung?
* Ist das CPM wissenschaftlich belegt?
* Was soll im Idealfall bei der Arbeit rauskommen?
  + *Entscheidungsbaum, wann CA oder CPM geeignet sind*

## Nächste Schritte:

* Einarbeitung Morpheus
* Das digitale Universum weiterlesen (*iterativ*)
* CPM vollständig verstehen (*iterativ*)

## Perspektivisch:

* *Alpha-Synchrone Aktualisierung (z.B. 10 % aller Zellen werden global aktualisiert, dann wieder 10 % usw. (z.B. 10 mal))*
* *Die Kurve der Zellausbreitung hat irgendwann einen „Knick“. Wann tritt dies ein und warum?*
* *Liefert eine größere Prozessdimension bessere Ergebnisse?*

# Wochen 3-4

## Was ich gemacht habe:

* Entwicklungsumgebung eingerichtet
* CA-Code nachvollzogen
* Mit Morpheus Daten generiert

## Welche Fragen aufkamen:

* Algorithmus für das Zellverhalten sehr einfach gehalten?
* Aktualisierungsart?
* „Eingeschlossene“ Zellen haben keinen weiteren Einfluss auf die Simulation
* Morpheus Umfang?

## Nächste Schritte:

* Restlichen Code verstehen (iterativ)
* Knickpunkt bestimmen (Formel, Simulation)
* Regel fürs Herausschieben
* CPM Teilungsregel überprüfen
* Arbeit lesen mit biologischem Hintergrund

## Perspektivisch:

* *Alpha-Synchrone Aktualisierung (z.B. 10 % aller Zellen werden global aktualisiert, dann wieder 10 % usw. (z.B. 10 mal))*
* *Die Kurve der Zellausbreitung hat irgendwann einen „Knick“. Wann tritt dies ein und warum?*
  + *Mögliche Lösung: Anzahl der handlungsfähigen Zellen ins Verhältnis der Gitterzellen setzen*
* *Liefert eine größere Prozessdimension bessere Ergebnisse?*

# Wochen 5-6

## Was ich gemacht habe:

* Energielevel eingeführt
  + Erste Einschätzungen gemacht
* Biologischen Artikel gelesen
  + Inspiration für zukünftige Versuche geholt
* Morpheusmodell angeschaut
* R-Skripte angeschaut

## Welche Fragen aufkamen:

* Präzisierung der Aufgabenstellung auf Basis des Artikels
  + Siehe handschriftliche Notizen
* Energielevel sinnig?
* Morpheusmodell sinnig?
* Modellvergleich arbeitet nicht auf denselben Voraussetzungen

## Nächste Schritte:

* Artikel ausführlich lesen und nach nützlichen Information durchsuchen
* CA-Projektstruktur überdenken

## Perspektivisch:

* *Vollständige Abbildung der Beobachtungen im Artikel in einem CA*

# Wochen 7-8

## Was ich gemacht habe:

* Artikel nochmal durchgearbeitet
* Entwicklerdokumentation angelegt
* Damit einhergehend Überlegungen zur Projektstruktur gemacht

## Welche Fragen aufkamen:

* Tritt die beschriebene „contact inhibition“ immer auf oder nur wenn etwas mit der Zellkolonie nicht stimmt?
* Wie werden Zellkolonien voneinander abgetrennt?
* Könnte man aus Fig 4 die Teilungswkt. Ermitteln?

## Nächste Schritte:

* Artikel ausführlich lesen und nach nützlichen Information durchsuchen
* CA-Projektstruktur überdenken

## Perspektivisch:

* *Vollständige Abbildung der Beobachtungen im Artikel in einem CA*

# Wochen 9-10

## Was ich gemacht habe:

* Entwicklerdokumentation aktualisiert
* C#-Prototyp angefertigt

## Welche Fragen aufkamen:

* Best practice Zufallszahlengeneratoren
* Best practice Datenaufbereitung/Speicherung um mit Matlab weiterzuarbeiten

## Nächste Schritte:

## Perspektivisch: