

- Frontend

- Was bringt die beste Simulation, ohne dass man sich die Werte angucken kann? Gar nichts. Deswegen haben wir 4 Dateien, die automatisch von dem Script in einem Ordner zusammengefasst werden:

- Log.txt

Die Log.txt ist eine Textdatei die, oh wie wunderbar, den Log der Simulation enthält. Hier wird immer wenn ein Tier stirbt, ein neuer Tag beginnt oder irgendwas anderes passiert, dies in eine neue Zeile eingetragen. Dies sorgt für eine unglaublich große Datei, die vor allem mit kurzen Zeilen gefüllt ist. Deswegen will man sich als Mensch diese Datei nur angucken müssen, wenn man unbedingt muss. Muss man aber nicht, da die wichtigen Informationen auch in den anderen Dateien stehen:

- Info.txt

Info.txt ist schon wieder eine Textdatei, aber diesmal eine deutlich kürzere: Die Info.txt hat nur 5 Zeilen: Eine Datei, die die Laufzeit (die Anzahl der Tage der Simulation), sowie die Anzahl und Gründe der Todesfälle nach Tierart gegliedert auflistet. So kann man aus dieser Datei genau auslesen, wie viele Füchse im Laufe der Simulation an Hunger gestorben sind, oder wie viele Hasen von Füchsen gefressen wurden.

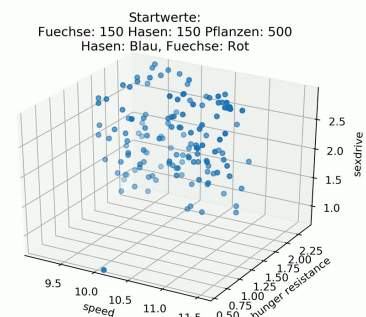
- Viel einfacher für den Menschen anzuschauen sind jedoch die beiden Videos, kreativ benannt: video.avi und video2.avi.

Video.avi ist hier eine Videodatei, die eine zeitliche Auflösung der Mutationen aufzeigt: Hier werden dreidimensionale Graphen dargestellt, auf jeder der Achsen jeweils einer der Werte, die wir mutieren:

Geschwindigkeit, Hungerresistenz und ein Skalar für den Sexualtrieb. Diese sehen ungefähr so aus:

Hier können wir eine sehr frische Zivilisation sehen; die Werte entsprechen noch sehr den ursprünglichen Werten. Wir können an diesen Graphen auch auslesen, dass die geringste Hungerresistenz, die jemals bei einem Tier in dieser Simulation vorhanden war 0.5 ist. Dies erkennt man, da Ober- und Untergrenzen der Achsen so gesetzt sind, dass die geringsten und größten innerhalb einer Simulation verzeichneten Werte die Ober- und Untergrenzen festlegen. Das liegt daran, dass in dem Video 15 Graphen pro Sekunde zu einem Video zusammengeschnitten sind, und diese andernfalls unlesbar wären.

Ähnlich ist auch video2.avi, welches eine visuelle Repräsentation der Simulation darstellt. Die Punkte, die sich nicht bewegen, sind hier Pflanzen, die dunklen bewegend Punkte Hasen und die hellen bewegend Punkte Füchse. Auch dieses Video ist mit 15 fps gerendert, und weil ein Bild jeweils einem Tag entspricht werden in den Videos also immer 15 Tage pro Sekunde dargestellt.



- Analyse

- Warum haben wir unser Ziel nicht erreicht? Wie Jonas schon ausgeführt hat ist es sehr kompliziert, die richtigen Werte zu finden. Was man also machen müsste ist, sehr viele Werte auszuprobieren. Hier kommen jedoch viele Probleme auf informatischer Seite: Zeit. Es dreht sich alles um Zeit. Jeder Tag einer Simulation dauert im Schnitt 0.1 bis 0.4 Sekunden, zur Vorrechnung werde ich den Wert 0.1 benutzen. Da jede Simulation auf grundlegendem Zufall basiert (bei der Bewegung, Mutation etc.), müssen wir also mehrere Simulationen mit den gleichen Werten machen, und die Mittelwerte nehmen, damit wir sicher sagen können, dass eine Wertekombination wirklich gut ist, und nicht einfach nur durch Zufall lange angehalten hat. Mein Computer kann gleichzeitig 3

