
BACHELORARBEIT

Herr

Andreas Hauswald

Workflowoptimierung für mobile AR-Geräte anhand der Microsoft HoloLens

Mittweida, 2017

Fakultät Computer und Biowissenschaften

BACHELORARBEIT

Workflowoptimierung für mobile AR-Geräte anhand der Microsoft HoloLens

Autor:
Herr

Andreas Hauswald

Studiengang:
Medieninformatik und interaktives Entertainment

Seminargruppe:
MI13w2-B

Erstprüfer:
Professor Alexander Marbach

Zweitprüfer:
Diplom Medieninformatiker (FH) Stefan Fibig

Einreichung:
Mittweida, 23.September.2017

Verteidigung/Bewertung:
Mittweida, 2017

Abstract

Diese Bachelorarbeit zielt darauf ab, die Vor- und Nachteile des Game Asset Creation Workflows und des Textureless Workflows, für mobile Augmented Reality Geräte anhand der Microsoft HoloLens zu ermitteln. Hierfür wurden beide Workflows in Sachen Performance und Erscheinungsbild mithilfe einer Testreihe verglichen. Es wurden für jeden Workflow jeweils drei Modelle mit unterschiedlichen Parametern erstellt. In der Testreihe wurden daraufhin verschiedene Texturen separat getestet, um genau sagen zu können, welche Texturen für Performanceeinbrüche sorgen. Das Ergebnis der Testreihe bestätigt die vorher getroffene These, dass der Textureless Workflow über die bessere Performance verfügt. Diese Bachelorarbeit ist sowohl für Modellierer von 3D Modellen für mobile Augmented Reality Geräte interessant, als auch für HoloLens Entwickler.

This bachelor thesis aims to discover the advantages and disadvantages of the Game Asset Creation Workflow and the Textureless Workflow for mobile augmented reality devices, based on the Microsoft HoloLens. For this, a test series was developed and both workflows have been tested in terms of performance and looks. For each workflow, three 3D Modells with different parameters have been developed. Within the test series, different textures have been tested separately. This was done, to determine which textures are not handled well by the HoloLens. The result of the test series approved, that the Textureless Workflow had the overall better performance. This bachelor thesis is interesting for 3D artist, which are creating content for mobile augmented reality devices, as well as for HoloLens developers.

Inhaltsverzeichnis

Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
Abkürzungsverzeichnis	vi
Glossar.....	vii
1. Grundlegendes	1
1.1. HoloLens – eine Technik für die Zukunft.....	1
1.2. Abgrenzung der Themenstellung	2
1.3. Aufbau der Arbeit	3
1.4. Ziel der Arbeit.....	3
1.5. Technische Limitierungen der Microsoft HoloLens	4
1.6. Verwendete Workflows	6
1.6.1. Game Asset Creation Workflow.....	6
1.6.2. Textureless Workflow	11
1.7. Technischer Hintergrund bei der Auswahl der Workflows.....	13
2. Erarbeitung des praktischen Teils	15
2.1. Untersuchungsdesign	15
2.2. Erstellte Modelle	16
3. Entwicklung des Testverfahrens.....	20
3.1. Licht im Testverfahren.....	20
3.2. 3D Modelle im Testverfahren	21
3.3. Das Testszenario	22
3.3.1. Äußere Faktoren.....	22
3.4. Vortests für das Testen mit dem Unity Profiler	23
3.5. Aufbau des Profiler Tests.....	24
3.6. Polygonperformance	24

3.7.	Testverfahren Game Creation Workflow	25
3.8.	Testverfahren Textureless Workflow	25
4.	Tests.....	26
4.1.	Tests Polygonperformance	26
4.2.	Tests Modell 1	26
4.3.	Tests Modell 2.....	27
4.4.	Tests Modell 3 Game	29
5.	Auswertung.....	31
5.1.	Polyperformance	31
5.2.	Modell 1 Game Asset Creation Workflow.....	31
5.3.	Modell 2 Game Asset Creation Workflow.....	31
5.4.	Modell 3 Game Asset Creation Workflow.....	32
5.5.	Zusammenfassung Performance Game Asset Creation Workflow	32
5.6.	Erscheinungsbild Game Asset Creation Workflow	33
5.7.	Mögliche Verbesserungen des Game Asset Creation Workflow für HoloLens Anwendungen.....	34
5.8.	Modell 1 Textureless Workflow	34
5.9.	Modell 1 Textureless Workflow	35
5.10.	Modell 1 Textureless Workflow	35
5.11.	Zusammenfassung Performance Textureless Workflow.....	35
5.12.	Erscheinungsbild Textureless Workflow	35
5.13.	Mögliche Verbesserungen des Textureless Workflow für HoloLens Anwendungen.....	36
5.14.	Vergleich der Performance	37
5.15.	Vergleich der Erscheinungsbilder	37
5.16.	Auswertung in Betracht auf die Füllrate	38
5.17.	Aufgetretene Probleme bei der Erstellung der 3D Modelle	38
5.17.1.	Probleme Textureless Workflow	38

5.17.2. Probleme Game Creation Workflow	39
6. Fazit.....	41
7. Ausblick	43
Literaturverzeichnis	X
Anhang.....	XIII
Selbstständigkeitserklärung	XV

Abbildungsverzeichnis

Abbildung 1: Eine Microsoft HoloLens	2
Abbildung 2: Beispiel einer Albedo Map.....	7
Abbildung 3: Beispiel einer Normal Map	8
Abbildung 4: Beispiel einer Ambient Occlusion Map	9
Abbildung 5: Beispiel einer Metallic Map.....	9
Abbildung 6: Beispiel einer Smoothness Map.....	10
Abbildung 7: Ein zu großer Winkel der Flächen untereinander sorgt für schwarze Stellen beim Aufteilen der Vertex Normalen	12
Abbildung 8: Bei ausreichend kleinem Winkel treten keine schwarzen Stellen auf	13
Abbildung 9: Modell 1 Game Asset Creation Workflow	16
Abbildung 10: Modell 1 Game Asset Creation Workflow Wireframe	16
Abbildung 11: Modell 1 Textureless Workflow.....	17
Abbildung 12: Modell 1 Textureless Workflow Wireframe	17
Abbildung 13: Modell 2 Game Asset Creation Workflow	17
Abbildung 14: Modell 2 Game Asset Creation Workflow Wireframe	17
Abbildung 15: Modell 2 Textureless Workflow.....	18
Abbildung 16: Modell 2 Textureless Workflow Wireframe	18
Abbildung 17: Modell 3 Game Asset Creation Workflow	18
Abbildung 18: : Modell 3 Game Asset Creation Workflow Wireframe	18
Abbildung 19: Modell 2 Textureless Workflow.....	19
Abbildung 20: Modell 3 Textureless Workflow Wireframe	19
Abbildung 21: Erscheinungsbild von Modell 1 mit dem Game Asset Creation Workflow	33
Abbildung 22: Erscheinungsbild von Modell 1 mit dem Game Asset Creation Workflow	34
Abbildung 23: Erscheinungsbild von Modell 1 mit dem Textureless Workflow	36
Abbildung 24: Erscheinungsbild von Modell 1 mit dem Textureless Workflow	36
Abbildung 25: Concept Art Modell 1,.....	XIV
Abbildung 26: Concept Art Modell 2.....	XIV
Abbildung 27: Concept Art Modell 3.....	XIV

Tabellenverzeichnis

Tabelle 1: Modell 1 Technische Informationen und Bilder.....	17
Tabelle 2: Modell 2 Technische Informationen und Bilder.....	18
Tabelle 3: Modell 3 Technische Informationen und Bilder.....	19
Tabelle 4: Tabellenvorlage für Tests der Polygonperformance	25
Tabelle 5: Tabellenvorlage für Tests mit dem Game Asset Creation Workflow ..	25
Tabelle 6: Tabellenvorlage für Tests mit dem Textureless Workflow	25
Tabelle 7: Profilertest Polygonperformance	26
Tabelle 8: Frameratetest Polygonperformance	26
Tabelle 9: Modell 1 Profilertest Game Asset Creation Workflow	26
Tabelle 10: Modell 1 Frameratetest Game Asset Creation Workflow	27
Tabelle 11: Modell 1 Profilertest Textureless Workflow.....	27
Tabelle 12: Modell 1 Frameratetest Textureless Workflow	27
Tabelle 13: Modell 2 Profilertest Game Asset Creation Workflow	28
Tabelle 14: Modell 2 Frameratetest Game Asset Creation Workflow	28
Tabelle 15: Modell 2 Profilertest Textureless Workflow.....	28
Tabelle 16: Modell 2 Frameratetest Textureless Workflow	28
Tabelle 17: Modell 3 Profilertest Game Asset Creation Workflow	29
Tabelle 18: Modell 3 Frameratetest Game Asset Creation Workflow	29
Tabelle 19: Modell 3 Profilertest Textureless Workflow.....	29
Tabelle 20: Modell 3 Frameratetest Textureless Workflow	30

Abkürzungsverzeichnis

AR	Augmented Reality
Bzw	Beziehungsweise
Ca	Circa
CPU	Central Processing Unity
Fps	Frames Per Second
GPU	Graphics Processing Unit
HPU	Holographic Processing Unit
M	Meter
Ms	Millisekunden
Retopo	Retopology
PBR	Physical Based Rendering

Glossar

Augmented Reality	Englisch für erweiterte Realität
Baking	Beschreibt das Übertragen von Informationen von einem 3D Modell in eine Bilddatei
Basemesh	Ein Mesh was für die Erstellung von Highpolymodellen dient oder als Grundlage für Texture Baking dient.
Content.	Englisch Für Inhalte für ein elektronisches Gerät oder eine Anwendung
Close Fit	Einstellungsoption für die Shadowprojektion. Liefert eine hohe Schattenauflösung, Schatten können aber etwas schwimmen bei Bewegung.
Device	Beschreibt ein elektronisches Gerät
Drawcall	Ein Drawcall ist eine Funktion welche Informationen an den Grafikchip sendet. (Michaud, 2015)
Edge	Ist die Kante eines Polygons, bzw. die Verbindung zwischen zwei Vertices
Edgeflow	Der Edgeflow beschreibt die Position der Edges und deren Verhältnis zueinander.
Face	Beschreibt die Fläche zwischen miteinander verbundenen Edges.
Frames Per Second	Beschreibt die von der Grafikeinheit berechneten Bilder pro Sekunde
Füllrate	Die Füllrate gibt an, mit wie viel Inhalt der Display der HoloLens gefüllt ist.
Grayscale	Englisch für Graustufen, ein Farbraum der Computergrafik, welcher nur Werte von null (schwarz) bis eins (weiss) ermöglicht.

Glossar

Hard Surface	Flache, harte Oberflächen, ohne organische Strukturen
Highpoly	Beschreibt eine Detailstufe eines 3D Modells mit möglichst viel Detail.
HoloLens	Ein von Microsoft entwickeltes Gerät zur Darstellung von 3D Inhalten im Raum
Lowpoly	Beschreibt die kleinstmögliche Polygonanzahl zur Darstellung eines bestimmten Detailgrads.
Polygon	Bestandteil eines 3D Modells. Beschreibt eine Fläche zwischen mindestens drei Vertices.
Map	Beschreibt den vom 3D Modell eingenommen Bereich in einem UV System
Material	Verbindung von Shader und Texturen zur Oberflächendarstellung von 3D Modellen
Mesh	Englisch Netz, ist das Netz eines 3D Modells
Mesh-Reduce	Reduzierung der Polygonanzahl eines Modells durch Algorithmen
Quad	Vierseitiges Polygon
Retopology	Retopology beschreibt ein erneutes Erstellen eines 3D Modells anhand des bereits bestehenden Modells, meist um Polygone zu sparen und so für Echtzeitberechnung nutzbar zu machen.
Sculpting	Sculpting ist eine Modelliertechnik, bei der ein Polygonnetz sich ähnlich wie Modelliermasse verhält und mit verschiedenen Tools bearbeitet werden kann.
Shading	Bezeichnet die Oberflächensimulation von 3D Objekten
Shadow Projection	Einstellungsmöglichkeit für Directional Light Schatten in den Qualitätseinstellungen von Unity

Glossar

Subdivision Modelling	Ist eine Modellieretechnik, bei der mit dem Einfügen neuer Unterteilungen von Flächen zusätzliches Detail mit Extrudieren erstellt wird.
Texel	Ein Texel ist die kleinste Einheit aus einer Textur
Textur	Eine Textur ist ein zweidimensionales Bild, welches zur Beschreibung von Oberflächen von 3D Modellen dient.
Tri	Dreieitiges Polygon
Unlit	Englisch für unbeleuchtet, beschreibt einen Shader, welcher das Modell unbeleuchtet lässt
UV-Map	Eine UV Map ist das Netz eines 3D Modells in zweidimensionaler Form.
UV-Shell	Eine UV Shell sind zusammenhängende Flächen in der UV Map.
Vertex	Ist ein Punkt in einem 3Dimensionalen Kartesischen Koordinatensystem und ein Bestandteil eines Polygons
Vertex Color	Zusätzliche in den Informationen eines Vertex gespeicherte Farbinformation
Virtual Reality	Englisch für Virtuelle Realität, durch das Tragen einer speziellen Brille erzeugt Immersion
Workflow	Englisch Arbeitsablauf, Beschreibt Schrittweise einen Arbeitsvorgang

1. Grundlegendes

Im Frühjahr 2017 bot sich mir die Gelegenheit, ein Praktikum bei VR Bits in Leipzig zu absolvieren. Während des Praktikums ergab sich für mich die Möglichkeit, eine Microsoft HoloLens ausgiebig zu testen und für diese zu entwickeln. In dieser Zeit entstand meinerseits eine gewisse Begeisterung für die Hologrammtechnik der HoloLens.

