

# Prozedurale Generierung

Angewandte Mathematik

## Inhaltsverzeichnis

Prozedurale Generierung .....	1
Problemstellung.....	2
Aufgabe .....	2
Herangehensweise .....	2
Interpolation .....	2
Nach Newton .....	3
Nach Lagrange .....	4
Trassierung .....	4
Darstellung im Raum .....	5
Mit Paraboloiden .....	5
Mit Distanzberechnung .....	6
Schlussgedanke .....	7
Fazit .....	8
Literaturverzeichnis .....	9

## Prozedurale Generierung

Prozedurale Generierung ist ein Verfahren, das unter anderem in Videospielen verwendet wird, um ohne das Zutun von Spielegrafik-Designern automatisch voll funktionsfähige Spielwelten oder Texturen zu generieren. Eines der bekanntesten Videospiele, das diese Technologie verwendet ist „No Man’s Sky“<sup>1</sup>, bei diesem Spiel wird eine unendlich große Galaxie mit Sternen und Planeten inklusive Vegetation und Tierwelt von einem Computerprogramm generiert. Dabei werden eine Vielzahl an mathematischen Funktionen mit unterschiedlicher Gewichtung überlagert, um zum Beispiel Felsformationen in diversen Variationen zu erschaffen. Die Vorteile dieser Technologie liegen auf der Hand. Der erforderliche Aufwand um Spielwelten zu erstellen ist geringer. Das spart Zeit und Kosten. Zudem können mit Hilfe der prozeduralen Generierung Dateigrößen stark reduziert werden. Nachteil dieser Methode ist, dass auch große Probleme auftreten können, wenn zum Beispiel bei der zufälligen Generierung ein Funktionsparameter einen Extremwert erreicht. Das Resultat kann in diesem Fall eine beschädigte oder nicht spielbare Welt sein. Weitere Vertreter dieser Videospiele-Kategorie sind zum Beispiel: „Minecraft“<sup>2</sup>, „Borderlands“<sup>3</sup> oder „Star Citizen“<sup>4</sup>.

---

<sup>1</sup> (Hello Games, 2020)

<sup>2</sup> (Mojang Studios, 2020)

<sup>3</sup> (Gearbox Software, 2020)

<sup>4</sup> (Cloud Imperium Games, 2020)

## Problemstellung

### Aufgabe

Es soll ein neues Videospiel entwickelt werden, in dem der Spieler eine von Meteoritenkratern übersäte Landschaft erkunden kann. Damit es dem Spieler nicht langweilig wird, soll jeder Krater einzigartig sein und sich in keinem Merkmal den anderen ähneln. Jeder einzelne Krater soll mit derselben mathematischen Funktion generiert werden wobei nach jedem Durchlauf die Anfangsparameter für Form, Durchmesser und Höhe, in einem zuvor festgelegtem Wertebereich verändert werden.

