1 Noise

1.1 Grundlagen

Noise bietet einige Eigenschaften, welche sich sehr gut zur Erzeugung prozedural erzeugtem Content eignen. Von einer einfach erzeugten Folge von Zufallszahlen grenzt sich eine Noise Funktion dadurch ab, dass sie eine Kontinuität aufweist. Durch die, je nach Funktion verschiedenen, Parametern lassen sie sich sehr gut an das gewünschte Ergebnis anpassen. Neben dem geringen Speicherverbrauch einer Implementierung ist besonders die Auswertbarkeit an einer beliebigen Position der Funktion für jede Echtzeitanwendung von entscheidender Bedeutung.

Ziel ist also eine Funktion der Form:

$noise(x) : \mathbb{R}^d \mapsto \mathbb{R}, d \in \mathbb{N}$

1.1.1 Auxilary-Function

Der erste Schritt zur Erzeugung von Noise ist in der Regel eine sog. *Auxilary-Function*[**?**] $\mathbb{S}(k) : Z^n \mapsto \mathbb{R}$ wobei $n \in \mathbb{N}$ ein Skalarfeld mit Dimension n beschreibt. Um mit der Funktion gut weiterarbeiten zu können sollten die Funktionswerte in zwischen -1 und 1 liegen.

Die Wahl der Auxilary-Function ist die erste Entscheidung, welches das Erscheinungsbild der Noise-Funktion entscheident beeinflusst. Hier muss für den jeweiligen Anwendungsfall unterschieden werden. Eine gleichverteilte Zufallsfunktion etwas hätte für eine Landschaft zur Folge, dass die Wahrscheinlichkeit für sehr hohe/tiefe Stellen unnatürlich hoch wäre. Hier bietet sich die Standartnormalverteilung an.

1.1.2 Interpolation und Fade-Function

Um aufbauend auf der Auxilary-Function $S(k)$ eine Funktion $S_1(x) : \mathbb{R} \mapsto \mathbb{R}$ zu definieren wird zwischen zwei nebeneinanderliegenden Werten $g = S(floor(x))$ und $h = S(ceil(x))$ interpoliert. Die Interpolation wird mit einer sog. Fade-Function[**?**] der Form $f(t, g, h)$ mit $t \in [0, 1]$ durchgeführt. Eine Funktion muss mindestens die Kriterien

$f(t_0) = 0 \wedge f(t_1) = 1$ erfüllen, wünschenswert sind jedoch ebenfalls die Eigenschaften $f'(t_0) = 0 \wedge f'(t_1) = 0 \wedge f''(t_0) = 0 \wedge f''(t_1) = 0$ um einen möglichst stetigen Übergang zu schaffen. Der Erfinder des Perlin-Noise Algorithmus benutzt hierfür das Polynom $6x^5 - 15x^4 + 10x^3$[**?**], welches alle genannten Eigenschaften hat und hier auch weiter genutzt werden soll.

Die Wahl dieser Funktion ist wieder ein entscheidender Faktor für das spätere Verhalten der Noise-Funktion.

Aus der Auxilary-Function ergibt sich durch lokale Interpolation mit der *Fade-function*[**?**] zwischen jeweils 2 benachbarten Koordinaten der Auxilary-Function eine Funktion $S_1(x) : \mathbb{R} \mapsto \mathbb{R}$.