Einerseits bietet die HoloLens völlig neue Möglichkeiten der Visualisierung von 3D Modellen, andererseits zeichneten sich schon im selben Augenblick verschiedene Limitierungen ab. Microsoft ist, was die Leistungsfähigkeit der HoloLens angeht jedoch sehr verschwiegen. Durch das neue und recht teure Gerät sind Erfahrungswerte durch Benutzer rar.

1.1. HoloLens – eine Technik für die Zukunft

Heutzutage sind die meisten Augmented Reality Anwendungen auf zweidimensionalen Content beziehungsweise einen zweidimensionalen Bildschirm beschränkt. Mit dem Release der HoloLens im März 2016 bot Microsoft nun die Möglichkeit, die echte Welt mit Hologrammen zu erweitern. Da in absehbarer Zeit niemand mehr nur große und schwere Virtual Reality Brillen aufsetzen möchte und auch das Anwendungsfeld von Augmented Reality Anwendungen viel breiter gefächert ist, werden sich diese immer weiter verbreiten. Die HoloLens erstellt mithilfe von Sensoren zweidimensionale Bilder die den Eindruck erwecken, dass sich dreidimensionale Objekte im Raum befinden. Mit dieser Technik können in technischen- und medialen Bereichen neue Möglichkeiten erschlossen werden. Durch eine Weiterentwicklung der HoloLens und dem Erscheinen von neuen Geräten wird die neue Hologramm Technik in den nächsten Jahren Einzug in unser alltägliches Leben finden.



Abbildung 1: Eine Microsoft HoloLens

1.2. Abgrenzung der Themenstellung

Die Arbeit soll einen Aufschluss über die mögliche Verwendung unterschiedlicher Workflows für mobile 3D Augmented Reality Geräte anhand der Microsoft HoloLens geben. Die HoloLens steht hierbei symbolisch für alle mobilen Endgeräte. Mobile Endgeräte besitzen verschiedenen Limitierungen. Sowohl die Leistung der verbauten Hardware, als auch der Stromverbrauch und die Handhabbarkeit müssen aufgrund der maximal verwendbaren Größe angepasst werden. Bei auf dem Kopf getragenen Geräten ist das Gewicht eine weitere Limitierung. Ein sprunghafter Anstieg der Leistungsfähigkeit ist in den nächsten Jahren nicht zu erwarten, deshalb ist eine Anpassung der Workflows erforderlich.

Der Workflow welcher meist bei der Erstellung von Echtzeitgrafiken, wie zum Beispiel Videospiele zum Einsatz kommt, erwies sich bei vorherigen Arbeiten nicht immer als performant. Nun stellt sich die Frage, ob die HoloLens genügend Leistung besitzt um Modelle, welche mit diesem Workflow erstellt wurden, darstellen kann und ob der performantere, texturlose Workflow, eine Alternative bildet. In dieser Arbeit wird der Textureless Workflow als Alternative zum Game Asset Creation Workflow getestet und Performance und Erscheinungsbilder miteinander verglichen. Das Hauptaugenmerk liegt dabei auf der Erstellung der nötigen Assets, also 3D Modellen und Texturen sowie deren Vergleich. Das eigenständige Schreiben von unterschiedlichen Shadern würde dabei den Rahmen dieser Arbeit sprengen. Die genauen Bezeichnungen der später verwendeten Shader, finden sie im Anhang unter Anhang 1. Die Arbeit begrenzt sich damit auf die wesentlichen gestalterischen Punkte. Eine tiefgehende technische Optimierung bleibt hierbei offen und sollte Bestandteil einer anderen Arbeit sein. In der Arbeit werden somit bereits bestehende Shader verwendet,

Grundlegendes

welche bereits alle notwendigen Eigenschaften besitzen um die jeweiligen Workflows darzustellen. Selbiges gilt für das Testen von Texturen in unterschiedlichen Größen.

1.3. Aufbau der Arbeit

Zu Beginn der Arbeit werden sowohl der Game Asset Creation Workflow, als auch der Textureless Workflow beschrieben. Im Anschluss daran, werden wichtige Begrifflichkeiten und Vorgänge noch einmal detailliert beschrieben. Im zweiten Teil der Arbeit, geht es darum, ein Testverfahren zu entwickeln, welches beide Workflows miteinander vergleichbar macht und mögliche Stärken oder Schwächen offen legt. Bei der Auswahl der Modelle gab es verschiedene Kriterien. So werden Modelle mit einer verschiedenen Anzahl an Objekten, Materialien und Texturen getestet, um möglichst weitreichende Aussagen über die Performance unterschiedlicher Kombinationen treffen zu können. Im dritten Abschnitt wird die Erstellung der 3D Modelle behandelt. Auftretende Probleme und Hindernisse bei er Erstellung befinden sich in der Auswertung. Als dritter Punkt werden die Tests geführt. Hier werden die ermittelten Werte in Tabellen dargestellt und bieten einen genauen Überblick. Die Auswertung der ermittelten Werte erfolgt in Abschnitt vier. Bei der Auswertung werden die Punkte Erscheinungsbild und Performance beider Workflows analysiert und gegenübergestellt.

1.4. Ziel der Arbeit

Ziel dieser Arbeit ist es, Vor- und Nachteile von zwei verschiedenen Workflows für mobile Augmented Reality Geräte zu ermitteln. Dies geschieht am Beispiel der Microsoft HoloLens. Hierfür werden verschiedene 3D Modelle aus zwei Workflows getestet. Die Arbeit soll Aufschluss darüber geben, wo welcher Workflow die bessere Darstellungsqualität in Sachen Performance und Optik besitzt. Zum Schluss sollen anhand der Vor- und Nachteile der ermittelten Testergebnisse, eine Aussage darüber getroffen werden können, welcher Workflow der besser für mobile AR Endgeräte ist. Die Arbeit soll zusätzlich eine Hilfestellung bieten, den richtigen Workflow für die zu entwickelnde Anwendung zu wählen.

1.5. Technische Limitierungen der Microsoft HoloLens

Völlig neu ist das System der Datenbrille nicht. Bereits 2012 veröffentlichte Google mit der Google Glass eine Datenbrille, welche Informationen in die echte Welt einblendete. Nach massiver Kritik von Datenschützer zog Google jedoch das Produkt 2015 vom öffentliche Markt zurück und bietet zurzeit nur noch Business Lösungen mit der Glass an. (Lewicki, 2016)

Die HoloLens ist jedoch das erste Gerät, welches die reale Welt mit einem virtuellen 3D Raum verbindet, ohne dabei die reale Welt erneut über einen Display zu projizieren. Durch die Einblendung von dreidimensionalen Inhalten direkt in das Sichtfeld des Nutzers wird das Anwendungsspektrum gegenüber zweidimensionalen Augmented Reality Brillen noch einmal deutlich erweitert und die Hände bleiben beim Betrachten der dreidimensionalen Objekte, anders als bei Smartphone AR Anwendungen frei.

Dieser Arbeit geht ein dreimonatiges Praktikum voraus, bei welchem bereits erste Erfahrungswerte mit der HoloLens gesammelt werden konnten und bereits eine gewisse Einarbeitung in die Thematik stattfand. Dabei fielen bereits verschiedenen Besonderheiten auf, welche kurz erläutert werden sollen und in der folgenden Arbeit als gegeben angesehen werden.

Das Kombinieren von Objekten besitzt einen Einfluss auf die Performance. Bei einem Modell welches während des Praktikums erstellt wurden war und circa 50.000 Polygone besaß, wurde zu Beginn ohne das Zusammenfügen von Einzelobjekten dargestellt. Nachdem es zu Performanceproblemen kam, wurden mehrere hundert Einzelobjekte zusammengefügt. Nach dem Zusammenfügen gab es keine Performanceprobleme mehr. Daraus folgt, dass Objekte möglichst Zusammengefügt werden sollten.

Dies ist nur dann ein Vorteil, wenn die zusammengefügten Objekte auch das selbe Material besitzen. Ein Objekt mit zwei Materialien muss erst in seine Einzelteile mit den unterschiedlichen Materialien zerlegt werden, um gerendert zu werden. Dies wiederum sorgt für eine schlechtere Performance als bei nicht zusammengefügten Objekten. (De Jong, o.D.)

Ein weiterer Punkt ist die Verwendung von nicht geeigneten Shadern. In der HoloLens steckt nahezu dieselbe Technik wie in Smartphones. Diese sind im

Grundlegendes

Vergleich zu PC's, welche oft für 3D Anwendungen zum Einsatz kommen, nicht besonders leistungsfähig. Für mobile Endgeräte gibt es deshalb bereits eigens geschriebene Shader, welche meist weniger Optionen bieten, jedoch auch deutlich performanter sind. Dieses Bild bot sich bereits bei vorangegangenen Arbeiten. Dabei wurde festgestellt, dass der Unity Standard Shader die Framerate sofort merklich absenkt. Für die Arbeit werden jeweils Shader verwendet, welche für den jeweiligen Workflow ausgelegt sind. Dabei handelt es sich beim Textureless Workflow um einen Unlit Shader ohne Lightmap Funktion, welcher auf die Farbwerte der Vertices zurückgreift. Für den Game Asset Creation Workflow kommt eine optimierte Variante des Unity Standard Shaders aus dem Holo Toolkit zum Einsatz. Die Shader werden dabei zum Bestandteil des Workflows gezählt.

Während des Testens für vorangegangene Projekte zeigte sich, dass das Bild welches die HoloLens erstellt, nicht immer den Eindruck macht, als ob es tatsächlich physisch vorhanden wäre. Die besten Ergebnisse für nicht transparente Bilder ergab sich, bei leicht abgedunkelter, diffuser Beleuchtung.

Die HoloLens besitzt eine Vielzahl von Einschränkungen, welche das Erscheinungsbild der dargestellten Inhalte massiv beeinflussen. Als erstes ist die Darstellung von schwarzen bzw. dunklen Farbtönen aufgrund der additiven Farbmischung des verbauten LCD Displays nicht möglich. Schwarze und dunkle Farben sollten daher nicht verwendet werden, da diese für Transparenz im Modell sorgen.

Zusätzlich ist anzumerken, dass die HoloLens aus Berechnungsgründen eine Frameratelimitierung von 60Fps besitzt. Diese 60Fps sind notwendig um eine stabile Darstellung der Inhalte zu gewährleisten. Bei geringeren Frameraten kann es passieren, dass sich Objekte während der Laufzeit verschieben, oder auch verschwimmen, d.h. sie bewegen sich um den eigentlichen Mittelpunkt des Objektes herum. Eine Framerate von 60Fps ist daher ein wichtiges Ziel bei der Erstellung von Anwendungen auf der HoloLens.

Die HoloLens besitzt ungefähr die Leistungsfähigkeit eines Smartphones, da sie mit einem Intel Atom x5 Prozessor, 2GB Arbeitsspeicher und 114 MB Grafikspeicher ausgestattet ist (Rubino, 2016).

Des Weiteren ist zu beachten, dass die HoloLens kein tatsächliches 3D Modell projiziert. Es handelt sich hierbei nur um ein zweidimensionales Bild, welches auf Grundlage mehrere Sensoren berechnet wurde und so einen dreidimensionalen Eindruck vermittelt.

1.6. Verwendete Workflows

1.6.1. Game Asset Creation Workflow

Der konventionelle Game Asset Creation Workflow kommt aus der Spieleentwicklung. Das Ziel des Game Asset Creation Workflows ist es, die bestmögliche Optik, so ressourcenschonend wie möglich darzustellen. Die Anforderungen an ihn sind dabei anpassbar. Der größte Nutzen im Game Asset Creation Workflow liegt darin, dass er schnell für die jeweilige Zielplattform angepasst werden kann. Das Prinzip ist dabei immer gleich.

Ein Objekt mit wenigen Polygonen soll mit Oberflächeninformationen eines Objektes mit vielen Polygonen erweitert werden, um die Berechnungszeit zu einem späteren Zeitpunkt so gering wie möglich zu halten. (Nguyen H., nach Karlsson, 2013)

Dabei kann zuerst ein Modell mit wenigen Polygonen erstellt werden und dieses dann erweitert werden, oder es wird zuerst ein Modell mit vielen Polygonen erstellt, welches wieder vereinfacht wird. Das hinzufügen dieses Details geschieht dann in Form von Texturen, so weit, bis der benötigte Detailgrad erreicht ist. Weitere Oberflächeninformationen können zusätzlich über das Material hinzugefügt werden. Das Lowpolyobjekt liefert dabei nur die Silhouette. Folgende Texturen wurden im weiteren Verlauf der Arbeit verwendet.