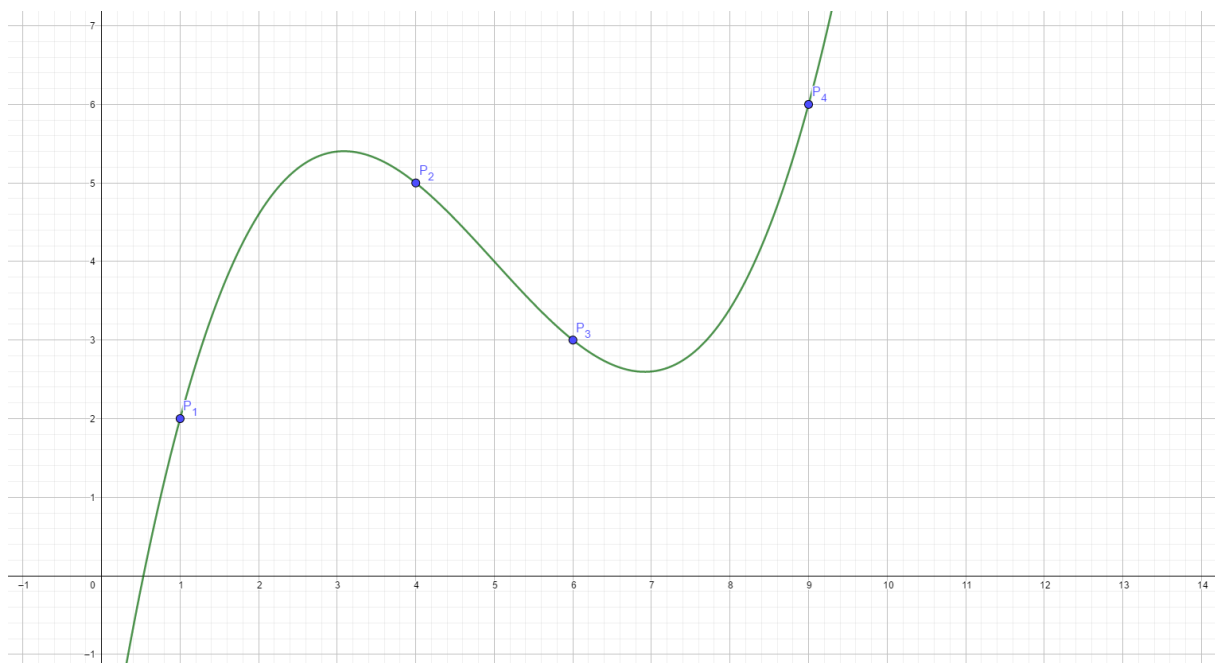
### Herangehensweise

Um die Spieleentwicklung zu erleichtern wird eine Game-Engine verwendet, die schon alle Grundfunktionen für das Erstellen und Rendern von Spielobjekten beinhaltet. Bei der Game-Engine handelt es sich um die Unity-Engine<sup>5</sup>, die – wie der Homepage des Herstellers zu entnehmen ist – die weltbeste Echtzeit-Entwicklungsplattform für Videospiele darstellt und ein robustes Ökosystem für die Verwirklichung einer Vielzahl an Projekten bietet. Unter dem Begriff „Rendern“ versteht man den Vorgang, bei dem alle im Spiel vorhandene Objekte mit Hilfe der Grafikkarte auf dem Monitor angezeigt werden.

Im folgenden Teil werden verschiedene Mathematische Verfahren beschrieben mit deren Hilfe eine Mathematische Funktion aufgestellt werden kann, die die in der Aufgabenstellung vorgegebenen Kriterien erfüllt.

## Interpolation

Interpolation ist ein Verfahren in der Mathematik mit dem aus einer Reihe an bekannten Punkten ein Polynom errechnet werden kann, welches diese verbindet. Es gibt verschiedene Arten ein Interpolationspolynom zu bilden. Zwei davon werden im folgenden Teil genauer erläutert. Es ist außerdem anzumerken, dass für jede Kombination an Punkten immer nur genau ein einziges Polynom existiert, das alle Punkte verbindet.



Ziel der Interpolation ist es, einen Graphen zu finden, der bekannte Punkte wie zum Beispiel Messwerte kontinuierlich miteinander verbindet. In obenstehendem Koordinatensystem<sup>6</sup> sind

<sup>5</sup> (Unity Technologies, 2020)

<sup>6</sup> (Weber, ProcGen - GeoGebra, 2020)

sowohl die Punkte als auch die resultierende Funktion eingezeichnet. Dieser Graph lässt sich nicht nur grafisch, sondern auch algebraisch herleiten.

### Nach Newton<sup>7</sup>

Das Interpolationsverfahren nach Newton folgt dem Prinzip der dividierten Differenzen um die konstanten Koeffizienten  $A_i$  zu berechnen.  $f_n$  ist die Summe aller Koeffizienten multipliziert mit den Basisfunktionen, die sich aus den Linearfaktoren zusammensetzen.

$$f_n(x) = \sum_{i=0}^n A_i \cdot N_i(x) \text{ mit } N_0(x) = 1 \text{ und } N_i(x) = (x - x_0) \cdot (x - x_1) \cdot \dots \cdot (x - x_{i-1})$$

Um die Vorgehensweise bei diesem Verfahren nachvollziehen zu können, werden die einzelnen Schritte anhand eines Beispiels veranschaulicht. Zu Beginn müssen die Punkte, durch die das Polynom verlaufen soll, festgelegt werden.

