

# **10. TEXTUREN | DREHBUCH**

## **COMPUTERGRAFIK.ONLINE**

Hochschule Furtwangen University | Fakultät Digitale Medien

Betreuer: Prof. Jirka Dell'Oro-Friedl | Projektstudium SoSe 18

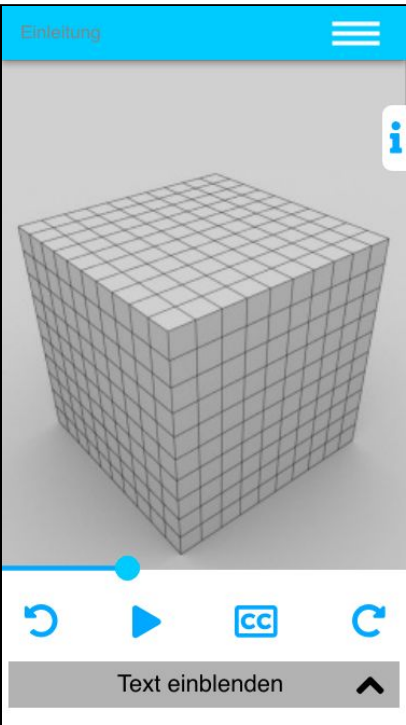
Version: 1.1 | Letzte Änderung: 27.07.2018

Autor: Lisa Würstle MKB 4

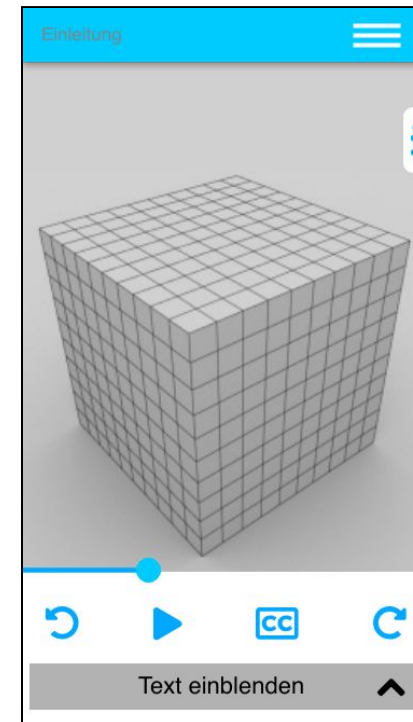
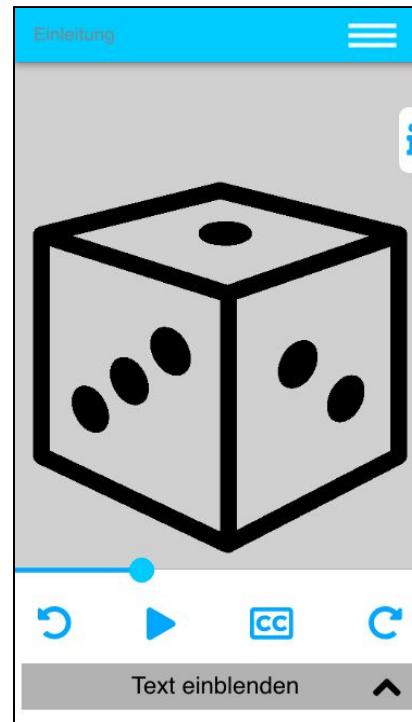
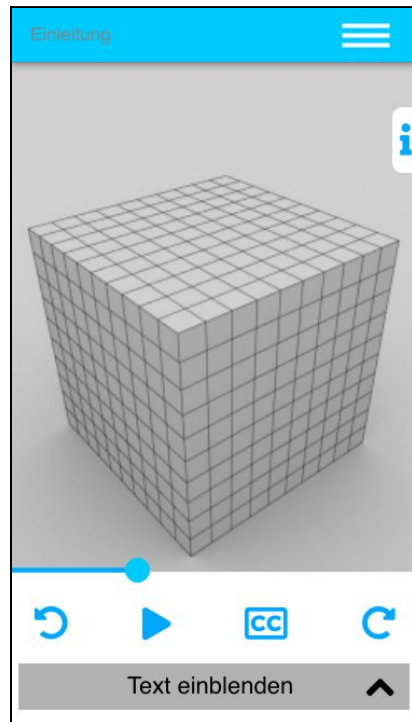
## Inhaltsverzeichnis