Grundlagen von Texturen

Eine Textur ist ein Array von Farbwerten. Diese Farbwerte werden als Texels bezeichnet („Programming with OpenGL: Advanced Rendering“, o.D.). Eine Textur kann eindimensional, zweidimensional oder auch dreidimensional sein, wobei es sich bei den nachfolgenden Texturen nur um 2D Texturen handelt. Texturen werden üblicherweise dafür eingesetzt, um schwierig darzustellende Oberflächen mithilfe von einfacher Geometrie und Farbwerten darzustellen („Programming with OpenGL: Advanced Rendering“, o.D.).

Grundlegendes

Texturen bilden im klassischen Game Creation Workflow eine entscheidende Rolle, da über sie ein Großteil des Details auf das Modell übertragen wird. Die folgenden Tests beziehen sich auf vier bzw. fünf Texturen, welche jeweils einen sehr großen Einfluss auf die Optik besitzen. Diese sollen den kompletten Gegensatz zum Textureless Workflow bilden.

Albedo Map

Die Albedo Map ist für die Farbinformation zuständig und unterscheidet sich in einigen Punkten von der immer noch sehr populären Diffuse Map. Die Diffuse Map wird beim klassischen Texturierungsworkflow eingesetzt. Der Unterschied zur Albedo Map liegt dabei in der Beleuchtung und in den Schattenwürfen, welche in der Diffuse Map vorhanden sind. Eine Albedo Map beinhaltet nur die reine Farbinformation der Oberfläche, ohne jegliche Einwirkung anderer Faktoren wie zum Beispiel Licht. Eine Albedo Map wird im aktuellen Physically Based Rendering Workflow eingesetzt. Bei diesem werden zusätzliche Informationen wie detaillierte Schattenwürfe, per zusätzlicher Textur eingefügt um den Realismus zu erhöhen. (CGFort PTY LTD, 2017, Reallusion Inc., o.D.)

Fotos welche sehr gern als Grundlage für Texturen genommen werden, sind durch vorhandenes und notwendiges Licht beim Fotografieren prinzipiell Diffuse Maps. Mithilfe von Photoshop oder anderen Bildbearbeitungsprogrammen können jedoch Schatten sehr gut entfernt oder minimiert werden und somit aus Fotos Albedo Texturen erstellt werden.



Abbildung 2: Beispiel einer Albedo Map

Normal Map

„Normal Maps sind Texturen, welche detailliert Einkerbungen und Erhöhungen mithilfe eines bitmap Bildes darstellen können. [...] Auch, wenn Normal Maps 2D Bilder sind, sind sie in der Lage 3D Informationen zu speichern. Die RGB Farbkanäle (rot, grün, blau) in einer Normal Map, entsprechen den X, Y und Z

Grundlegendes

Koordinaten der Oberflächennormalen. Während Normal Maps nicht in der Lage sind tiefe Einkerbungen und starke Verformungen darzustellen, so sind sie jedoch in der Lage kleine Dellen und Erhebungen zu imitieren. Imitieren deshalb, weil eine Normal Map keine Geometrie hinzufügt. Vielmehr fügt jeder einzelne Pixel weitere Surface Normalen hinzu welche bei den Lichtberechnungen mit einbezogen werden, was wiederum in Highlights und Schatten resultiert, welche vorhanden wären, wenn zusätzliche Geometrie da wäre, auch ohne Geometrie.“ (Pettit, 2015a)

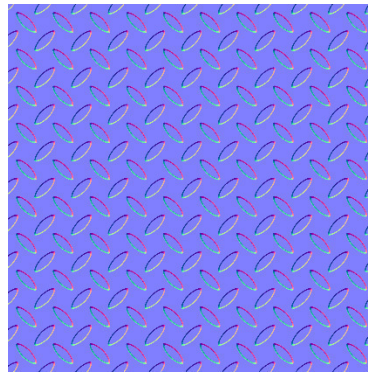


Abbildung 3: Beispiel einer Normal Map

Ambient Occlusion Map

„Die Occlusion Map wird benutzt, um zu definieren, welche Stellen des Modells stärker von indirektem Lichteinfall betroffen sind als andere. Indirektes Licht kommt von Ambient Light und Reflektionen. So werden steile Teile des Modells, wie zum Beispiel Risse oder Falten, in der realen Welt nicht viel indirektes Licht abbekommen. Occlusion Texturen werden in der Regel vorher durch 3D Anwendungen direkt aus dem 3D Modell berechnet.“ (Unity Technologies, o.D.)

Jedoch gibt es auch die Möglichkeit Ambient Occlusion zur Laufzeit zu berechnen. Diese Berechnungsmethoden, wie zum Beispiel Screen Space Ambient Occlusion, sind jedoch sehr rechenintensiv und nicht sehr genau. Eine optisch bessere und performantere Lösung sind somit die schwieriger zu erstellenden Ambient Occlusion Maps. (NVIDIA Corporation, o.D.)

„Eine Occlusion Map ist (...) eine Greyscale Textur, mit Weiss, welche stellen mit einer hohen indirekten Beleuchtung beschreibt und Schwarz, welche Stellen ohne indirekte Beleuchtung indiziert.“ (Unity Technologies, o.D.)

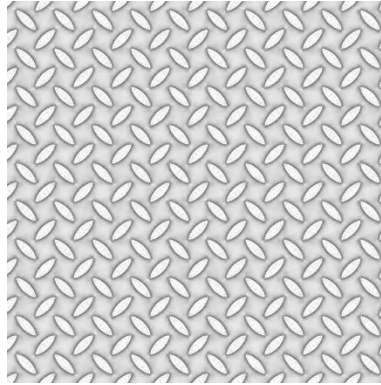


Abbildung 4: Beispiel einer Ambient Occlusion Map

Metallic Map

„PBR Materialien welche Metallic Maps nutzen werden in (...) zwei Kategorien unterteilt: metallische und nicht-metallische. (...) Der Grund für diese Unterteilung ist, dass PBR Materialien beschreiben müssen, wie Licht von einer Oberfläche reflektiert wird. Reflektionen auf metallischen Oberflächen neigen dazu farbig zu sein, wohingegen Reflektionen auf nicht-metallischen Flächen eher neutral sind. Am besten Stellt man sich die Reflektionen auf einer verchromten Stoßstange und einem glänzenden Stück Plastik vor [...]“. (Pettit, 2015)

„Eine Metallic Map beschreibt nun einfach welcher Teil des Materials metallisch und welcher nicht-metallisch ist. Weiße Pixel sind metallisch und Schwarze Pixel sind nicht-metallisch. Sie sollten prinzipiell keine Werte dazwischen besitzen. Eine Ausnahme für leichte Grautöne stellt die Überlagerung von Metallen (Aluminum) durch nicht-Metalle (z.B. Rost oder Dreck) dar“. (Pettit, 2015)

Bei einer Metallic Map handelt es sich zusammengefasst um eine Grayscale Textur. Weiße Stellen auf der Textur beschreiben metallische Stellen des Materials, wohingegen dunkle und schwarze Stellen nicht metallischen Ursprungs sind.

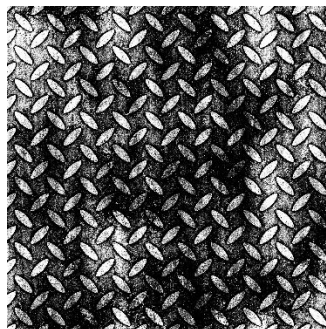


Abbildung 5: Beispiel einer Metallic Map

Smoothness Map

„Die Smoothness Map beschreibt wie glatt oder rau eine Oberfläche erscheint. Genau wie die Metallic Map, ist auch die Smoothness Map eine Grayscale Map. Weiße Pixel sind zu 100% reflektierend und damit sehr glänzend, wohingegen schwarze Pixel nicht reflektieren“. (Pettit, 2015)

Die Smoothness Map imitiert dabei die unterschiedlichen Reflektionswinkel einer Oberfläche durch kleine Unebenheiten der Oberfläche. Desto heller eine Stelle in der Smoothness Map ist, desto stärker ist der Glanzpunkt. Die Menge des reflektierten Lichts, ist jedoch von der Smoothness Map unabhängig und bleibt immer gleich.

Die Smoothness Map muss sich bei der Verwendung des Unity Standard Shaders im Alphakanal der Metallic Map befinden.

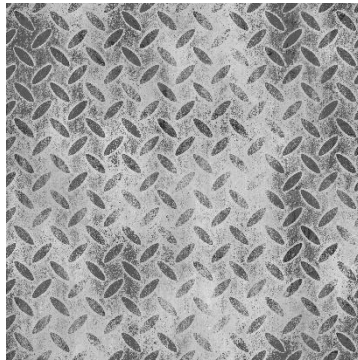


Abbildung 6: Beispiel einer Smoothness Map

Durch die erforderliche Echtzeitberechnung von Inhalten in Spielen, wird sehr auf die Anzahl der Polygone geachtet und der Großteil der Details über Texturen, bzw. Materialien erreicht. Modelle welche mit dem Game Creation Workflow erstellt werden, besitzen neben dem eigentlichen 3D Mesh mit Vertex Informationen also noch Texturen. Der Unterschied zu 3D Modellen für Offline Renderings liegt dabei in der Optimierung der Modelle und Texturen für die jeweilige Zielplattform.

Der folgende Workflow ist nur eine Möglichkeit das gewünschte Ergebnis zu erzielen. Jeder Workflow bietet dabei nur eine Richtlinie zur Erstellung von Assets. Aufgrund äußerer Gegebenheiten, kann dieser jedoch zu jedem Zeitpunkt angepasst werden, um Zeit zu sparen oder um Detail zu gewinnen. Auch die Verwendung anderer Software ist möglich. Der hier aufgeführte Workflow bezieht sich dabei auf eine abgeänderte Version des Tutorials von Aidy

Grundlegendes

Burrows (2015) auf „cgmasters.net“. Grundsätzlich läuft der Workflow wie folgt ab:

Die Erstellung eines Highpolymodells bildet die Grundlage für den Workflow. In diesem Schritt wird so viel Detail wie möglich modelliert um dieses dann auf eine Vereinfachte Version zu Übertragen. Die Erstellung kann dabei mit verschiedenen Programmen stattfinden.

Auf Grundlage des Highpolymodells kann daraufhin eine vereinfachte Version des Modells erstellt werden, welche die selbe Silhouette besitzt und in der Lage ist, sämtliches Detail des Highpolymodells aufzunehmen. Die Erstellung kann dabei mit verschiedenen Techniken geschehen.

Bevor die Oberflächeninformationen des Highpolyobjektes mittels eines Bakes auf das Lowpolyobjekt übertragen wird, muss dieses ein UV Mapping erhalten. Dies ist notwendig, da bei einer Überlappung von UV Koordinaten Oberflächeninformationen an einer falschen Stelle zu sehen wären.

Wurde ein UV Mapping erstellt, kann nun mit spezieller Software die Oberflächeninformation des Highpolyobjektes auf das Lowpolyobjekt transferiert werden. Jedoch gibt es weitere Möglichkeiten Texturen zu erstellen. So können mit Bildbearbeitungssoftware Texturen erstellt werden, was bei Smoothness und Metallic Map meist der Fall ist. Ebenso ist es möglich mit spezieller Software Oberflächeninformationen zusätzlich in bestehende Texturen einzufügen, oder gar von Grund auf neu zu erstellen.

Nachdem alle benötigten Texturen erstellt wurden, kann das Modell mit sämtlichen Texturen in eine Gameengine geladen werden und die Werte der Texturen im Shader angepasst werden.

1.6.2. Textureless Workflow

Der Textureless Workflow zielt auf die Erstellung von 3D Modellen ohne Texturen ab. Der Grundgedanke dahinter ist, dass mit dem Entfernen von Texturen die Anzahl der Drawcalls sinkt, was damit direkt die Performance verbessert. Sämtliche Farb- und Normaleninformationen werden dabei in den Vertices gespeichert. Dies sorgt dafür, dass am Ende des Workflows keine Texturen vorhanden sind. Er ähnelt somit der Erstellung von CAD Modellen, welche meist auf eine ähnliche Art der Texturierung setzen.

Grundlegendes

Der folgende Workflow bezieht sich auf einen Blogeintrag von Tomas Sala. Dieser Entwickelte für das Spiel Oberon's Court einen Textureless Workflow (Sala, 2014).

Im ersten Schritt wird das 3D Modell erstellt. Dieses erste 3D Modell ist das einzige Modell, was für die Erstellung des jeweiligen Assets benötigt wird. Bei der Erstellung von texturlosen 3D Modellen muss bedacht werden, dass keine separate Normal Map Detail hinzufügen kann. Deshalb muss von Anfang an auf 3D Detail geachtet werden. Um nun die Anzahl der Polygone nicht zu sehr in die Höhe zu treiben, muss entschieden werden, an welchen Stellen Detail wichtig ist und auf welches man verzichten kann. Zu Beginn sollte jedoch so viel Detail wie möglich erstellt werden. Sollte es zu Problemen kommen, kann dies auch später wieder entfernt werden.

Da auf Normal Maps verzichtet wird, wird eine andere Technik zur Definition der Silhouette eingesetzt. Das Aufteilen der Vertex Normalen sorgt für ein abrunden harter Kanten. Werden hingegen diese beiden Vertex Normalen zusammengeführt, kommt es zu einer harten Kante. Beim Abrunden von Kanten sollte jedoch beachtet werden, dass der Winkel nicht zu groß sein darf, da es sonst zu shading Problemen kommen kann. Diese machen sich durch schwarze Stellen bemerkbar.