Simulationen laufen lassen kann (er hat 4 Kerne, also eine Simulation pro Kern und ein Kern für die Kontrolle und Verarbeitung der Informationen), deswegen haben wir pro Werte jeweils 5 Simulationen gemacht, dadurch konnten wir den Fehler durch den Zufall zufriedenstellend beheben, ohne die Zeit pro Wertekombination in die Länge zu ziehen. Das führt dazu, dass jeder Tag in diesen Simulationen letztendlich 0.2 bis 0.8 Sekunden braucht, da ja nicht nur eine Simulation, sondern 5 Simulationen berechnet werden müssen, davon jedoch nur 3 gleichzeitig. Zwei Kerne können nicht an der selben Simulation rechnen.

Machen wir also eine einfache Rechnung: Gehen wir von einer durchschnittlichen Laufzeit von 100 Tagen aus. Dann brauchen wir, um jeweils eine Wertekombination zu testen:  $100 * 0,2s = 20s$  oder, mit dem Größeren Wert,  $100 * 0,8s = 80s$ . Dieser Wert ist für ein Wertepaar. Und jetzt kommt das Problem: Wir wissen nicht, welche Werte gut sind, und welche schlecht. Wir müssen den Computer also raten lassen. Aber wenn jetzt jeder Wert 50 Sekunden (Mittelwert zwischen 20 und 80) dauert, dann sind das mit den Schritten dass der Teste-Script auch noch erkennen muss, ob das gerade gut oder schlecht war, eine Minute pro Wertekombination. Und damit schaffen wir 60 Wertekombinationen pro Stunde-einfach Raten geht also nicht.

Nehmen wir also einen Ansatz aus der AI-Technologie: Machine Learning, genauer: einen Genetic Algorithym. Dieser nimmt letztendlich die beiden besten Wertekombinationen, die er hat, und bastelt aus den beiden ein neues Wertepaar. Dieses wird nun mit einer geringen Wahrscheinlichkeit um einen geringen Wert abgeändert, und getestet. Danach beginnt das ganze von Vorne. Theoretisch ist dieser Anlauf ein sehr guter, und führte bei meinen anderen Anwendungen auch schon mehrfach zu erfolgreichen AIs. Was ich hier nicht bedacht hatte: Genetic Algorithms brauchen tausende generationen, um wirklich ein gutes Ergebnis zu liefern. Und auch nur eintausend Generationen laufen zu lassen hat bei unseren Geschwindigkeiten 12 Stunden gebraucht. Noch dazu ist eine Simulation auszutariieren eine sehr komplizierte Aufgabe. Als Vergleich: Eine AI zu schreiben, und zu trainieren, die Minesweeper Spielt (eine sehr leichte Aufgabe) hatte bei mir mit einem Genetic Algorithym 14.000 Generationen gebraucht, bis es das hinbekommen hat – dank einer Effizienteren Programmiersprache und eines einfacheren Problems habe ich damals aber 500 generationen pro Minute hinbekommen. Bei der Simulation erwarte ich also (im Nachhinein) sogar sechsstellige Generationenzahlen, was (mit dem Wert der Minute Pro Simulaiton) 69,4 Tage dauern würde. Diese Zahl ist wahrscheinlich sogar sehr optimistisch, da mit besseren Werten die Simulationen länger dauern würden, und die maximale Länge der Simulationen auf 10.000 Tage festgelegt ist. Es war also von Anfang an unmöglich, die Simulation auf die Art und Weise, wie wir sie gemacht haben, so auszutariieren, dass wir werte haben, mit denen wir eine Population haben, die überhaupt überlebensfähig ist. Und erst, wenn nicht alle 500 Hasen in den ersten 10 Tagen sterben, können wir untersuchen, wie sich die Hasen verändern, wenn wir z.B. mehr Füchse haben, oder weniger Essen.