3D

Automatische 3D-Objektmodellierung

Luka Petkovic & Ben Geisler

Facharbeit 9spez

Inhaltsverzeichnis

Motivation

Wir beschäftigen uns schon längere Zeit mit 3D Objektmodellierung. Dabei gilt jedoch, dass die Objekte so lebensecht wie möglich aussehen müssen.

|  |  |
| --- | --- |
|  |  |

Beide Donuts sind modelliert. Der linke Donut ist jedoch detailreicher und sieht somit echter aus. So einen Donut mit Kaffee zu modellieren kostet etwa 9 Stunden. Deshalb wäre ein 3D-Scanner nützlicher, da man dadurch viel mehr Zeit sparen würde. Die Scanner auf den Markt sind jedoch unterschiedlich. Es gibt welche für 100€ bis 5000€. Deshalb haben wir uns als Aufgabe gesetzt, einen funktionalen und vor allem billigeren 3D-Scanner mit passender Modellierungssoftware selber zu bauen und zu erstellen.

Verwendung und Bedeutung eines 3D Scanners

3D-Scanner werden häufig von Hobbybastlern, Personen mit einem 3D-Drucker und Filmproduzenten genutzt. Mithilfe eines guten und effizienten Scanners ist es möglich, schnell und einfach detailreiche Gegenstände zu modellieren. Das Resultat ist dann per 3D-Visualisierungssoftware (zum Beispiel Unity oder Blender) aufrufbar. Somit könnten zum Beispiel Filmproduzenten eine Miniaturlandschaft bauen, diese dann einscannen und am Schluss vergrößern und als Fläche in einem Film als Kulisse verwenden. Für Personen mit einem 3D Drucker ist ein Scanner von Vorteil, um ein Objekt zu duplizieren. Mechanische Anlagen können einfach via Kraft-Simulationen am Computer getestet werden, ohne diese im realen Leben zu beschädigen. Dabei können schnell und einfach Schwachstellen gefunden werden.

Der 3D Scanner wurde in den 80ern entwickelt. Dort wurden häufig Kontaktproben gegen das Objekt gedrückt. Um das Objekt herum ist somit eine Art Maschennetz entstanden. Zwischen Kontaktproben, welche die Maschen bildeten, wurde die Entfernung gemessen. Mithilfe dieser Entfernungen wurde dann am Computer ein 3D Modell erstellt. Laserscanning und Fotogrammmetrie sind die Haupttechniken.   
Ein Laser besitzt ein oder zwei Laser und oftmals eine Kamera. Für kleiner Gegenstände wird eine Drehplatte mit eingebaut. Die Laser messen den Abstand zum Objekt. Die Kamera (falls vorhanden) zeichnet die Deformierung des Objektes auf. Der Winkel zwischen den Sensoren, der Kamera und das Objekt, sowie die Entfernungen dienen dazu, ein digitales Gitternetz (Mesh) am Computer zu generieren. Ein Nachteil ist, dass die Laser und Kameras bei reflektierenden Flächen Messfehler haben. Bei verschiedenen Farben gilt dies auch. Das ist jedoch zu gering um es überhaupt zu merken. Desweitern sind billigere 3D Scanner (unter 100€) nicht so genau, wenn es um Details geht.

Auswahl der Programmiersprache, Visualisierungssoftware und IDE

Die in dieser Arbeit genutzte Programmiersprache ist C#. Diese wurde aufgrund der Kompatibilität mit Unity gewählt. Des Weiteren beherrschen die Autoren dieser Arbeit die Sprache bereits recht gut, weshalb die Recherche und das Sammeln von Informationen erleichtert wird. Außerdem ist diese Sprache weit verbreitet, weswegen Anleitung zu einem bestimmten Thema leicht zu finden sind.  
Als Visualisierungssoftware für die Erstellung des Meshs wird Unity verwendet. Unity besitzt den Vorteil, dass sie leicht zu benutzen und anfängerfreundlich ist. Auch das Unity die meist benutzte Game-Engine ist, ist ein dazu beitragender Faktor.

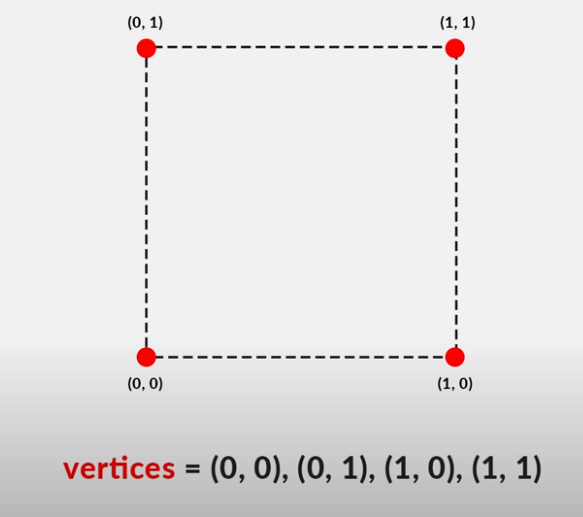
Informatische Grundlagen

In diesem Teil werden die Grundlagen zur Mesherstellung in Unity…. erläutert.

Mesherstellung in Unity

In Unity sind Eckpunkte des zu erstellenden Meshs und Dreiecke, die durch das Verbinden der Eckpunkte entstehen, notwendig, um ein Mesh darstellen zu können. Beide werden als ein Array dargestellt.

Um beispielsweise ein Quadrat in Unity zu erstellen, benötigt man vier Eckpunkte mit jeweils 2 Koordinaten, der x-Achse und y-Achse.



Danach werden die Dreiecke erstellt. Ein Quadrat wird beispielsweise in zwei Dreiecke unterteilt, dabei werden den Eckpunkten des Quadrates der Reihenfolge nach Zahlen zugeordnet.  
Gleichzeitig muss auf die Aufstellung des Arrays geachtet werden. Bei einer falschen Reihenfolge kann es zum sogenannten Backface-Culling kommen. Das bedeutet, dass die Dreiecke des Meshs nur auf einer Seite gerendert werden. Die andere Seite wird nicht angezeigt (Normalerweise wird sie auch nicht benötigt). Die meisten Visualisierungssoftwares, wie Unity, nutzen diesen Prozess aus Leistungsgründen. Richtig wäre, die Punkte im Uhrzeigersinn in das Array hinzuzufügen.

|  |  |
| --- | --- |
| Pfeil kreis dünne linie | Kostenlose Icon |  |

Übertragung der Daten von Arduino nach Unity

Hierfür wurde ein bereits existierendes Programm von GitHub verwendet.   
Zunächst wird der Port festgelegt an dem der Arduino mit dem Computer verbunden ist. Danach folgt die Start-Funktion, in der der Serial Stream des Arduinos geöffnet wird. In der nächsten Funktion werden die Informationen ausgelesen und die alten Daten gelöscht. Des Weiteren werden die Daten in einzelne Strings mithilfe der *.Split* Methode aufgeteilt. Als nächstes wird überprüft, ob alle Werte erhalten wurden, falls dies zutrifft, ist es noch notwendig die Daten in floats oder in ints umzuwandeln, um sie für unsere Zwecke richtig nutzen zu können.