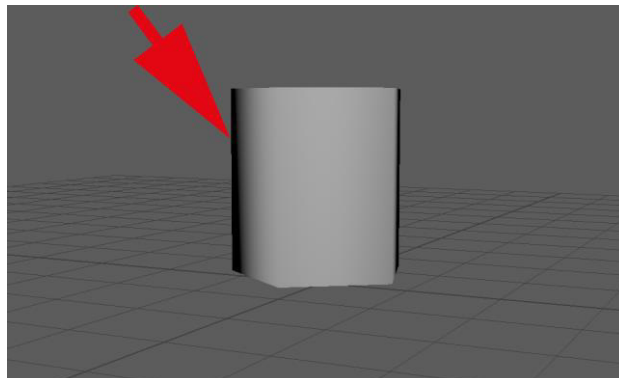


Abbildung 7: Ein zu großer Winkel der Flächen untereinander sorgt für schwarze Stellen beim Aufteilen der Vertex Normalen

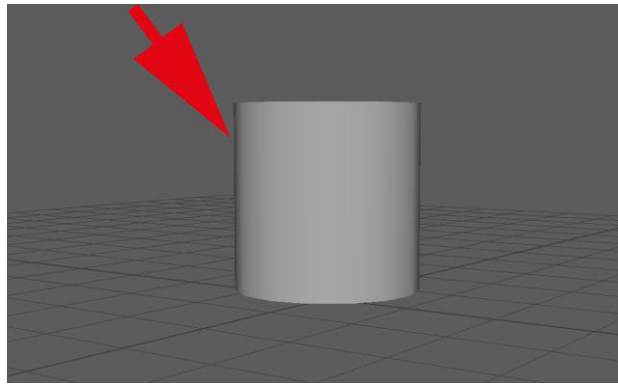


Abbildung 8: Bei ausreichend kleinem Winkel treten keine schwarzen Stellen auf

Farbinformationen werden mittels Vertex Color realisiert. Vertex Color sind dabei Farbinformationen welche zusätzlich in den Vertices gespeichert werden.

Da die Farb- und Oberflächeninformation nur über Vertex Color und Vertexnormalen definiert werden kann, ist der optische Detailgrad im Textureless Workflow mit der Dichte der Vertices verknüpft. Es kann also in eine Fläche nicht noch zusätzliche Farbinformation eingebracht werden. Das Vertex Painting erfolgt über die im 3D Programm verwendeten Hilfsmittel. Diese Hilfsmittel ermöglichen es, Farben mit verschiedenen Modi auf die Vertices und Flächen aufzubringen. Diese Farbinformationen entsprechen den reinen Farbinformationen der Oberfläche ohne Schattierungen.

In aktuellen 3D Programmen ist es möglich, Ambient Occlusion Informationen auf Vertices zu übertragen. Diese Informationen werden vorerst in einem zweiten Vertex Color Kanal gespeichert und anschließend mittels einer Operation zusammengefügt. So besitzt der einzige Vertex Color Kanal sowohl die reine Farbinformation als auch Ambient Occlusion Information.

Nachdem alle benötigten Informationen in den Vertices und die Vertexnormalen festgelegt wurden, kann das Modell als FBX Datei exportiert werden und in Unity3D importiert werden. Mit einem Vertexshader, welcher alle Informationen des Modells nutzt, kann das Modell dargestellt werden.

1.7. Technischer Hintergrund bei der Auswahl der Workflows

Beim Rendern von 3D Computergrafik müssen verschiedene Schritte abgearbeitet werden. Diese Aneinanderreihung von Schritten wird Rendering

Grundlegendes

Pipeline genannt. Einige Schritte dieser Pipeline sind jedoch nicht zwingend notwendig. (Khronos, 2017)

Der Game Asset Creation Workflow greift dabei fast immer auf alle möglichen Stufen zurück. Besonders der Fragment Shader und der Tessellation Shader werden beim Game Asset Creation Workflow oft stark verwendet um die gesamte Polygonanzahl so gering wie möglich zu halten. Der Fragment Shader berechnet anhand von Faktoren wie Geometrie, Licht und Texturen, für jeden einzelnen Pixel den entsprechenden Farbwert.

Der Textureless Workflow soll nun den Punkt Fragment Shader in der Rendering Pipeline so weit wie möglich entlasten. Die Polygonanzahl des Modells wird dafür erhöht um Detail zu erzeugen, jedoch soll mit dem Verzicht auf Texturen die Arbeit des Fragment Shaders verringert werden. Damit steigen jedoch die Berechnung der frühen Stufen der Rendering Pipeline, welche die gesamten Vertex Informationen verarbeiten. Das Verarbeiten von Vertex Informationen ist jedoch nicht so rechenaufwendig, was zu einem Performancegewinn führt.

2. Erarbeitung des praktischen Teils

Zur Darstellung dreidimensionaler Inhalte mit Augmented Reality Geräten, sind 3D Modelle erforderlich. Um diese dann mit der HoloLens darstellen zu können, muss zusätzlich mit Unity3D eine Anwendung nach bestimmten Regeln erstellt werden. In den folgenden Abschnitten erfolgt eine Auflistung der erstellten Modelle und eine Beschreibung des Untersuchungsdesigns, welches die Grundlage für die Entwicklung der Tests bildet.

2.1. Untersuchungsdesign

Zur Untersuchung der Frage, ob der Textureless Workflow eine Alternative zum Game Asset Creation Workflow bietet und wo das technische Limit der HoloLens liegt, sind eine Vielzahl von Faktoren zu beachten. Der Test wird hierbei auf die Parameter Textur, Material und Objektmenge eingegrenzt. Vortest sind nötig, um Faktoren wie Licht zu definieren.

Ein erster Vortest soll statisches und dynamische Licht miteinander vergleichen. Die performantere Variante soll in weiteren Tests als Standard verwendet werden.

Bereits am Anfang der Tests soll geklärt werden, wo das maximale Polygonlimit der HoloLens liegt, um aufzuzeigen welche Polygonanzahl nicht überschritten werden darf. Dies soll geschehen, damit die Polygonanzahl nicht die eigentlichen Vor- und Nachteile des Workflows verfälscht und die Framerate nicht bereits durch die reine Polygonanzahl abgesenkt wird.

In den Haupttests soll überprüft werden, ob der Textureless Workflow wie vermutet die bessere Performance der beiden Workflows besitzt und welchen Einfluss verschiedene Texturtypen, die Komplexität von Materialien, die Anzahl von Materialien und die Anzahl von Objekten auf die Performance haben. Hierfür werden Modelle mit einer unterschiedlichen Anzahl von Objekten, Materialien und Texturen erstellt und miteinander verglichen. Damit soll Aufschluss darüber gegeben werden, an welchen Stellen Probleme auftreten und wie diese optimiert werden können.

Zusätzlich besitzen die Modelle unterschiedliche Materialeigenschaften, um optisch einen Vergleich zwischen den beiden Workflows ziehen zu können. Aus diesem Grund wurden die hier aufgeführten Modelle ausgewählt.

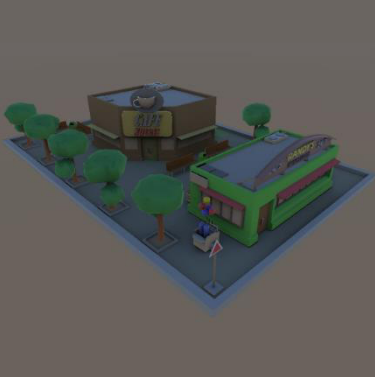
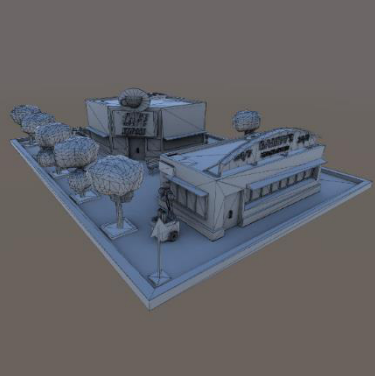
Erarbeitung des praktischen Teils

Bei den Tests zum Game Asset Creation Workflow ändert sich von Modell zu Modell die Anzahl der Polygone, die Anzahl der Objekte, die Anzahl der Materialien, die Anzahl der Texturen und die Füllrate, bedingt durch die unterschiedlichen Formen der Modelle. Der Shader des Materials bleibt hingegen gleich. Beim Textureless Workflow ändert sich die Füllrate und die Anzahl der Polygone.

Als zu testende Werte, werden primär Berechnungszeiten aus dem Unity Profiler verwendet, als auch angezeigt Framerraten.

2.2. Erstellte Modelle

Nachfolgend sind die erstellten 3D Modelle mit einigen Eckdaten zu sehen. Die dafür als Vorlage verwendeten Concept Arts, finden sie im Anhang unter Anhang 3.

<p>Modell 1 Game Asset Creation Workflow</p>	<p>Anzahl Vertices:12574 Anzahl Polygone: 12463 Anzahl Tris: 21907 Anzahl der Objekte: 5 Anzahl Materialien: 5</p>	 <p>Abbildung 9: Modell 1 Game Asset Creation Workflow</p>  <p>Abbildung 10: Modell 1 Game Asset Creation Workflow Wireframe</p>
--	--	---

Erarbeitung des praktischen Teils


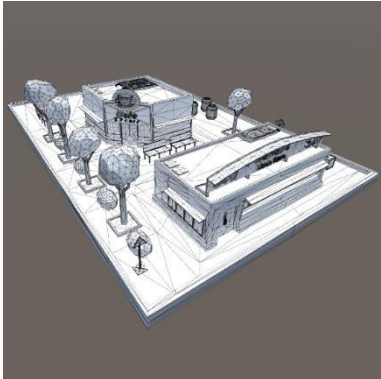


<p>Modell 1 Textureless Workflow</p>	<p>Anzahl Vertices: 23679 Anzahl Polygone: 23246 Anzahl Tris: 46492 Anzahl der Objekte: 1 Anzahl Materialien: 1</p>	 <p>Abbildung 11: Modell 1 Textureless Workflow</p>  <p>Abbildung 12: Modell 1 Textureless Workflow Wireframe</p>
--	---	---

Tabelle 1: Modell 1 Technische Informationen und Bilder

<p>Modell 2 Game Asset Creation Workflow</p>	<p>Anzahl Vertices: 642 Anzahl Polygone: 650 Anzahl Tris: 1196 Anzahl der Objekte: 2 Anzahl Materialien: 2</p>	 <p>Abbildung 13: Modell 2 Game Asset Creation Workflow</p>  <p>Abbildung 14: Modell 2 Game Asset Creation Workflow Wireframe</p>
--	--	---

Erarbeitung des praktischen Teils

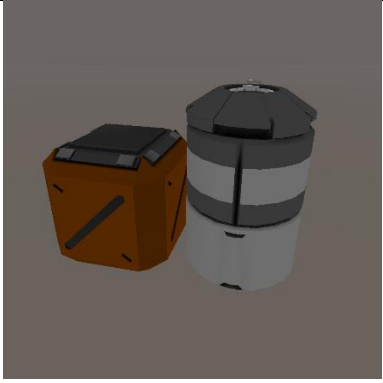



<p>Modell 2 Textureless Workflow</p>	<p>Anzahl Vertices: 1492 Anzahl Polygone: 1529 Anzahl Tris: 2834 Anzahl der Objekte: 1 Anzahl Materialien: 1</p>	 <p>Abbildung 15: Modell 2 Textureless Workflow</p>  <p>Abbildung 16: Modell 2 Textureless Workflow Wireframe</p>
--	--	---

Tabelle 2: Modell 2 Technische Informationen und Bilder

<p>Modell 3 Game Asset Creation Workflow</p>	<p>Anzahl Vertices: 3132 Anzahl Polygone: 2941 Anzahl Tris: 5599 Anzahl der Objekte: 1 Anzahl Materialien: 1</p>	 <p>Abbildung 17: Modell 3 Game Asset Creation Workflow</p>  <p>Abbildung 18: : Modell 3 Game Asset Creation Workflow Wireframe</p>
--	--	---

Erarbeitung des praktischen Teils



<p>Modell 3 Textureless Workflow</p>	<p>Anzahl Vertices: 3046 Anzahl Polygone: 2884 Anzahl Tris: 5577 Anzahl der Objekte: 1 Anzahl Materialien: 1</p>	 <p>Abbildung 19: Modell 2 Textureless Workflow</p>  <p>Abbildung 20: Modell 3 Textureless Workflow Wireframe</p>
--	--	---

Tabelle 3: Modell 3 Technische Informationen und Bilder

3. Entwicklung des Testverfahrens

In den folgenden Abschnitten wird beschrieben, wie das spezielle Testverfahren für die HoloLens hergeleitet wurde. Die Testverfahren sollen möglichst präzise Auskunft geben, an welchen Punkten verschiedenen Workflows, aufgrund der heute verwendeten Technik nicht praktikabel sind.