$$P_1(1|2); P_2(4|5); P_3(6|3); P_4(9|6)$$

Als nächstes wird eine Tabelle mit den einzelnen  $x$ - und  $y$ -Koordinaten erstellt und die Koeffizienten Schritt für Schritt berechnet. Zuerst wird die  $x$ -Koordinate des zweiten Punkts unter und die  $y$ -Koordinate über den Bruchstrich geschrieben. Darauf folgt ein Minus und anschließend die  $x$ -Koordinate des ersten Punkts unter und die  $y$ -Koordinate über dem Bruchstrich. In der nächsten Spalte werden anstatt der  $y$ -Koordinaten der Punkte die Ergebnisse der vorherigen Berechnungen verwendet. Als  $x$ -Werte verwendet man nur die äußersten beiden von denen, die bis jetzt in die Berechnung mit eingeflossen sind.

$x_k$	$y_k$			
1	2			
		$\frac{5-2}{4-1} = 1$		
4	5		$\frac{-1-1}{6-1} = -\frac{2}{5}$	
		$\frac{3-5}{6-4} = -1$		$\frac{\frac{2}{5} - (-\frac{2}{5})}{9-1} = \frac{1}{10}$
6	3		$\frac{1-(-1)}{9-4} = \frac{2}{5}$	
		$\frac{6-3}{9-6} = 1$		
9	6			

Im letzten Schritt muss nur noch aus den Koeffizienten, die den vier rot markierten Zahlen entsprechen, ein Polynom gebildet werden. Dabei steht der erste Koeffizient immer allein und bei allen weiteren wird jedes Mal ein Linearfaktor mehr angehängt. Anschließend werden alle Funktionsteile addiert. Zu beachten ist auch, dass aus dem  $x$ -Wert des letzten Punktes kein Linearfaktor mehr gebildet wird.

$$f_3(x) = 2 + 1 \cdot (x - 1) + \left(-\frac{2}{5}\right) \cdot (x - 1) \cdot (x - 4) + \frac{1}{10} \cdot (x - 1) \cdot (x - 4) \cdot (x - 6)$$

$$f_3(x) = \frac{1}{10}x^3 - \frac{3}{2}x^2 + \frac{32}{5}x - 3$$

<sup>7</sup> (Dr.-Ing. Bartsch, Interpolationsformeln, 1984)

## Nach Lagrange<sup>8</sup>

Beim Interpolationsverfahren nach Lagrange werden mit Hilfe der  $x$ -Werte aller gegebenen Punkte die Basisfunktionen bestimmt.  $f_n$  ist die Summe aller  $y$ -Koordinaten multipliziert mit den Basisfunktionen, die sich aus den  $x$ -Werten zusammensetzen.

$$f_n(x) = \sum_{i=0}^n y_i \cdot N_i(x) \text{ mit } N_i(x) = \frac{(x - x_0) \cdot (x - x_1) \cdot \dots \cdot (x - x_{i-1})(x - x_{i+1}) \cdot \dots \cdot (x - x_n)}{(x_i - x_0) \cdot (x_i - x_1) \cdot \dots \cdot (x_i - x_{i-1})(x_i - x_{i+1}) \cdot \dots \cdot (x_i - x_n)}$$

Um die Vorgehensweise auch bei diesem Verfahren nachvollziehen zu können, werden die einzelnen Schritte anhand desselben Beispiels veranschaulicht. Zu Beginn müssen wieder die Punkte, durch die das Polynom verlaufen soll, festgelegt werden.

$$P_1(1|2); P_2(4|5); P_3(6|3); P_4(9|6)$$

Im zweiten Schritt ist es sinnvoll die  $x$ -Koordinaten untereinander aufzulisten, um die Basisfunktionen einfacher berechnen zu können. Dann werden die einzelnen Basisfunktionen nach dem obigen Schema berechnet. Zu beachten ist, dass es bei vier gegebenen Punkten auch vier Basisfunktionen gibt. Diese bestehen allerdings insgesamt nur aus drei Faktoren.

