UNIVERSITÄT BAYREUTH

Lehrstuhl für Angewandte Informatik III
(Robotik und Eingebettete Systeme)
Prof. Dr. Dominik Henrich

Bachelorpraktikum AI3

Aufgabe 4: Virtuelle Umgebung

Ziel dieser Aufgabe ist das Erstellen, Verwalten, und Zeichnen der virtuellen Umgebung für die Biologiesimulation.

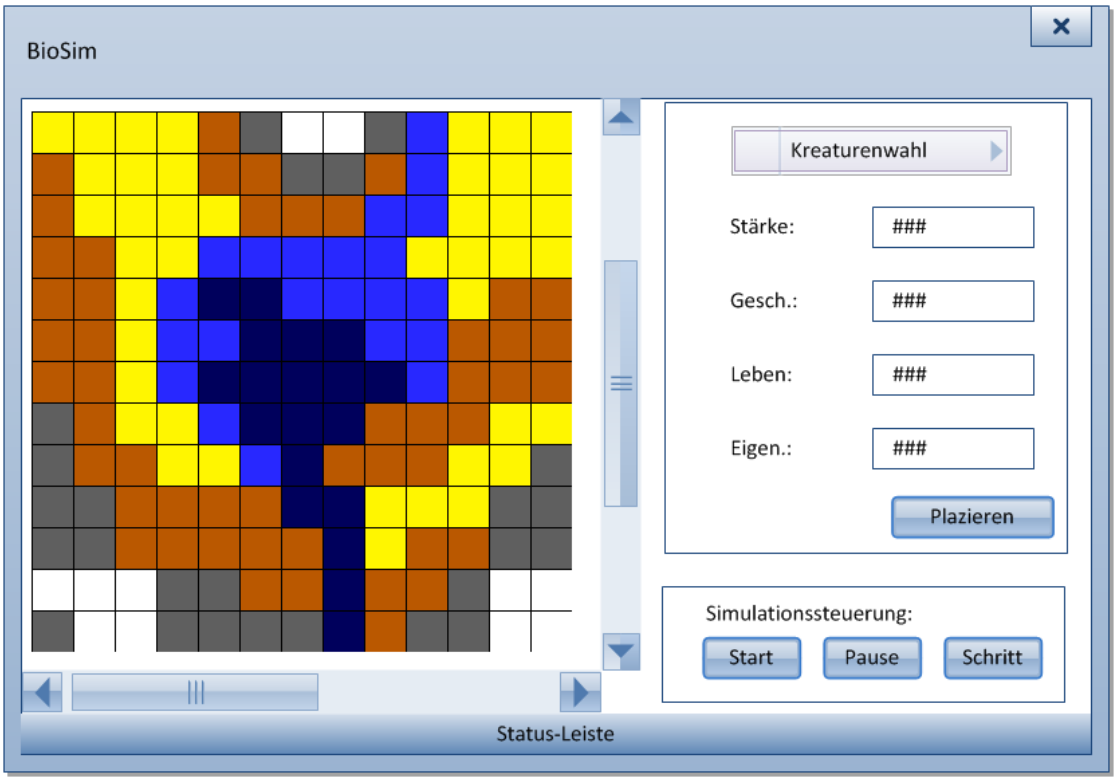
Grundlagen

Die Landschaft ist als ein zweidimensionales Gitter zu realisieren. Jede Gitterzelle entspricht dabei einem Feld („Kachel“), auf dem sich eine oder mehrere Kreaturen aufhalten können. Jede Kachel hat ferner einen zugeordneten Klima-Typ.

Die folgenden sechs Klimata sollen unterstützt werden: Tiefes Wasser, seichtes Wasser, Sand, Erde, Steine, und Schnee.

Die Größe der Landschaft ist nicht fest ins Programm zu integrieren, sondern soll über eine geeignete Variable zur Compilezeit gewählt werden können.

Die folgende Skizze zeigt exemplarisch eine kleine Kachel-Landschaft im Simulationsfenster:

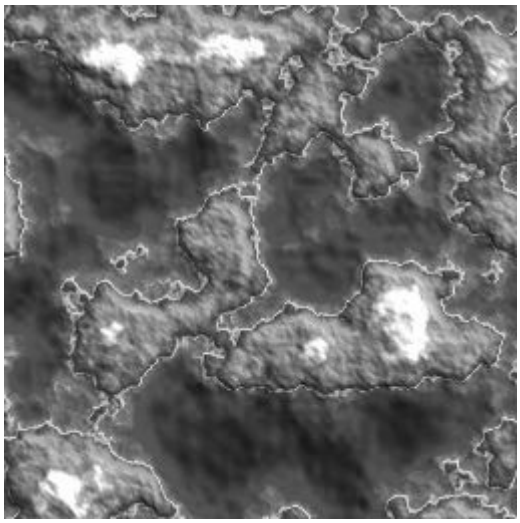


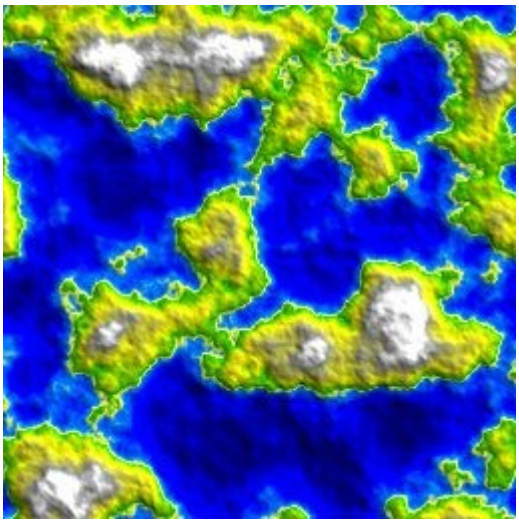
Erstellen der Landschaft

Um bei jedem Start der Simulation einen anderen Simulationsablauf mit jeweils neuen Herausforderungen zu generieren, ist die Landschaft zufällig zu erstellen.

Um ein realistisches Landschaftsbild zu erhalten, ist es jedoch nicht ausreichend, über einen Zufallsgenerator für jedes Gitterfeld einen beliebigen Landschaftstypen zu wählen. Stattdessen muss eine andere, geeignete Zufallsfunktion herangezogen werden. Hier bietet sich die Zufallsfunktion Perlin Noise an.

Die Anwendung von Perlin Noise ergibt im Regelfall für jede Kachel einen einzelnen Ganzzahlwert. Durch passende Assoziation von Wertebereichen mit Klimazonen lassen sich die Ganzzahlwerte schliesslich auf Klimatypen abbilden. Die folgenden Bilder zeigen als Beispiel links ein Ergebnis von Perlin Noise, und rechts eine über Wertebereiche zugeordnete Klima-Einfärbung.





Bildquelle:
http://libnoise.sourceforge.net/tutorials/images/newheightmap.jpg

Verwalten der Landschaft

Für das Verwalten der Landschaft im Arbeitsspeicher ist eine geeignete Datenstruktur zu nutzen.

Dabei ist zu bedenken, dass später auf jedem Feld der virtuellen Landschaft eine oder mehrere Kreaturen stehen können. Ferner wird in späteren Aufgaben (z.B. zur Bahnplanung) ein wahlfreier Zugriff auf einzelne Felder in Abhängigkeit von (x, y)-Positionen erforderlich sein. Achten Sie darauf, eine möglichst speicherkompakte Repräsentation zu wählen, um auch große und sehr große Karten zu unterstützen.

Die Datenstruktur ist gemäß dem Model-View-Presenter-Muster in das Modell der bestehenden Anwendung zu integrieren.

Darstellung der Landschaft

Schliesslich soll die Landschaft im Simulationsfenster der bestehenden GUI-Anwendung dargestellt werden.

Als vorbereitender Schritt müssen bei Programmstart zuerst die Grafiken für die Klimakacheln in den Arbeitsspeicher geladen werden. Dazu ist der bestehende Algorithmus zum Einlesen von Bildern zu nutzen. Wichtig ist hier, jedes Eingabebild nur einmal einzulesen und nur einmal im Speicher zu halten. Passende Klimata-Bilder finden sich im eLearning-Kurs.

Beim Zeichnen der Landschaft wird nun die Datenstruktur aus dem Modell traversiert und über die Benutzeroberfläche dargestellt. Beim Darstellen der Kacheln kann für jedes Vorkommen eines Klimatyps die entsprechende, bereits vorgehaltene Grafik gewählt werden. Zum Zeichnen selbst bietet es sich an, bestehende Grafik-Funktionalität der ausgewählten GUI-Bibliothek zu nutzen. Je nach Bibliothek kann das entweder den Einsatz von OpenGL oder die Arbeit mit bibliotheksspezifischen Funktionen bedeuten.

Bei großen Landschaften darf im Simulationsfenster nur ein Ausschnitt der gesamten Karte dargestellt werden. Um exzessivem Rechenaufwand vorzubeugen, dürfen in diesem Fall nur Kacheln berücksichtigt werden, die sich zumindest teilweise innerhalb des Kartenausschnitts befinden. Damit fallen alle Lösungen weg, die einmalig bei Programmstart eine Grafik des gesamten Terrains erzeugen. Stattdessen muss dynamisch nach jeder Scrolloperation der Bildausschnitt nur mit den aktuell sichtbaren Kacheln gefüllt werden.

Eine Verschiebung des aktuellen Karten-Ausschnitts muss über die Scrollbalken im Simulationsfenster ermöglicht werden. Dabei soll das Verhalten den üblichen GUI-Standards entsprechen. Beispielsweise soll die Größe der Scrollbalken einen Aufschluss über die Grösse der aktuellen Ansicht relativ zur gesamten Umgebung geben. Das Scrolling soll weich auf Pixelbasis erfolgen, nicht mit Kachelsprüngen.

Rahmenbedingungen

Analog zu vorangehenden Aufgaben ist bei der Implementierung auf korrektes C++, vollständige und sinnvolle Dokumentation, sowie saubere Formatierung und Strukturierung zu achten.

Google-Stichwörter

enum, perlin noise, <Name der GUI-Bibliothek> graphics, OpenGL, blitting, tile sets