3.1. Licht im Testverfahren

Ein Vortest für die Art der Beleuchtung war notwendig, da ein Testen aller Modelle mit jeweils dynamischen und statischem Licht für diese Arbeit zu aufwendig gewesen wäre. Mit dem Vortest kann jedoch eine recht genaue Aussage getroffen werden, welche Beleuchtungsart eine bessere Performance liefert. Des Weiteren sind Schatten immer dunkler bzw. dunkel und schwarz. Beschattete Stellen werden somit immer mit einer gewissen Transparenz dargestellt. Schatten sind deshalb nicht so stark sichtbar wie auf einem normalen Bildschirm. Im folgenden kleineren Vortest, wird nur ein Modell genommen und dieses sowohl mit statischem Licht, als auch mit dynamischen Licht beleuchtet. Das Beleuchtungsmodell, welches geringeren Einfluss auf die Performance hat, wird dann in allen anderen Test als Standard verwendet. Dies gilt jedoch nur für Shader welche Licht unterstützen. Bei Unlit Shadern spielt direkte Beleuchtung keine Rolle, jedoch wäre es möglich Schatten zu erstellen. Die dabei entstehende Lightmap ist aber wiederum eine Textur und wird deshalb nicht in Betracht gezogen.

Statisches Licht oder auch Baked Lighting

„Unity berechnet die Beleuchtung der Baked Lights vor der Laufzeit und verzichtet komplett auf Lichtberechnungen während der Laufzeit. Alle Baked Lights werden zuvor berechnet“. (Unity Technologies, 2017)

Dynamisches Licht oder auch Realtime Lighting

„Unity berechnet und aktualisiert das Licht der Realtime Lichter jeden Frame während der Laufzeit. Realtime Lichter sind nicht vorberechnet“. (Unity Technologies, 2017)

Der Test

Für den folgenden Test werden drei Modelle mit jeweils ca. 10.000 Polygonen beleuchtet. Das verwendete Material besitzt einen Unity Mobile Shader Diffuse und keine Textur, sondern nur einen Farbwert.

Bereits bei der Projekterstellung nach Microsoft Vorgaben (Microsoft, o.D.) zeigte sich, dass die optimale Performance der HoloLens nicht mit dynamischer Beleuchtung und Schatten zu erreichen ist. In den Project Quality Settings wird empfohlen, die schnellste Einstellung zu wählen. Diese wird in Unity selbst als „Fastest“ bezeichnet. Diese Voreinstellungen unterstützen keinerlei Schatten in Echtzeit. Dies kann jedoch noch angepasst werden. So wurden harte Schatten aktiviert um diese in Echtzeit zu rendern. Des Weiteren wurde Close Fit bei der Shadow Projection aktiviert, um optisch ein ähnliches Ergebnis zu erreichen wie mit der vorberechneten Lightmap.

Die Entfernung zu den Objekten Betrag gleichbleibend zwei Meter und die Füllrate lag bei geschätzten 30-40 Prozent.

Beim Testen der statisch beleuchteten Szene ergaben sich konstante 60Fps, wohingegen bei der dynamisch beleuchteten Szene bereits Einbrüche in der Framerate zu verzeichnen waren. Somit wird deutlich, dass statisches Licht eine deutlich bessere Performance liefert als dynamisches Licht.

3.2. 3D Modelle im Testverfahren

„Ein 3D Model ist eine mathematische Repräsentation eines dreidimensionalen Objektes (egal ob echt oder Fiktion) in einer 3D Software Umgebung. [...] 3D Modelle können mit spezieller Software von jedem Winkel aus betrachtet, skaliert, rotiert und frei modifiziert werden“. (Slick, 2017)

Die Microsoft HoloLens verarbeitet zusammen mit Unity Polygonmodelle. „Polygon Modelle [...] sind die am weitesten verbreitete Form von 3D Modellen in Animation, Film und der Spieleindustrie“. (Slick, 2017)

Ein Teil der 3D Performance wird durch das Polygonmodell beschrieben, welches dann weitere Eigenschaften besitzt. Um ein möglichst genaues Bild der Leistungsfähigkeit abgeben zu können werden drei Modelle bzw. Szenen getestet.

3.3. Das Testszenario

Das nachfolgend beschriebene Testverfahren soll einen Abriss geben, wo die technischen Limitierung der Microsoft HoloLens im Bereich der 3D Performance liegen. Als Wert für die Performance gelten Primär die benötigten Berechnungszeiten der HoloLens. Als sekundärer Richtwert gilt die ausgegebene Framerate. Diese wird nur als sekundärer Wert verwendet, da die HoloLens die Framerate auf 30 und 60Fps begrenzt um eine stabile Darstellung zu gewährleisten. Die Ermittlung der Berechnungszeiten erfolgt über den Unity internen Profiler, welcher aufschlüsselt, wie lang welche Prozesse zur Berechnung benötigen.

3.3.1. Äußere Faktoren

Um gleichbleibende Bedingung für alle Tests zu gewährleisten, ist eine Vielzahl von Faktoren genau festzulegen. Diese sollen ausschließen, dass zum Beispiel Dateitypen und Größen sowie Programmversionen einen nicht berechenbaren Einfluss auf die Ergebnisse besitzen. Die folgenden Faktoren bilden dabei nur einen konkreten Anwendungsfall.

HoloToolkit

Von Microsoft wird ein nicht kommerzielles Toolkit empfohlen, welches eine große Anzahl von Scripts und Shadern für die HoloLens bereitstellt. Dieses Kit wird zurzeit weiterentwickelt (Stand Juli 2017). Die verwendete Version ist für Unity 2017.1.0f3 optimiert und wurde am 11.07.2017 veröffentlicht. HoloToolkit (HoloToolkit, o.D.) besitzt keine Versionsnummer und wird permanent mit kleineren Updates versehen.

Modellgröße in der HoloLens

Die in der HoloLens abgebildeten Modelle, müssen in einen 1mx1mx1m großen "Käfig" passen. Dies soll sicherstellen, dass die Füllrate während des Test desselben Modells nicht schwankt und dadurch Veränderungen in den zu messende Punkten auftreten.

Entfernung zum Model

Die Entfernung zum Modell soll Aufschluss darüber geben, ob und wenn ja die Füllrate des Displays Einfluss auf die Framerate der HoloLens besitzt. Die

Entwicklung des Testverfahrens

Abstände der Objekte zum Benutzer wird dabei über die Kamera und Objektposition in Unity festgelegt. Der Abstand beträgt 0,5m und 2m zum Objekt.

Software Version

Die verwendete Unity Version ist Unity3D 2017.1.0f3, MayaLT 2017 und Photoshop CC 2017.

Datei Typen

Für den Import und Export von 3D Modellen wird der von Autodesk entwickelte FBX Dateityp verwendet. Für Texturen werden PNG Dateien verwendet, welche beim Export nicht komprimiert werden. Die Verwendung von anderen Dateitypen sollte jedoch nichts ändern, da importierte Dateien von Unity in ein eigenes Format umgewandelt werden und daher nach dem Import der Dateityp keine Rolle mehr spielt. Für die Tests werden jedoch einheitliche Dateitypen verwendet.

Texturgröße

Bei den Vortests ergab sich bereits eine maximale Texturauflösung von 4096x4096 Pixeln. Eine höhere Auflösung der Texturen führte zu signifikanten Framerateeinbrüchen. Bei dem angewendeten Testfall, reicht eine Textur von 2048x2048 Pixeln aus. Ein höheres Detail ist nicht notwendig und würde daher das Ergebnis nur unnötig beeinflussen.

Modelle als Testobjekte

Um ein möglichst breites Spektrum an verschiedenen Anwendungsszenarien und Stilen abzudecken, werden drei verschiedene Konzepte mit den beiden Workflows modelliert.

3.4. Vortests für das Testen mit dem Unity Profiler

Als Vortest wurden vier verschiedene Modelle getestet. Der Vortest sollte zeigen, welche Werte am relevantesten sind. Sowohl der Prozess `PostLateUpdate.PresentAfterDraw` als auch `PostLateUpdate.PlayerSendFrameStarted` sind nur Leerlaufprozesse. Bei `PresentAfterDraw` handelt es sich um Wartezeiten der CPU auf Ergebnisse der GPU. `PlayerSendFrameStarted` ist ein Prozess der GPU welcher die Frameraten limitiert. Dieser hält also die Ergebnisse der GPU zurück um die Framerate auf 60 bzw. 30Fps zu begrenzen und so ein möglichst stabiles Bild zu gewährleisten.

Entwicklung des Testverfahrens

Diese beiden haben somit keinen Einfluss auf die eigentliche Berechnungszeit und müssen deshalb nicht weiter betrachtet werden.

Beim weiteren Betrachten der CPU fiel auf, dass die eigentliche Berechnungszeit abzüglich des Leerlaufprozesses nur noch bei ca. 4ms lag. Eine Leistungslimitierung durch die CPU ist damit ausgeschlossen und die CPU Leistung wird damit in den folgenden Test nicht weiter betrachtet. Die Auslastung lag bei allen anderen Tests in ähnlichen Bereichen.

Bei der Betrachtung der GPU Ergebnisse fiel auf, dass neben `PlayerSendFrameStarted` nur ein weiterer Prozess signifikante Auswirkung auf die gesamte Berechnungszeit besaß. In `PostLateUpdate.FinishFrameRendering` werden unter anderem die Zeiten für das Rendern von Geometrie verbucht. Alle weiteren Prozesse der GPU lagen bei nahezu 0% der gesamten Rechenzeit. Somit können diese vernachlässigt werden und es wird in den folgenden Test nur `FinishFrameRendering` betrachtet.

3.5. Aufbau des Profiler Tests

Nach Auswertung der Vortestdaten ergibt sich, dass die Zeit von `PostLateUpdate.FinishFrameRendering` als entscheidender Wert für die Ermittlung der Leistungsfähigkeit zu sehen ist. Der folgende Test soll nun die Berechnungszeiten der GPU für alle Modelle ermitteln. Hierfür werden alle Modelle getestet und die maximale Berechnungsdauer von `FinishFrameRendering` notiert. Dies geschieht wieder in Abhängigkeit der Entfernung von 0,5m und 2m.

3.6. Polygonperformance

Bei der Ermittlung der Polygonperformance wird ein Material mit Unlit Shader aus dem HoloToolkit ohne Beleuchtung und Lighmaps verwendet. Als Modell wird eine unterteilte Kugel verwendet. Der Abstand beträgt sowohl einen halben als auch zwei Meter. Die Kugel besitzt außerdem einen Durchmesser 0,25 Meter.

	Abstand 0,5m	Abstand 2,0m
6.000		
25.000		
100.000		

175.000		
250.000		

Tabelle 4: Tabellenvorlage für Tests der Polygonperformance mit der Polygonanzahl der jeweiligen Modelle

3.7. Testverfahren Game Creation Workflow

Das Testverfahren für den Game Creation Workflow beinhaltet verschiedene Zusammensetzungen von Texturen, um so Auskunft darüber geben zu können, welche Texturen am meisten für Performance Einbrüche sorgen. Es wird jeweils ein Test für die Berechnungszeiten aus dem Profiler als auch für die Framerate durchgeführt.

	Abstand 0,5m	Abstand 2,0m
Mesh und Albedo		
Mesh und NM		
Mesh und AO		
Mesh und Meta		
Mesh und Albedo + NM		
Mesh und NM + AO		
Mesh und Albedo + AO + Meta		
Mesh und alle Texturen		

Tabelle 5: Tabellenvorlage für Tests mit dem Game Asset Creation Workflow

3.8. Testverfahren Textureless Workflow

Das Testverfahren für den Textureless Workflow besitzt weniger Testoptionen als der Game Asset Creation Workflow. Es entfällt das Testen von Texturen. Es wird jeweils ein Test für die Berechnungszeiten aus dem Profiler als auch für die Framerate durchgeführt.