$x_n$	
1	$N_0 = \frac{x-4}{1-4} \cdot \frac{x-6}{1-6} \cdot \frac{x-9}{1-9} = -\frac{1}{120} \cdot (x-4) \cdot (x-6) \cdot (x-9)$
4	$N_1 = \frac{x-1}{4-1} \cdot \frac{x-6}{4-6} \cdot \frac{x-9}{4-9} = \frac{1}{30} \cdot (x-1) \cdot (x-6) \cdot (x-9)$
6	$N_2 = \frac{x-1}{6-1} \cdot \frac{x-4}{6-4} \cdot \frac{x-9}{6-9} = -\frac{1}{30} \cdot (x-1) \cdot (x-4) \cdot (x-9)$
9	$N_3 = \frac{x-1}{9-1} \cdot \frac{x-4}{9-4} \cdot \frac{x-6}{9-6} = \frac{1}{120} \cdot (x-1) \cdot (x-4) \cdot (x-6)$

Als letztes muss nur noch aus den errechneten Basisfunktionen ein Polynom gebildet werden. Dabei werden diese immer mit der entsprechenden  $y$ -Koordinate multipliziert und anschließend miteinander addiert. Das Ergebnis ist überraschenderweise dasselbe wie oben.

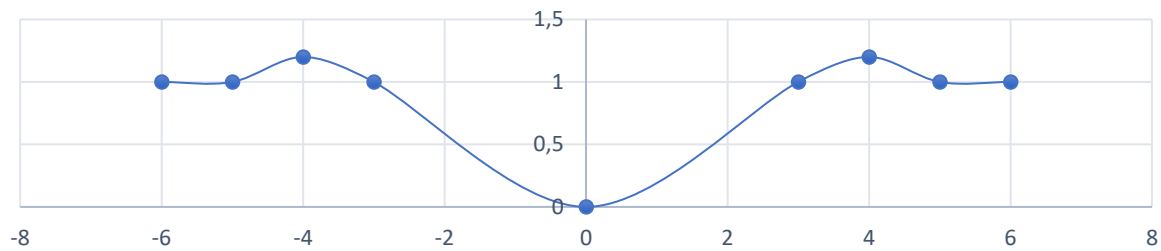
$$f_3(x) = \left[ 2 \cdot \left( -\frac{1}{120} \right) \cdot (x-4) \cdot (x-6) \cdot (x-9) \right] + \left[ 5 \cdot \frac{1}{30} \cdot (x-1) \cdot (x-6) \cdot (x-9) \right] \\ + \left[ 3 \cdot \left( -\frac{1}{30} \right) \cdot (x-1) \cdot (x-4) \cdot (x-9) \right] + \left[ 6 \cdot \frac{1}{120} \cdot (x-1) \cdot (x-4) \cdot (x-6) \right] \\ f_3(x) = \frac{1}{10}x^3 - \frac{3}{2}x^2 + \frac{32}{5}x - 3$$

## Trassierung

Bei der Trassierung wird keine alleinstehende Funktion gebildet, sondern mehrere einzelne, die einen zusammenhängenden Graph erzeugen. Diese Art von Funktionen wird allgemein als „abschnittsweise definierte Funktionen“ bezeichnet. Für die in der Aufgabenstellung geforderten Eigenschaften der Generierungs-Funktion ist dieser Lösungsansatz der beste. In folgendem Diagramm wird ein möglicher Querschnitt eines Meteoritenkraters dargestellt.

<sup>8</sup> (Dr.-Ing. Bartsch, Interpolationsformeln, 1984)

## Meteoritenkrater



Im folgenden Beispiel wird der Vorgang zur Berechnung einer der Teilfunktionen genauer erläutert. Zu Beginn ist es sinnvoll sich bekannte Eckdaten der Anfangs- und Endpunkte sowie der dort vorherrschenden Steigung zu notieren. Der Graph wird im Bereich  $x \in [-3; 0]$  betrachtet.

$$f(-3) = 1; \quad f(0) = 0; \quad f'(-3) = -1; \quad f'(0) = 0$$

Mit Hilfe der vier gegebenen Werte kann nun ein Polynom dritten Grades bestimmt werden. Dabei werden die ersten beiden in die Grundform eingesetzt und der dritte und vierte Wert in die Ableitung der Grundform. Anschließend werden die aufgestellten Gleichungen vereinfacht.

$$a \cdot (-3)^3 + b \cdot (-3)^2 + c \cdot (-3) + d = 1$$

$$-27a + 9b - 3c + d = 1$$

$$a \cdot 0^3 + b \cdot 0^2 + c \cdot 0 + d = 0$$

$$d = 0$$

$$3 \cdot a \cdot (-3)^2 + 2 \cdot b \cdot (-3) + c = -1$$

$$27a - 6b + c = -1$$

$$3 \cdot a \cdot 0^2 + 2 \cdot b \cdot 0 + c = 0$$

$$c = 0$$

Nun kann das aufgestellte Gleichungssystem beispielsweise mit dem Gauß-Verfahren gelöst werden. Zum Schluss ergibt sich die gewünschte Teilfunktion.