	Abstand 0,5m	Abstand 2,0m
Mit Shader		

Tabelle 6: Tabellenvorlage für Tests mit dem Textureless Workflow

4. Tests

4.1. Tests Polygonperformance

Polygonperformance Profilertest

	Abstand 0,5m	Abstand 2,0m
6.000	3,5ms	2,3ms
25.000	4,6ms	3,1ms
100.000	8,1ms	6,7ms
175.000	12,8ms	10,3ms
250.000	15,4ms	15,5ms

Tabelle 7: Profilertest Polygonperformance

Polygonperformance Frameratetest

	Abstand 0,5m	Abstand 2,0m
6.000	60Fps	60Fps
25.000	60Fps	60Fps
100.000	60Fps	60Fps
175.000	30Fps	30Fps
250.000	30Fps	30Fps

Tabelle 8: Frameratetest Polygonperformance

4.2. Tests Modell 1

Game Asset Creation Workflow Profilertest

	Abstand 0,5m	Abstand 2,0m
Mesh und Albedo	14ms	4,2ms
Mesh und NM	13,4ms	4,5ms
Mesh und AO	11,7ms	4ms
Mesh und Meta	12,3ms	4,4ms
Mesh und Albedo + NM	14,7ms	4,6ms
Mesh und NM + AO	15,0ms	4,5ms
Mesh und Albedo + AO + Meta	13,1ms	4,3ms
Mesh und alle Texturen	16,4ms	5ms

Tabelle 9: Modell 1 Profilertest Game Asset Creation Workflow

Game Asset Creation Workflow Frameratetest

	Abstand 0,5m	Abstand 2,0m
Mesh und Albedo	30-60Fps (30Fps bei hoher Füllrate)	60Fps
Mesh und NM	30-60Fps (30Fps bei hoher Füllrate)	60Fps
Mesh und AO	30-60Fps (30Fps bei hoher Füllrate)	60Fps
Mesh und Meta	30-60Fps (30Fps bei hoher Füllrate)	60Fps
Mesh und Albedo + NM	30-60Fps (30Fps bei hoher Füllrate)	60Fps
Mesh und NM + AO	30-60Fps (30Fps bei hoher Füllrate)	60Fps
Mesh und Albedo + AO + Meta	30-60Fps (30Fps bei hoher Füllrate)	60Fps
Mesh und alle Texturen	30-60Fps (30Fps bei hoher Füllrate)	60Fps

Tabelle 10: Modell 1 Frameratetest Game Asset Creation Workflow

Textureless Workflow Profilertest

	Abstand 0,5m	Abstand 2,0m
Vertex Shader	4,2ms	2ms

Tabelle 11: Modell 1 Profilertest Textureless Workflow

Textureless Workflow Frameratetest

	Abstand 0,5m	Abstand 2,0m
Vertex Shader	60Fps	60Fps

Tabelle 12: Modell 1 Frameratetest Textureless Workflow

4.3. Tests Modell 2

Game Asset Creation Workflow Profilertest

	Abstand 0,5m	Abstand 2,0m
Mesh und Albedo	17,7ms	8,1ms
Mesh und NM	21,1ms	9,5ms

Tests

Mesh und AO	17,2ms	8,1ms
Mesh und Meta	13,4ms	7,8ms
Mesh und Albedo + NM	16,8ms	8,1ms
Mesh und NM + AO	17,3ms	8,2ms
Mesh und Albedo + AO + Meta	13ms	7,5ms
Mesh und alle Texturen	16,7ms	9ms

Tabelle 13: Modell 2 Profilertest Game Asset Creation Workflow

Frameratetest Game Asset Creation Workflow

	Abstand 0,5m	Abstand 2,0m
Mesh und Albedo	30-60FPS	40-60Fps (meist 60Fps)
Mesh und NM	30-60FPS (meist im Bereich 30-45)	30-60FPS (meist im Bereich 45-60)
Mesh und AO	30-60Fps	60FPS (kurze Einbrüche auf 40)
Mesh und Meta	30-60Fps (bei bewegung höher)	60FPS (kleinere Einbrüche auf 40)
Mesh und Albedo + NM	30-60Fps	30-60Fps(meist 40-55)
Mesh und NM + AO	30-60FPS (meist im Bereich 29-45)	30-60Fps(meist 40-60)
Mesh und Albedo + AO + Meta	30-60Fps	30-60Fps(meist 45-60)
Mesh und alle Texturen	30-50Fps	30-60Fps(meist 35-55)

Tabelle 14: Modell 2 Frameratetest Game Asset Creation Workflow

Textureless Workflow Profilertest

	Abstand 0,5m	Abstand 2,0m
Vertex Shader	4,5ms	2,6ms

Tabelle 15: Modell 2 Profilertest Textureless Workflow

Frameratetest Textureless Workflow

	Abstand 0,5m	Abstand 2,0m
Vertex Shader	60Fps	60Fps

Tabelle 16: Modell 2 Frameratetest Textureless Workflow

4.4. Tests Modell 3 Game

Game Asset Creation Workflow Profilertest

	Abstand 0,5m	Abstand 2,0m
Mesh und Albedo	5,9ms	1,9ms
Mesh und NM	7,9ms	2,4ms
Mesh und AO	5,7ms	1,9ms
Mesh und Meta	6,7ms	2,2ms
Mesh und Albedo + NM	7,8ms	2,4ms
Mesh und NM + AO	9ms	2,7ms
Mesh und Albedo + AO + Meta	5,8ms	2,1ms
Mesh und alle Texturen	7,9ms	2,5ms

Tabelle 17: Modell 3 Profilertest Game Asset Creation Workflow

Game Asset Creation Workflow Frameratetest

	Abstand 0,5m	Abstand 2,0m
Mesh und Albedo	60Fps	60Fps
Mesh und NM	60Fps	60Fps
Mesh und AO	60Fps	60Fps
Mesh und Meta	60Fps	60Fps
Mesh und Albedo + NM	60Fps	60Fps
Mesh und NM + AO	45-60Fps(nur kleine Einbrüche)	60Fps
Mesh und Albedo + AO + Meta	60Fps	60Fps
Mesh und alle Texturen	45-60Fps(nur kleine Einbrüche)	60Fps

Tabelle 18: Modell 3 Frameratetest Game Asset Creation Workflow

Textureless Workflow Profilertest

	Abstand 0,5m	Abstand 2,0m
Vertex Shader	2,4ms	1,2ms

Tabelle 19: Modell 3 Profilertest Textureless Workflow

Tests

Textureless Workflow Frameratetest

	Abstand 0,5m	Abstand 2,0m
Vertex Shader	60Fps	60Fps

Tabelle 20: Modell 3 Frameratetest Textureless Workflow

5. Auswertung

In den folgenden Abschnitten erfolgt eine Auswertung der in den Tests ermittelten Werte.

5.1. Polyperformance

Bei der Polygonperformance zeigt sich, dass die HoloLens bis 100.000 Polygone flüssige Frameraten bei allen Abständen liefert. Bereits bei 175.000 Polygonen beträgt die Framerate nur noch 30Fps. Die Berechnungszeiten betragen zwischen 10,3 und 12,8ms. Bei 250.000 Polygonen liegt die Berechnungszeit bei 15,4-15,5ms. Damit zeigt sich, dass die Berechnungszeiten sowohl bei 175.000 als auch bei 250.000 noch eine Framerate von 60Fps liefern müsste. Jedoch wird bereits bei einer Berechnungszeit von 10,3ms ein Frame übersprungen. Für eine Framerate von 60Fps ist allerdings eine Berechnungszeit von unter 16,67ms notwendig. Es gab keinen ersichtlichen Grund, warum die Framerate künstlich auf 30Fps abgesenkt wurde.

5.2. Modell 1 Game Asset Creation Workflow

Bei einem Abstand von zwei Metern zum Modell lagen die Berechnungszeiten immer im einstelligen Millisekunden Bereich. Die Framerate betrug bei allen verwendeten Materialien 60Fps. Dabei ist zu berücksichtigen, dass aufgrund der Silhouette des Modells, die Füllrate immer sehr gering war. Bei einem Abstand von nur 0,5 Metern waren die Berechnungszeiten im Durchschnitt um das Dreifache höher. Die Maxima lagen dabei zwischen 11,7-16,5 Millisekunden. Die Framerate war selten über 30Fps. Dies geschah immer dann, wenn Teile des Modells betrachtet wurden, welche die Füllrate absenkten. Des Weiteren besitzt das Modell noch eine recht hohe Anzahl an Polygonen und Materialien, was den Effekt verstärkte.

5.3. Modell 2 Game Asset Creation Workflow

Bei diesem Modell ist die Framerate sowohl bei 0,5m als auch bei 2m nicht konstant. Durch die Form des Modells, wird sowohl bei einem Abstand von zwei Metern als auch bei 0,5 Metern eine sehr hohe Füllrate erreicht. Die Berechnungszeiten lagen in den Maxima zwischen 7,5 und 9 Millisekunden bei einem Abstand von zwei Metern. Mit der Verringerung auf 0,5m stieg die Berechnungszeit auf 13-21,1 Millisekunden an. Auffällig ist auch, dass bei

Auswertung

Materialien mit Normal Map im Schnitt eine höhere Berechnungszeit benötigt wurde.

5.4. Modell 3 Game Asset Creation Workflow

Modell 3 lieferte geringe Berechnungszeiten bei allen Materialien und Abständen. Die Berechnungszeiten liegen bei zwei Metern Abstand bei 1,9-2,5 Millisekunden und bei 0,5 Metern bei 5,7-9 Millisekunden. Das Modell liefert nahezu konstante Frameraten bei allen Materialien. Die Polygonanzahl lag dabei aber im Mittelfeld der getesteten Modelle. Die Füllrate war hingegen etwas geringer als bei den anderen Modellen. Die Framerate beträgt bei zwei Metern Abstand, egal bei welchem Material 60Fps. Bei 0,5 Metern Abstand kommt es bei Materialien mit Normal Map zu kleineren Einbrüchen. Die Framerate liegt jedoch auch hier meist bei 60Fps. Des Weiteren ist zu beachten, dass das Modell aus nur einem Objekt besteht und daher auch nur ein Material besitzt, Die Anzahl der Drawcalls ist damit am geringsten.

5.5. Zusammenfassung Performance Game Asset Creation Workflow

Der Game Asset Creation Workflow bietet nicht die Optimale Performance. In vielen Versuchen fiel die Framerate unter die zu wünschenden 60Fps und die Berechnungszeit stieg dabei stark an. Dabei fällt auf, dass die größeren Einbrüche und die längeren Berechnungszeiten immer bei einem Abstand von 0,5m auftraten. Der geringere Abstand ist dabei gleichzusetzen mit einer höheren Füllrate des Displays. Das Modell wurde jeweils aus verschiedenen Blickwinkeln betrachtet, was meist eine Veränderung der Framerate und der benötigten Berechnungszeit hervorrief. Diese Schwankungen deuten wiederum auf die Füllrate als entscheidenden Faktor bei der Berechnungszeit hin.

Eine weitere Auffälligkeit ist, dass Materialien mit einer Normal Map meist eine leicht erhöhte Berechnungszeit im Vergleich zu Materialien ohne Normal Map aufweisen. Dies wird vor allem beim Vergleich der Materialien mit Albedo-Ambient Occlusion-Metallic und Normal-Ambient Occlusion deutlich. Hier ergibt sich mit zwei Texturen eine schlechtere Performance, als mit drei Texturen. Damit zeigt sich, dass die Normal Map einen schwerwiegenderen Einfluss auf die Berechnungszeit hat, als alle anderen getesteten Texturen.

Auswertung

Auffällig ist außerdem, dass wenn sich mehrere Modelle und Materialien in einer Szene befinden die Berechnungszeit höher ist, als bei einem Modell und einem Material. Die Anzahl der Polygone war bei dem einzelnen Modell sogar höher. Jedoch variierte auch die Füllrate zwischen den einzelnen Modellen. Somit kann nicht genau gesagt werden, ob die auf eine unterschiedliche Füllrate oder die oben genannten Parameter für die höheren Berechnungszeiten verantwortlich sind. Jedoch ist es allgemein so, dass Materialien einen sehr starken Einfluss auf die Performance haben. Dies kommt durch die erhöhte Anzahl der benötigten Drawcalls.

5.6. Erscheinungsbild Game Asset Creation Workflow

Mit dem Game Asset Creation Workflow ist es möglich, mehrere Materialitäten in einem Material darzustellen. Mit der Metallic Map ist es möglich in einem Modell metallische und nicht-metallische Objekte Oberflächen darzustellen. Da eine Metallic Map Oberflächen jedoch meist dunkler macht, ist diese jedoch nicht immer von Vorteil. Auf der einen Seite wird damit ein Materialunterschied sichtbar, auf der anderen Seite kann dieser nicht sehr effektiv von der HoloLens dargestellt werden. Somit sind nicht alle Formen der Metalle möglich. Auch Glanzpunkte sind mit der HoloLens kaum sichtbar.

Bei der Verwendung der Normal Map fiel auf, dass diese teilweise schlecht zu erkennen war. Die Ambient Occlusion Map hingegen bot eine deutliche Verbesserung in Richtung realistischer Darstellung war.



Abbildung 21: Erscheinungsbild von Modell 1 mit dem Game Asset Creation Workflow



Abbildung 22: Erscheinungsbild von Modell 1 mit dem Game Asset Creation Workflow

5.7. Mögliche Verbesserungen des Game Asset

Creation Workflow für HoloLens Anwendungen

Da die Normal Map sowohl erheblichen Einfluss auf die Performance besitzt, jedoch der optischen Verbesserung nur geringfügig beisteuert, sollte diese aus dem Workflow genommen werden. Eine Ambient Occlusion Map besitzt ähnliche Fähigkeiten kleines Detail darzustellen und besitzt dabei eine bessere Performance. Des Weiteren kann eine Albedo Map mit einer Ambient Occlusion Textur in einer Diffuse Textur zusammengeführt werden. Somit wird zusätzlich noch Texturspeicher gespart.

Detail kann zusätzlich hinzugefügt werden, indem es modelliert wird. Dies ist aber nur bedingt möglich. Die maximale Polygonanzahl der HoloLens darf dabei nicht überschritten werden.

Trotz der Verwendung eines Shaders welcher für die HoloLens optimiert wurde, sollte ein eigens für den speziellen Fall geschriebener Shader die Performance noch einmal steigern können. In dem verwendeten Shader waren diverse Slots nicht belegt. Nicht verwendeter Code in einem Shader, sorgt hierbei für erhöhten Rechenaufwand ohne Mehrnutzen.

5.8. Modell 1 Textureless Workflow

Der Textureless Workflow zeigt bei diesem Modell weder Einbrüche in der Renderzeit, noch in der Framerate. Seine durchschnittliche Renderzeit beträgt nur 3,1ms. Die Framerate liegt immer bei konstanten 60Fps.

5.9. Modell 1 Textureless Workflow

Die Renderzeiten sind wie bei Modell 1 sehr gering. Mit einer durchschnittlichen Renderzeit von 3,5ms liegt die Renderzeit knapp über der von Modell 1. Die Framerate beträgt konstante 60Fps.

5.10. Modell 1 Textureless Workflow

Auch bei Modell 3 gibt es keine Performanceeinbrüche. Hier liegt die durchschnittliche Renderzeit bei nur 1,8ms und ist somit die niedrigste aller getesteten Modelle. Die Framerate beträgt in jedem Fall konstant 60Fps.

5.11. Zusammenfassung Performance Textureless Workflow

Der Textureless Workflow bietet eine sehr gute Performance. Die dargestellten 3D Modelle wurden zu jeder Zeit flüssig und mit sehr niedrigen Berechnungszeiten dargestellt. Es kam zu keinem Zeitpunkt zu Einbrüchen der Framerate oder erhöhten Berechnungszeiten während der Laufzeit. Sowohl die Polygonanzahl als auch die Füllrate des Displays hatten bei den getesteten Modellen nur minimalen Einfluss auf die Berechnungszeiten und keinen Einfluss auf die Framerate. Jedoch zeigt sich, dass bei einem Abstand von 0,5m die Berechnungszeiten doppelt so hoch waren wie bei 2m. Durch die sehr geringen Berechnungszeiten hatte dies jedoch keinen Einfluss auf die Framerate.

Der Textureless Workflow bietet mit seinen geringen Berechnungszeiten sehr viel Spielraum für weitere Geometrie. Die Polygonanzahl erreicht bei weitem nicht das ermittelte Limit an Polygonen der HoloLens. Nimmt man die Berechnungszeit von 15,5ms bei 250,000 Polygonen, so liegt das Modell mit der höchsten Polygonanzahl bei knapp 10% der theoretisch möglichen Polygone.

5.12. Erscheinungsbild Textureless Workflow

Bei dem Erscheinungsbild muss der Textureless Workflow jedoch Abstriche machen. Das Erstellen von natürlich wirkenden Materialitäten ist nicht möglich. So kann die reflektierende Eigenschaft einer Oberfläche nur über einen konstanten Wert festgelegt werden. Die Ambient Occlusion in den Vertices ist meist zu gering um diese zuverlässig und detailliert zu erstellen. Des Weiteren ist der Workflow für einige Arten von Modellen nahezu ungeeignet. Vegetation kann

Auswertung

nur schwer dargestellt werden und besitzt wenn dann nur eine "feste" Struktur. Das Modellieren von einzelnen Blättern und Grashalmen wäre dabei zu rechenintensiv. Somit besitzen texturelose immer eine gewisse Stilisierung. Eine Darstellung von Mikrostrukturen wie zum Beispiel Stoff ist ebenso nicht möglich.

Es ist jedoch möglich sowohl echtzeit als auch statische Beleuchtung zu verwenden. Diese fanden in den Tests jedoch keine Anwendung.



Abbildung 23: Erscheinungsbild von Modell 1 mit dem Textureless Workflow

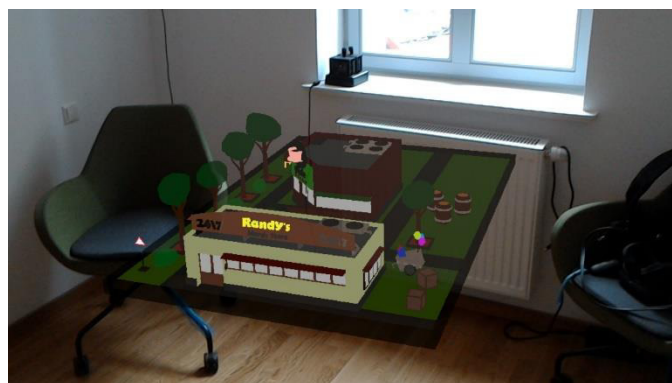


Abbildung 24: Erscheinungsbild von Modell 1 mit dem Textureless Workflow

5.13. Mögliche Verbesserungen des Textureless Workflow für HoloLens Anwendungen

Ein Lösungsansatz für die beschränkte Auswahl an Materialitäten wäre das Zuweisen mehrerer Materialien. Das bedeutet, dass zum Beispiel alle metallischen Flächen in einer Gruppe zusammengefasst werden und dann ein separates Material zugewiesen bekommen. Ein Nachteil daran wäre jedoch, dass es zu einer erhöhten Anzahl an Objekten und Materialien in der Szene kommen würde. Diese könnten die Berechnungszeiten wieder negativ beeinflussen.

Auswertung

Für die Verwendung von Schatten ist ein sich nicht überlappendes UV Mapping notwendig. Dieses kann jedoch beim Import des Modells von Unity selbst erzeugt werden. Dabei ist jedoch zu beachten, dass auch hier eine höhere Berechnungszeit zu erwarten ist. Des Weiteren werden die vorher berechneten Lichtinformationen in Texturen gespeichert und verwendet. Aufgrund dessen finden diese in den vorangegangenen Tests keine Verwendung. Lightmaps besitzen jedoch eine sehr gute Performance und können nahezu bedenkenlos verwendet werden. Eine statische Beleuchtung wäre damit eine gute Option um Texturlose 3D Modelle zu beleuchten und damit die Optik zu verbessern.

5.14. Vergleich der Performance

Bei der Performance zeigen sich starke Unterschiede. So liegen die benötigten Berechnungszeiten des Textureless Workflows deutlich unter denen des Game Asset Creation Workflows. Mit einem gemessenen Maximalwert von 21,1ms beim Game Asset Creation Workflow liegt dieser mehr als viermal so hoch wie die gemessenen 4,5ms beim Textureless Workflow. Dies spiegelt sich in allen Tests wieder. In keinem Test kommt der Game Creation Workflow nur Ansatzweise an den Textureless Workflow heran.

Der Textureless Workflow bot in allen Tests eine stabile Framerate von 60Fps wohingegen der Game Creation Workflow in vielen Fällen unter die angestrebten 60Fps fiel.

Der Performancevergleich zeigt, dass der Textureless Workflow in den konkreten Beispielen die deutlich bessere Performance liefert.

5.15. Vergleich der Erscheinungsbilder

Der Game Creation Workflow und der Textureless Workflow unterscheiden sich in optischer Hinsicht sehr stark. Eine gewisse Stilisierung ist bei dem Textureless Workflow nicht zu verhindern, wohingegen der Game Creation Workflow die Möglichkeit besitzt, realistischere Modelle darzustellen. Hierbei ist aber zu beachten, dass eine realistische Darstellung mit der HoloLens aufgrund der Limitierung und der damit einhergehenden besonderen Darstellungsform nicht möglich ist.

5.16. Auswertung in Betracht auf die Füllrate

Bei allen Modellen zeigt sich, dass desto näher sich ein Objekt am Betrachter befindet, umso schlechter ist die Berechnungszeit. Damit steht die Füllrate in direktem Zusammenhang mit der benötigten Berechnungszeit. Jedoch kann eine genaue Bestimmung der Füllrate stattfinden.

Im Game Asset Creation Workflow besitzt Modell 2, welches sowohl weniger Polygone, als auch Objekte und Materialien besitzt wie Modell 1, aufgrund seiner Form eine deutlich höhere Füllrate. Die Füllrate beträgt bei 2m Abstand geschätzt etwa 70%. Bei einem Abstand von 0,5m liegt die Füllrate ungefähr bei 95-100%. Bei Modell 1 betragen die Füllraten bei 0,5m Abstand geschätzte 65-75% und bei 2m etwa 30%.

Vergleicht man nun die Berechnungszeiten von Modell 1 mit einem Abstand von 0,5 mit denen von Modell 2 mit einem Abstand von 2m, so ist zu erkennen, dass trotz ungefähr gleicher Füllrate, die Berechnungszeiten von Modell 1 noch einmal 4-8ms über denen von Modell 2 liegen.

Daraus ergibt sich, dass sowohl die Füllrate als auch andere Faktoren einen deutlichen Einfluss auf die Berechnungszeiten besitzen. In welcher Höhe genau, ist dabei nicht zu ermitteln.

5.17. Aufgetretene Probleme bei der Erstellung der 3D Modelle

Während der Erstellung der Modelle gab es verschieden Probleme und Hindernisse die aufgetreten sind. Der folgende Abschnitt soll auf diese kurz eingehen und Lösungsansätze liefern.

5.17.1. Probleme Textureless Workflow

Bei der Erstellung des ersten Modells wurden recht häufig Booleans Operationen eingesetzt, um die Polygonanzahl so gering wie möglich zu halten. Dies führte dazu, dass der Edgeflow an vielen Stellen nicht optimal war. Beim darauffolgenden Erstellen der Ambient Occlusion, reichte die zu geringe Polygonanzahl dann nicht aus, um diese detailliert darzustellen. Dies machte ein Hinzufügen von neuen Unterteilungsebenen notwendig. Aufgrund des schlechten

Auswertung

Edgeflows kam es jedoch zu Löchern und sich überlagernden Flächen, welche von Hand nachgebessert werden mussten.

Prinzipiell sollte also mit einer leicht erhöhten Polygonanzahl gearbeitet werden, besonders in Bereichen wo mit Ambient Occlusion zu rechnen ist. Die Erstellung von Einzelobjekten erscheint sinnvoll. An Stellen wo Einzelobjekte ineinanderstecken und damit Ambient Occlusion auftritt, sollte separat die Polygonanzahl erhöht werden. Dies ist sowohl ressourcenschonender als auch optisch genauer als das Unterteilen des gesamten Objektes.

Bei der Erstellung von Vegetation musste auf das Erstellen einzelner Blätter verzichtet werden, da diese die Polygonanzahl über die maximale Polygonanzahl getrieben hätte. Vegetation kann daher nur sehr stilistisch dargestellt werden.

Modell 2 lieferte von Anfang an bessere Ergebnisse. Jedoch kam es trotz einer erhöhten Polygonanzahl und besserem Edgeflow zu Ungenauigkeiten beim Erstellen der Ambient Occlusion. Beim Erstellen der Ambient Occlusion wird jedoch ein separater Vertex Color Kanal angelegt, welcher einfach zu bearbeiten ist. In diesen konnte nun mit Schwarz Ambient Occlusion eingezeichnet und mit Weiß gelöscht werden.

Ein Problem, welches nur bei Modell 3 auftrat war, dass die Rückseiten von Flächen in Unity nicht gerendert werden, oder im Gegenzug einen deutlich rechenintensiveren Shader benötigt hätten um diese darzustellen. Das Modell wurde dahingehend nachgebessert, dass an durchsichtigen Stellen Flächen mit nach außen gerichteten Normalen erstellt wurden.

5.17.2. Probleme Game Creation Workflow

Beim ersten anfertigen eines UV Mappings gab es bei einem Modell überlappende UV Shells. Nach dem ersten Baken kam es daraufhin zu falscher Ambient Occlusion und falschen Normalen an verschiedenen Stellen. Daraufhin mussten die bereits erstellten UV Shells erneut ohne Überlappung ausgelegt werden.

Da die Szene mitunter Objekte enthielt, welche mehrfach vorkamen, wurden diese nur einmal modelliert und einmal ein passendes UV Mapping für das Objekt erstellt. Verschiedene Objekte wurden danach zu Gruppen zusammengefasst um das Texturieren zu erleichtern und Texturenspeicher zu sparen. Dafür wurden

Auswertung

die UV Shells der einzelnen Objekte in mehreren größeren UV Maps zusammengefasst. Die Einzelobjekte wurden im Anschluss vervielfacht und in der Szene verteilt.

Beim Baken mit und ohne Cage von Normal Maps, kam es an unterschiedlichen zu Ungenauigkeiten. So wurden zwei Normal Maps erstellt und diese dann so in Photoshop so zusammengefügt, dass alle Fehler entfernt wurden.

Ein weiteres Problem, welches jedoch reinweg auf die Verwendung von MayaLT zurückzuführen ist, dass der gesculpote Anteil von Modell 3 nicht exportiert werden konnte, da die maximale Anzahl der Polygone überschritten wurde. Deshalb musste diese Normal und Ambient Occlusion Map mit den MayaLT internen Tools erstellt werden.

Beim Export weiterer Teile kam es dann mit der verwendeten Software zu dem Problem, dass nur Objekte mit Tris oder Quads gebaked werden konnten. Somit musste in MayaLT das Cleanup Werkzeug verwendet werden, um alle nicht konformen Stellen auszubessern.

Bei der Exportierung von Texturen aus Photoshop kam es dann zu einem weiteren Problem. Die Metallic Map musste als Tiff und nicht als PNG exportiert werden, da sonst der Alphakanal nicht funktionierte. Dies hängt mit dem Export aus Photoshop zusammen. Dabei wird, wenn keine Ebene eine Transparenz besitzt, der Alphakanal wieder verworfen. Dies ist bei der erstellten Metallic Map der Fall, da im Alphakanal die Smoothness Map liegt.