$$f(x) = -\frac{1}{27}x^3 \quad \text{für } x \in [-3; 0]$$

## Darstellung im Raum

Nachdem die Grundfunktion bzw. die Grundfunktionen berechnet wurden, müssen diese im Raum dargestellt werden. Dafür gibt es in der Welt der Spieleentwicklung grundsätzlich zwei Möglichkeiten. Die Erste nutzt den mathematisch korrekten Weg mit Hilfe von Paraboloiden Gleichungen und die zweite Möglichkeit ist so gesehen aus der Schreibfaulheit von Softwareentwicklern heraus entstanden und stützt sich auf den Fakt, dass Meteoritenkrater in den meisten Fällen symmetrisch sind.

### Mit Paraboloiden<sup>9</sup>

Paraboloiden bestehen sozusagen aus zwei Parabel Gleichungen oder in diesem Fall aus zwei verschiedenen Polynomen, eines verläuft entlang der  $x$ -Achse und das andere entlang der  $z$ -Achse. Werden nun beide Teile addiert, ergibt sich die  $y$ -Koordinate. Es ist allerdings anzumerken, dass die Achsenbeschriftungen variieren können, je nachdem wie das Koordinatensystem orientiert ist. In der Unity-Engine ist die  $y$ -Achse nach oben orientiert. Außerdem ist es notwendig, dass die beiden Funktionen addiert und nicht subtrahiert werden, da eine quadratische Funktion im dreidimensionalen Raum sonst nicht trichterförmig bzw.

<sup>9</sup> (Dr.-Ing. Bartsch, Das Paraboloid, 1984)

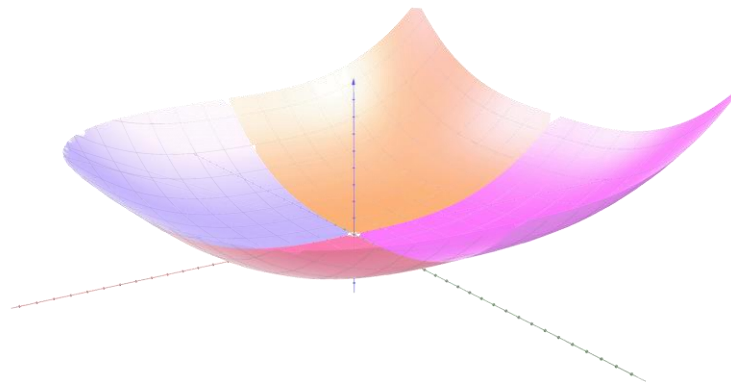
elliptisch ist, sondern eher einem Sattel gleicht. Diese Form nennt sich dann ein hyperbolisches Paraboloid.

$$f(x; z) = P(x) + P(z)$$

Da der Meteoritenkrater symmetrisch ist, sind die beiden Funktionen für die  $x$ - und  $y$ -Achse identisch also ergibt sich folgende Funktionsgleichung.

$$f(x; z) = -\frac{1}{27}x^3 + \left(-\frac{1}{27}z^3\right) \text{ für } x \in [-3; 0] \text{ und } z \in [-3; 0] \text{ bei } \mathbb{W} = [0; 1]$$

Einige wichtige Details dürfen allerdings nicht übersehen werden, denn die obige Funktion beansprucht nur ein Viertel des gesamten Kraters. Also muss sie entsprechend gespiegelt oder gedreht werden, um einen vollständigen Krater darstellen zu können. Außerdem sind Funktionen ab dem dritten Grad nicht mehr kreisförmig, wenn sie von oben betrachtet werden. Für die Darstellung eines Meteoritenkraters ist das allerdings notwendig. Des Weiteren muss zusätzlich zur Begrenzung der Werte für  $x$  und  $z$  auch noch ein Maximum für die  $y$ -Koordinate festgelegt werden, da der Krater sonst ebenfalls nicht rund erscheint. Aber auch die Festlegung eines Mittelpunktes für den Meteoritenkrater bereitet sich bei dieser Methode als schwieriger im Vergleich zur nächsten Methode.



In obenstehendem 3D-Modell<sup>10</sup> sind alle vier Teilfunktionen vorhanden. Bei zwei davon wurde kein Maximalwert für die  $y$ -Koordinate festgelegt, um den Unterschied deutlich zu machen.

### Mit Distanzberechnung<sup>11</sup>

Aufgrund der gerade genannten Probleme bei der Darstellung mit Hilfe von Paraboloiden greifen Softwareentwickler häufig auf eine andere Methode zurück. Dabei wird der Mittelpunkt des Meteoritenkraters im Vorfeld beispielsweise auf  $M(0|0|0)$  festgelegt und anstatt zwei Funktionen zu verwenden und zu addieren, wird nur eine einzige verwendet. Um den Funktionsgraphen nun richtig darstellen zu können, wird die Differenz der  $x$ - und  $z$ -Koordinaten gebildet und mit Hilfe des „Satz des Pythagoras“<sup>12</sup> die Entfernung zum Mittelpunkt berechnet. Anzumerken ist allerdings, dass die Distanzberechnung auf der Ebene  $y = 0$  stattfindet.

$$d_M(x; z) = \sqrt{(x - x_M)^2 + (z - z_M)^2}$$

Anschließend wird auf den resultierenden Wert die entsprechende Funktion angewandt, wobei eventuell Vorzeichen angepasst werden müssen, sodass der Krater weiterhin nach oben geöffnet ist.

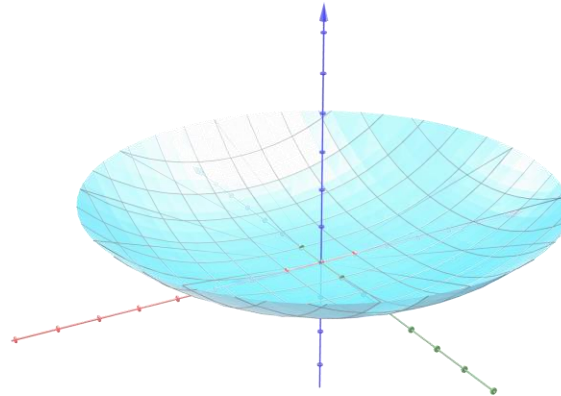
$$f(x; z) = \frac{1}{27} \left( \sqrt{(x - 0)^2 + (z - 0)^2} \right)^3$$

<sup>10</sup> (Weber, 3DGraph - GeoGebra, 2021)

<sup>11</sup> (Dr. Altrichter, et al., 2017)

<sup>12</sup> (Dr. Huber, et al., 2019)

Das Resultat ist ein kreisrunder Krater<sup>13</sup>, der exakt der vorgegebenen Funktion entspricht. Wie schon bei der ersten Methode angesprochen, muss in diesem Fall auch ein Maximalwert für die  $y$ -Koordinate festgelegt werden, da der Funktionsgraph sonst bis in die Unendlichkeit weiter geht. In vielen Programmiersprachen gibt es allerdings schon vorgefertigte Funktionen für Fälle wie diesen, deshalb wird diese Methode über der anderen präferiert.



## Schlussgedanke

Wenn nun alle Teilfunktionen mit Hilfe dem unter dem Punkt Trassierung beschriebenen Verfahren bestimmt wurden, kann der gesamte Graph dargestellt werden. Anschließend ist noch etwas Programmiergeschick gefragt, um den Krater in der Unity-Engine darzustellen. Um die Form, den Durchmesser und die Höhe eines Kraters verändern zu können, müssen nur die im Punkt Trassierung beschriebenen Anfangsparameter verändert werden. Die Position kann ebenfalls sehr leicht verändert werden, in dem – wie im Punkt Distanzberechnung beschrieben – der Mittelpunkt vorher festgelegt wird.



Damit der Einschlagskrater möglichst realistisch aussieht, wurde nachträglich noch eine Rauschfunktion, auch genannt „Perlin noise“ bzw. „Simplex noise“ hinzugefügt, um die Landschaft um den Krater herum anschaulicher zu machen. Außerdem wurde ein Simulationsprogramm für hydraulische Erosion angewandt, um witterungsbedingte Einwirkungen auf die Umgebung darzustellen wie zum Beispiel die Abtragung von Sediment durch Regen oder Wind.