6. Fazit

Diese Arbeit hat versucht die Frage zu beantworten, welcher Workflow für mobile AR Geräte der bessere ist und ob die Microsoft HoloLens genügend Leistung besitzt um 3D Modelle effektiv darzustellen. Die Arbeit bezog sich dabei auf den Game Asset Creation Workflow und den Textureless Workflow. Um dies zu überprüfen, wurde eine Testreihe entwickelt, welche sowohl die benötigten Berechnungszeiten des Grafikchips als auch die Frameraten verschiedener Modelle ermittelte. Diese wurden unter der Betrachtung mehrerer Faktoren ermittelt.

Die ermittelten Werte zeigen, dass beim Game Creation Workflow die Normal Map für erhöhte Berechnungszeiten und einen Einbruch der Framerate in nahezu allen Fällen sorgt. Des Weiteren ist zu beobachten, dass es bei geringem Abstand sowohl beim Game Asset Creation Workflow als, auch beim Textureless Workflow die benötigte Berechnungsdauer im Vergleich zu einer größeren Entfernung stark ansteigt und damit die Framerate oft auf unter 60Fps sinkt. Daraus ergibt sich, dass die Füllrate direkt in Verbindung mit der Framerate steht.

Bei den Tests mit dem Textureless Workflow kam es bei keinem Versuch zu Frameraten unter 60Fps und die Berechnungszeiten lagen deutlich unter denen des Game Asset Creation Workflows. Somit bietet der Textureless Workflow wie erwartet eine deutlich bessere Performance als der Game Asset Creation Workflow.

Aufgrund der sehr geringen Berechnungszeiten des Textureless Workflows und der sehr geringen Menge an verwendeten Polygonen im Vergleich zur getesteten maximalen Polygonmenge, ist ein Vergleich der Erscheinungsbilder sehr schwierig. Detail hätte als Geometrie hinzugefügt werden können und die Optik positiv beeinflussen können.

Anzumerken ist außerdem, dass nur statische Modelle ohne sonstige Inhalte wie Sound, Animationen oder Interaktion getestet wurden. Sämtliche Rechenleistung stand also für das Rendering der 3D Modelle zur Verfügung.

Die Nachteile des Textureless Workflows ergaben sich dann im Bereich des Erscheinungsbildes. Der Textureless Workflow besitzt nur begrenzte Möglichkeiten bei der Erstellung von Materialien und Effekten. Eine realistische

Fazit

Darstellung von Objekten ist mit ihm in den meisten Fällen nicht möglich und begrenzt deshalb sein Anwendungsspektrum. Der Nachteil des Game Creation Workflows liegt hingegen bei der Performance. Eine realistischere und freiere Gestaltung von 3D Modellen ist die Stärke des Workflows. Die Optik kann ohne Restriktion frei erstellt werden.

Die ursprüngliche Idee, nur die Framerate als Indikator für die Leistungsfähigkeit zu nehmen, erwies sich als nicht ausreichend. Sie musste daher durch das erneute Testen mit dem Unity Profiler ergänzt werden. Da die Framerate jeweils nur 30Fps oder 60Fps betragen kann und andere Werte auf Rundungsfehler der Fps Anzeige zurückzuführen sind.

Die Füllrate des Displays ist ein nicht genau zu bestimmender Faktor der Arbeit. Zwar hat diese großen Einfluss auf die Berechnungszeit und die Framerate, jedoch ist eine genaue Bestimmung der Füllrate nicht möglich. Die Füllrate kann daher nur geschätzt werden.

Aufgrund der vielen Faktoren, welche Auswirkung auf die Performance haben, kann nicht genau bestimmt werden, welcher Faktor tatsächlich den meisten Einfluss hat und welcher die Performance begrenzt.

Beim Vergleich der ermittelten Berechnungszeiten und der Framerate fiel auf, dass die ermittelte Framerate nicht immer mit der errechneten Framerate anhand der Berechnungszeiten einherging. In dieser Arbeit wurde keine Ursache für die auftretenden Differenzen gefunden. Jedoch liegt die Vermutung nahe, dass es sich hierbei um eine automatische Absenkung der Framerate durch das Betriebssystem Windows 10 handelt, welches auf der HoloLens verwendet wird. Dafür verantwortlich könnte die neben der GPU verbaute HPU sein, welche nicht die erforderliche Leistung besitzt. Jedoch ist anhand der ermittelten Werte in den meisten Fällen genügend Leistung vorhanden um stabile Frameraten von 60Fps zu garantieren.

Der Anwendungsfall mit der HoloLens kann somit prinzipiell eine Aussage treffen, wo die Stärken und Schwächen eines Workflows liegen und welcher die bessere Performance bietet. Jedoch variiert die Leistungsfähigkeit mobiler Endgeräte sehr stark und die Technik entwickelt sich sehr schnell weiter. Eine Auswahl des Workflows je nach Plattform und zu entwickelnder Anwendung sollte deshalb

unbedingt erfolgen. Eine einfache Aussage welcher Workflow der bessere ist, ist nicht möglich und stark von dem zu realisierenden Projekt abhängig.

7. Ausblick

Die Optimierung hinsichtlich technischer Seite blieb weitestgehend unbeachtet. Eine weitere Optimierung von Shadern, Texturen und Modellen kann die Performance der bestehenden und hier getesteten Workflows weiter verbessern. Vor allem der Game Asset Creation Workflow bietet mit seinem großen Umfang einen großen Spielraum für Optimierungen.

Software Updates, sowohl für die HoloLens als auch Unity und die Weiterentwicklung bestimmter Techniken und Standards werden die Performance weiter verbessern. Schon während dieser Arbeit gab es signifikante Verbesserungen, was die Performance von verschiedenen Shadern anging. Somit sind die konkret ermittelten Ergebnisse auf die festgelegten Rahmenbedingungen begrenzt.

Anzumerken ist auch, dass die HoloLens zum Zeitpunkt der Tests bereits knapp 1,5 Jahre auf dem Markt war. Die Leistungsfähigkeit liegt deshalb bereits deutlich hinter der von aktuellen High-End Smartphones wie dem iPhoneX (September 2017), Samsung Galaxy S8 (März 2017) und aktuellen Tablets. Die Berechnungszeiten und Frameraten sollten auf High-End Smartphones deutlich besser sein, als auf der HoloLens.

Die Arbeit bietet einen guten Leitfaden zum Auswählen eines Workflows für ein Projekt, besonders für ein HoloLens Projekte. Durch modernere und leistungsfähigere Hardware in anderen mobilen Endgeräten, kann der Game Creation Workflow wahrscheinlich deutlich besser angewendet werden, als auf der Microsoft HoloLens.

Literaturverzeichnis

- Burrows, A. (2015, 23.April). What To Know When Creating Next Gen Assets [Tutorial]. Abgerufen von <https://cgmasters.net/free-tutorials/what-to-know-when-creating-next-gen-assets/> (abgerufen am 11.09.2017)
- CGFort PTY LTD (2017, 26.Juni). How do I use the different texture maps? [Artikel]. Abgerufen von <http://poliigon.helpscoutdocs.com/article/30-what-are-texture-maps-for> (abgerufen am 10.07.2017)
- De Jong, S. (o.D.). Tutorial – Mesh Rendering [Tutorial]. Abgerufen von <http://www.hourences.com/tutorials-mesh-rendering/> (abgerufen am 13.09.2017)
- HoloToolkit (o.D.). HoloToolkit [GitHub]. Abgerufen von <https://github.com/Microsoft/MixedRealityToolkit-Unity> (abgerufen am 23.09.2017)
- Karlsson, R. (2013, Mai). Improving Baked Textures as a Content Creator (Bachelor-Thesis). Abgerufen von <http://www.diva-portal.org/smash/get/diva2:832182/FULLTEXT01.pdf> (abgerufen am (21.09.2017)
- Khronos (2017, 21.Februar). Rendering Pipeline Overview [Wiki] Abgerufen von https://www.khronos.org/opengl/wiki/Rendering_Pipeline_Overview (abgerufen am 22.09.2017)
- Lewicki, M. (2016, 07.Oktober), Was ist eigentlich aus der Google Glass geworden? [Artikel]. Abgerufen von <https://www.techbook.de/techstyle/wearables/was-ist-eigentlich-aus-google-glass-geworden> (abgerufen am 24.07.2017)
- Mastin, L. (o.D.). Aesthetics [Website]. Abgerufen von http://www.philosophybasics.com/branch_aesthetics.html (abgerufen am 19.07.2017)
- Michaud, S. (2015, 10.August). What exactly is a drawcall [Artikel] Abgerufen von <https://www.pcper.com/reviews/Editorial/What-Exactly-Draw-Call-and-What-Can-It-Do> (abgerufen am 20.09.2017)

- Microsoft (o.D.). Holograms 100 [Anleitung]. Abgerufen von https://developer.microsoft.com/en-us/windows/mixed-reality/holograms_100 (abgerufen am 26.07.2017)
- NVIDIA Corporation (o.D.). Enabling Ambient Occlusion in Games [Artikel]. Abgerufen von <http://www.geforce.com/whats-new/guides/ambient-occlusion#1> (abgerufen am 10.07.2017)
- Pettit, N. (2015, 17.November). The Beginner's Guide to Physically Based Rendering in Unity [Artikel]. Abgerufen von <http://blog.teamtreehouse.com/beginners-guide-physically-based-rendering-unity> (abgerufen am 10.07.2017)
- Pettit, N. (2015, 08.Dezember). Understanding Normal Maps [Artikel]. Abgerufen von <http://blog.teamtreehouse.com/understanding-normal-maps> (abgerufen am 10.07.2017)
- Photoshop (2017). Adobe Photoshop [Webseite]. Abgerufen von <https://www.adobe.com/de/products/photoshop.html> (abgerufen am 11.09.2017)
- Programming with OpenGL: Advanced Rendering [PDF Datei] (o.D.). Abgerufen von <http://www.inf.pucrs.br/flash/tcg/aulas/texture/texmap.pdf> (abgerufen am 04.07.2017)
- Reallusion Inc. (o.D.). Physically Based Rendering–PBR [Dokumentation]. Abgerufen von http://manual.reallusion.com/Character_Creator_2/ENU/2.0/15_Multiple_Channel_Texture_Mapping/PBR/PBR.htm (abgerufen am 10.07.2017)
- Rubino, D. (2016, 02.Mai). Microsoft HoloLens - Here are the full processor, storage and RAM specs [Artikel]. Abgerufen von <https://www.windowscentral.com/microsoft-hololens-processor-storage-and-ram> (abgerufen am 13.09.2017)
- Sala, T. (2014, 06.Februar). Making textureless 3D work [Blog]. Abgerufen von <http://blog.littlechicken.nl/creating-a-textureless-pure3d-look-as-seen-in-oberons-court/> (abgerufen am 11.09.2017)

Literaturverzeichnis

Slick, J. (2016, 21.Oktober) 3D Model Components – Vertices, Edges, Polygons and More [Artikel]. Abgerufen von <https://www.lifewire.com/3d-model-components-1952> (abgerufen am 04.07.2017)

Unity Technologies (2017, 08.Juni). Lighting Modes [Dokumentation]. Abgerufen von <https://docs.unity3d.com/Manual/LightModes.html> (abgerufen am 04.07.2017)

Unity Technologies (o.D.). Occlusion Map [Dokumentation]. Abgerufen von <https://docs.unity3d.com/Manual/StandardShaderMaterialParameterOcclusionMap.html> (abgerufen am 10.07.2017)

Zucconi, A. (o.D.). A Gentle Introduction to Shaders [Tutorial]. Abgerufen von <https://unity3d.com/de/learn/tutorials/topics/graphics/gentle-introduction-shaders> (abgerufen am 18.07.2017)

Anhang

Anhang 1

Verwendete Shader

Ausgeführter Test	Verwendeter Shader
Beleuchtungstest	Unity Standard Mobile Diffuse
Polygonperformance	HoloToolkit Unlit Configurable
Textureless Workflow	Unlit Basic (selbst erstellt)
Game Creation Workflow	HoloToolkit Standard Fast

Der geschriebene Shadercode für den Textureless Workflow ist unter Anhang 2 zu finden.

Anhang 2

Unlit Basic:

```
Shader "custom/unlit_basic"
{
    Properties{
    }

    SubShader{
        Tags{ "RenderType" = "Opaque" }
        LOD 100

        Pass{
            Lighting Off
        }
    }
}
```

Anhang

Anhang 3

Concept Modell 1

Quelle: <https://artbygfxengine.wordpress.com/3d-stuff/#jp-carousel-328>
(abgerufen am 22.09.2017)



Abbildung 25: Concept Art Modell 1,

Concept Modell 2

Quelle: <http://talros.deviantart.com/art/QR-Props-01-405815960>
(abgerufen am 22.09.2017)



Abbildung 26: Concept Art Modell 2

Concept Modell 3

Quelle: <https://oscarvisual.wordpress.com/2009/06/09/vehicles-and-props/#jp-carousel-374> (abgerufen am 22.09.2017)

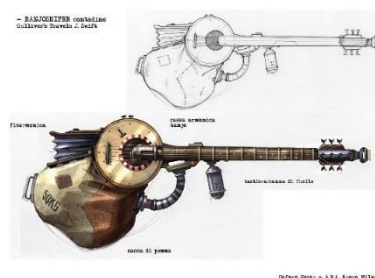


Abbildung 27: Concept Art Modell 3

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 23.September.2017

Andreas Hauswald