<sup>13</sup> (Weber, 3DGraph2 - GeoGebra, 2021)

## Fazit

Zusammenfassend kann man sagen, dass das Verfahren der Trassierung kombiniert mit der Darstellungsmethode der Distanzberechnung hervorragend für die prozedurale Generierung von Meteoritenkratern geeignet ist. Oft werden in der Programmierung aber auch nur simple quadratische Funktionen verwendet, um Ressourcen zu schonen und Berechnungszeiten zu verkürzen. Außerdem wird in Computersystemen zum Lösen von Ebenengleichungen beispielsweise kein Gauß Verfahren verwendet, sondern eine computerfreundlichere Alternative.

Zusätzliche Materialien sowie die entwickelte Software finden sich unter:

<https://github.com/FelixDoubleu/Unity-Procedural-Generation><sup>14</sup>

<https://www.youtube.com/watch?v=9P3s0ULxxa0>

<https://www.youtube.com/watch?v=p0W0JB6NXm8>

<https://www.youtube.com/watch?v=Z6NpyJcl23M>

---

<sup>14</sup> (Weber, FelixDoubleu - Unity-Procedural-Generation, 2021)



## Literaturverzeichnis

Cloud Imperium Games. (20. Januar 2020). *Homepage (en)*. Von cloudimperiumgames: <https://cloudimperiumgames.com/> abgerufen

Dr. Altrichter, V., Fielk, W., Ioffe, M., Körner, D., Konstandin, S., Meier, P., . . . Roßmann, F. (2017). Abstand zwischen zwei Punkten. In V. Dr. Altrichter, W. Fielk, M. Ioffe, D. Körner, S. Konstandin, P. Meier, . . . F. Roßmann, *Mathematik Berufliche Oberschule Bayern Technik Band 1* (S. 189-192). Berlin: Cornelsen Verlag GmbH.

Dr. Huber, C., Fritzlar, K., Dr. Graupner, R., Schiller, C., Hoffmann, B., & Dr. Haitzer, M. (2019). Merkhilfe Mathematik. In C. Dr. Huber, K. Fritzlar, R. Dr. Graupner, C. Schiller, B. Hoffmann, & M. Dr. Haitzer, *Formelsammlung* (S. 9). München: Kastner AG.

Dr.-Ing. Bartsch, H.-J. (1984). Das Paraboloid. In H.-J. Dr.-Ing. Bartsch, *Taschenbuch Mathematischer Formeln* (S. 336-338). Leipzig: VEB Fachbuchverlag Leipzig für den Verlag Harri Deutsch ▪ Thun.

Dr.-Ing. Bartsch, H.-J. (1984). Interpolationsformeln. In H.-J. Dr.-Ing. Bartsch, *Taschenbuch Mathematischer Formeln* (S. 159-161). Leipzig: VEB Fachbuchverlag Leipzig für den Verlag Harri Deutsch ▪ Thun.

Gearbox Software. (20. Januar 2020). *Homepage (en)*. Von Gearboxsoftware: <https://www.gearboxsoftware.com/> abgerufen

Hello Games. (20. Januar 2020). *Homepage (en)*. Von Hellogames: <https://hellogames.org/> abgerufen

Mojang Studios. (20. Januar 2020). *Homepage (de)*. Von Minecraft: <https://www.minecraft.net/de-de/> abgerufen

Unity Technologies. (20. Januar 2020). *Homepage (de)*. Von Unity: <https://unity.com/de> abgerufen

Weber, F. (20. Januar 2020). *ProcGen - GeoGebra*. Von GeoGebra: <https://www.geogebra.org/classic/eemfhdpp> abgerufen

Weber, F. (20. Februar 2021). *3DGraph - GeoGebra*. Von GeoGebra: <https://www.geogebra.org/classic/scccwrkw> abgerufen

Weber, F. (20. Februar 2021). *3DGraph2 - GeoGebra*. Von GeoGebra: <https://www.geogebra.org/classic/zr62sgch> abgerufen

Weber, F. (13. März 2021). *FelixDoubleu - Unity-Procedural-Generation*. Von GitHub: <https://github.com/FelixDoubleu/Unity-Procedural-Generation> abgerufen