<b>10.1</b>	<b>Einleitung</b>	<b>2, 3</b>
<b>10.2</b>	<b>Einleitung - Interaktion</b>	<b>4, 5</b>
<b>10.3</b>	<b>Texturkoordinaten und UV-Mapping</b>	<b>6</b>
	10.3.1 Texturkoordinaten	6, 7
	10.3.2 UV-Mapping	8, 9
	10.3.3 UV-Mapping - Interaktion	10
<b>10.4</b>	<b>Mip-Mapping</b>	<b>11, 12</b>
<b>10.5</b>	<b>Bump-Mapping</b>	<b>13</b>
<b>10.6</b>	<b>Bump-Mapping - Interaktion</b>	<b>14, 15</b>
<b>10.7</b>	<b>Displacement-Mapping</b>	<b>16</b>
<b>10.8</b>	<b>Displacement-Mapping - Interaktion</b>	<b>17, 18</b>
<b>10.9</b>	<b>Environment-Mapping</b>	<b>19</b>
	9.10.1 Sphärisches Environment-Mapping	20
	9.10.2 Kubisches Environment-Mapping	21, 22


## 10.1 Einleitung

Finaler Screen	Sprechertext	Screentext (i)	Regieanweisung
	<p>#100101 Eine Textur dient im Bereich der Computergrafik dazu, dreidimensionale Objekte realistischer und detailreicher darzustellen.</p> <p>#100102 Die Textur besteht aus einem zweidimensionalen Bild, welches dem Objekt eine Struktur verleiht. Wie die Textur auf das jeweilige Objekt projiziert wird, kann auf verschiedene Art und Weise geschehen. Diese Verfahren nennt man Mapping-Verfahren.</p>	<ul style="list-style-type: none"> <li>- Ziel: 3D-Objekt realistischer wirken zu lassen</li> <li>- Lösung: 2D-Bild = 2D-Textur</li> <li>- verschiedene Mapping-Verfahren</li> </ul>	<p>#100101 Einblenden eines Kubus ohne Texturierung Der Kubus fängt an sich um die eigene Achse zu drehen</p> <p>#100102 Eine Würfeltextur erscheint auf dem Kubus für etwa 3 Sekunden, danach sieht man wieder den Kubus ohne Textur</p> <p>Dieser Vorgang wiederholt sich bis der Sprecher fertig ist</p>

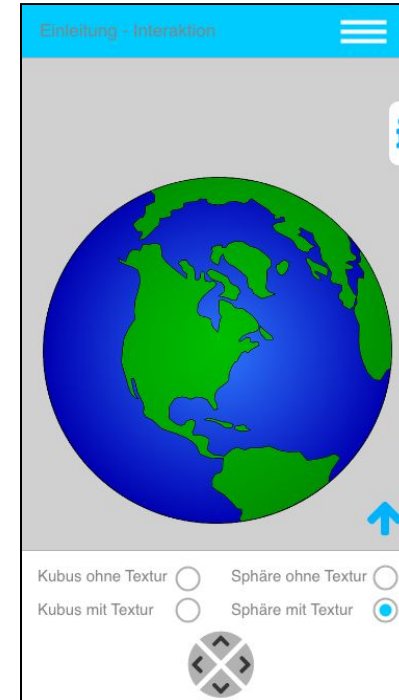
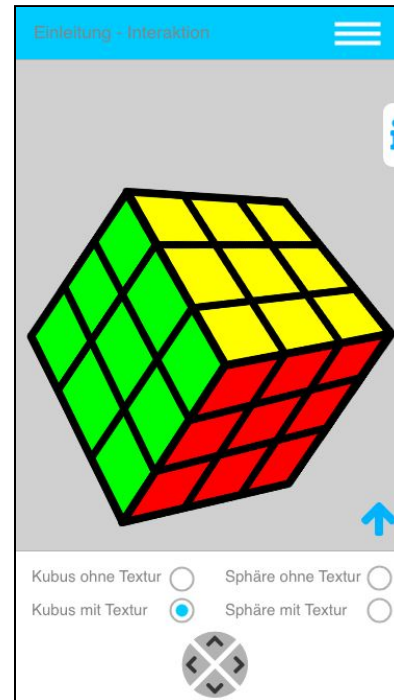
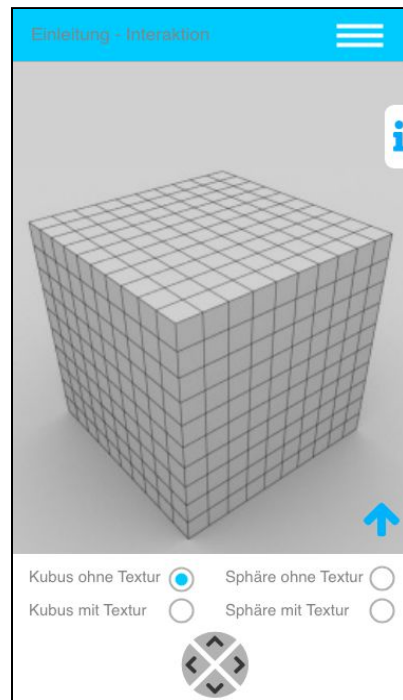
Grafikablauf:



## 10.2 Einleitung - Interaktion

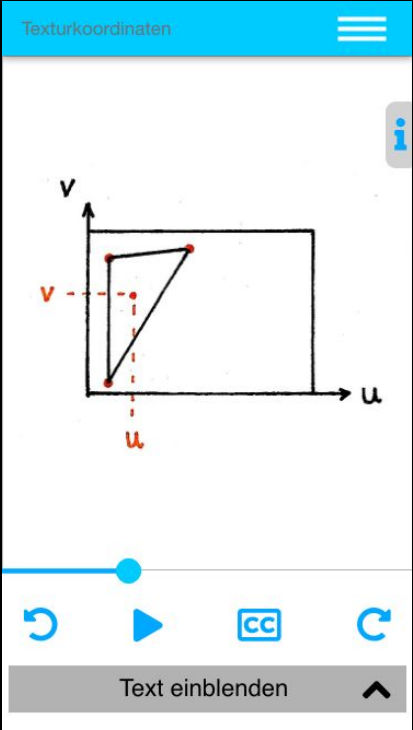
Finaler Screen	Sprechertext	Screentext (i)	Regieanweisung
	<p>#100201 Wähle den Kubus oder die Sphäre aus. Lade eine von dir ausgewählte Textur hoch und lasse dir den Vergleich zwischen dem Objekt ohne und mit Textur anzeigen. Betrachte das 3D-Objekt dabei von allen Seiten.</p>	<p>Wähle den Kubus oder die Sphäre aus. Lade eine von dir ausgewählte Textur hoch und lasse dir den Vergleich zwischen dem Objekt ohne und mit Textur anzeigen. Betrachte das 3D-Objekt dabei von allen Seiten.</p>	<p>#100201 Aufgabe wird gesprochen (über das i kann sich der Nutzer die Aufgabe anzeigen lassen)</p> <p>Klickt der Nutzer auf den Upload-Pfeil öffnet sich ein kleines Fenster in dem er lokale Dateien hochladen kann (.png oder .jpg)</p> <p>Der Nutzer kann zwischen den Auswahlmöglichkeiten "Kubus ohne Textur", "Kubus mit Textur", "Sphäre ohne Textur" und "Sphäre mit Textur" auswählen, wobei die Punkte "mit Textur" nur ausgewählt werden können, wenn der Nutzer ein Bild hochgeladen hat</p> <p>3D-Objekt kann vom Nutzer beliebig über das Steuerkreuz gedreht und von allen Seiten betrachtet werden</p>

Grafikablauf:

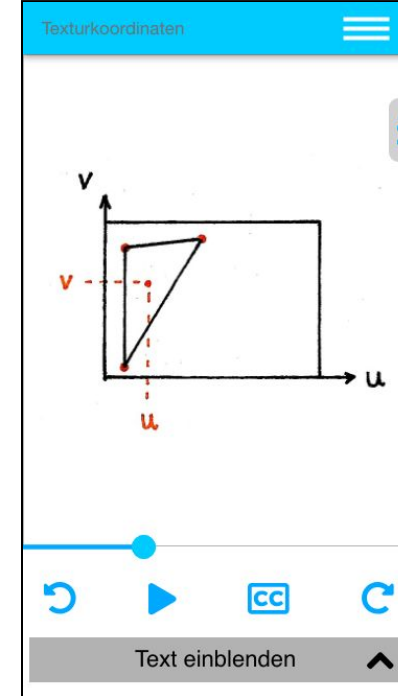
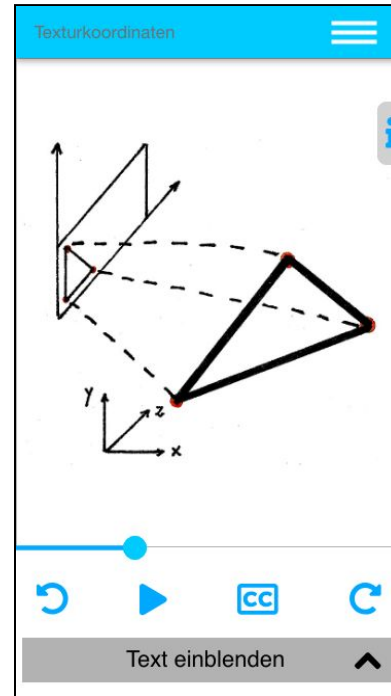
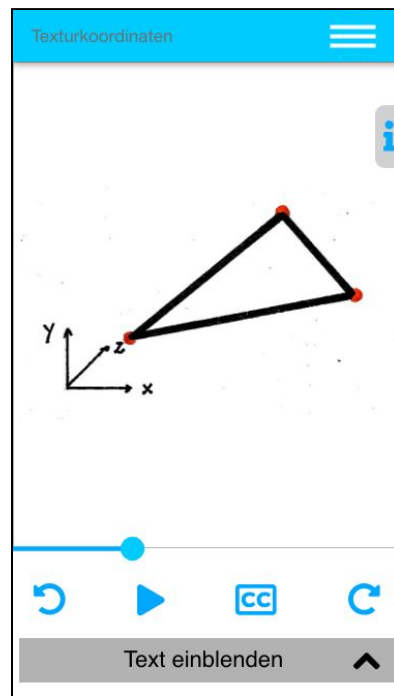


## 10.3 Texturkoordinaten und UV-Mapping

### 10.3.1 Texturkoordinaten

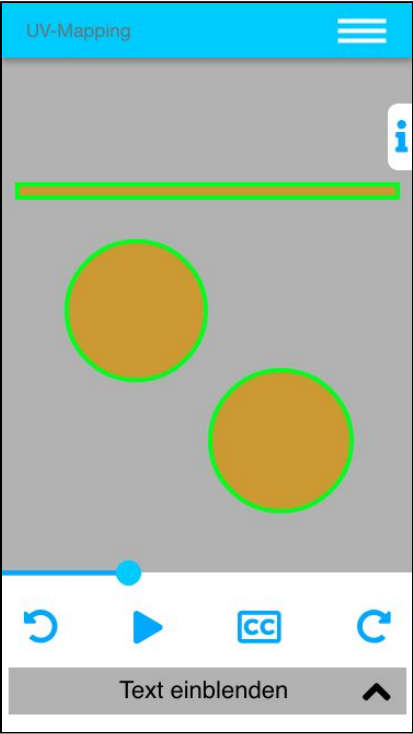
Finaler Screen	Sprechertext	Screentext (i)	Regieanweisung
	<p>#10030101 Nun stellt sich die Frage, wie die Textur auf dem 3D-Objekt positioniert wird. Für jeden Eckpunkt eines Polygons werden zuerst die lokalen Objektraum-Koordinaten im 3D-Raum definiert. Diese Koordinaten werden anschließend in Texturkoordinaten umgewandelt.</p> <p>#10030102 Es erhält somit jeder Eckpunkt des Polygons eine genaue Position auf der Texture-Map. Diese Position ist dann ein Textur-Pixel, auch Texel genannt.</p> <p>#10030103 Die Texturkoordinaten werden in einem kartesischen Koordinatensystem dargestellt, die Achsen werden in der Regel mit u und v beschriftet.</p>	<ul style="list-style-type: none"> <li>- Umwandlung von Objektraum-Koordinaten (x, y, z) in Texturkoordinaten (u, v)</li> <li>- Textur-Pixel = Texel</li> <li>- Texel: Position des Polygons auf der Texture-Map</li> </ul>	<p>#10030101 Einblenden eines Dreiecks im 3D-Raum  Eckpunkte werden markiert</p> <p>#10030102 Einblenden eines kartesischen Koordinatensystems in dem eine 2D-Textur liegt Dreieck wird in 2D-Raum projiziert (Zoom zum Koordinatensystem) und jeder Eckpunkt erhält eine genaue Position auf der Textur</p> <p>#10030103 Die Achsen werden mit u und v beschriftet Ein sichtbarer Pixel erhält eine genaue Texturposition</p>

Grafikablauf:

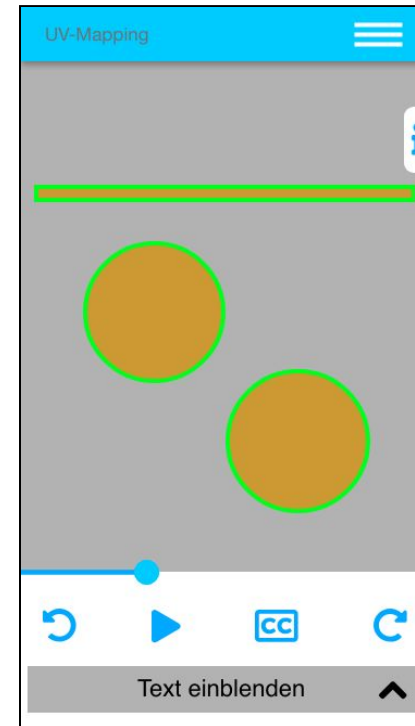
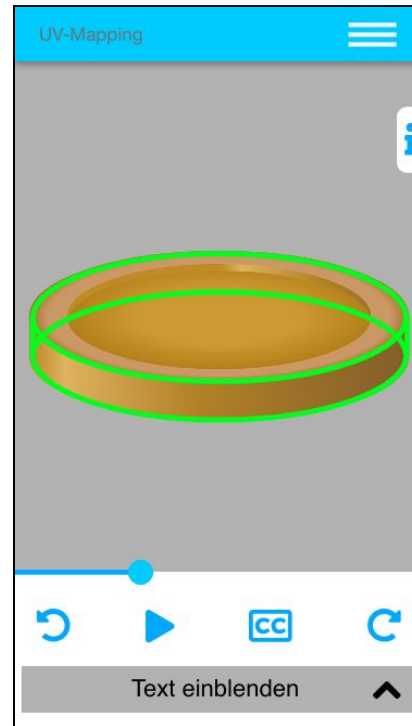




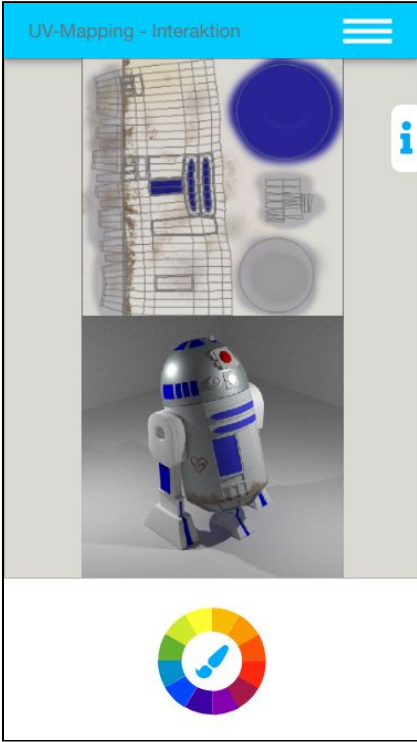
### 10.3.2 UV-Mapping

Finaler Screen	Sprechertext	Screentext (i)	Regieanweisung
	<p>#10030201 Beim UV-Mapping bildet man die Texture-Map mit einer einfachen Abwicklung. Der Prozess wird auch als "Unwrapping" bezeichnet.</p> <p>#10030202 Bildlich kann man sich die Abwicklung so vorstellen, dass das 3D-Objekt an bestimmten Stellen "aufgeschnitten" und anschließend "abgewickelt" wird, wie bei dieser 1€ Münze.</p> <p>#10030203 Versuche den Prozess der Abwicklung nachzuvollziehen.</p>	<ul style="list-style-type: none"> <li>- Texture-Map entsteht durch einfache Abwicklung</li> <li>- Abwicklung = Unwrapping</li>   <li>- Objekt wird "aufgeschnitten" und anschließend "abgewickelt"</li> </ul>	<p>#10030201 Einblenden einer Münze</p> <p>#10030202 Münze wird an ihren Kanten aufgeschnitten und anschließend abgewickelt</p> <p>Texture Map (siehe Screen) entsteht</p>

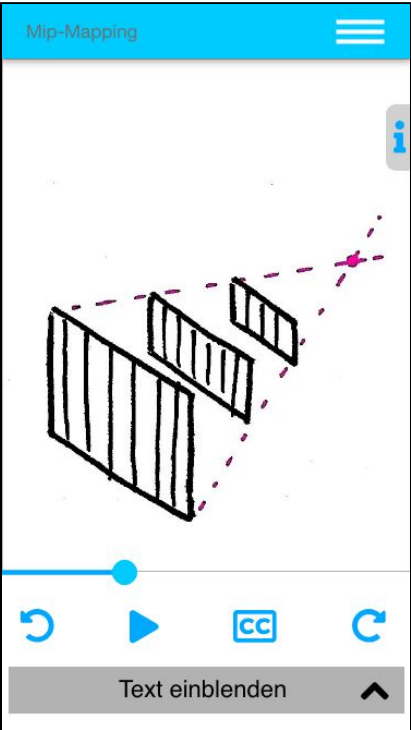
Grafikablauf:



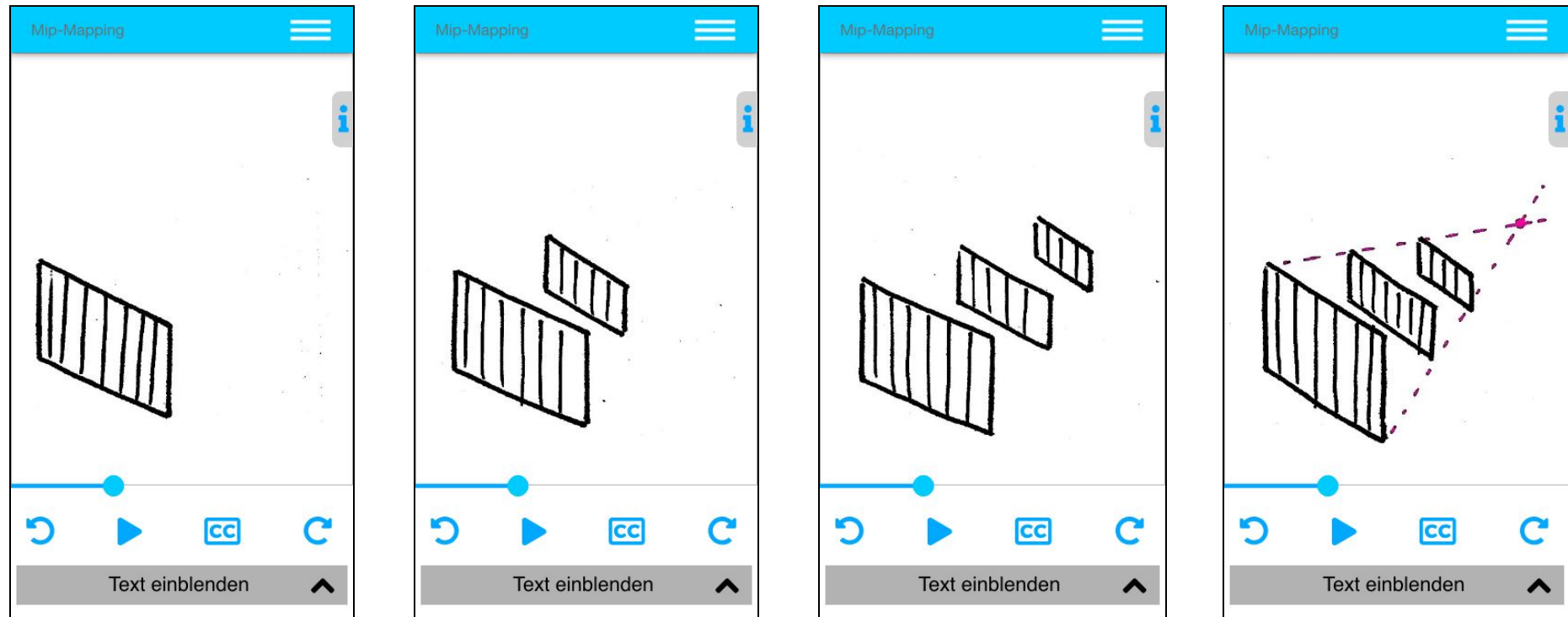
### 10.3.3 UV-Mapping - Interaktion

Finaler Screen	Sprechertext	Screentext (i)	Regieanweisung
	<p>#10030301 Wähle den Pinsel und suche dir eine beliebige Farbe aus. Nun kannst du die abgewickelte Texture-Map des Roboters, oder den Roboter direkt bemalen.</p>	<p>Wähle den Pinsel und suche dir eine beliebige Farbe aus. Nun kannst du die abgewickelte Texture-Map des Roboters, oder den Roboter direkt bemalen.</p>	<p>#10030301 Aufgabe wird gesprochen (über das i kann sich der Nutzer die Aufgabe anzeigen lassen)  Klickt der Nutzer auf die Texture-Map vergrößert sich diese  Klickt der Nutzer auf den Roboter, vergrößert sich dieser</p>

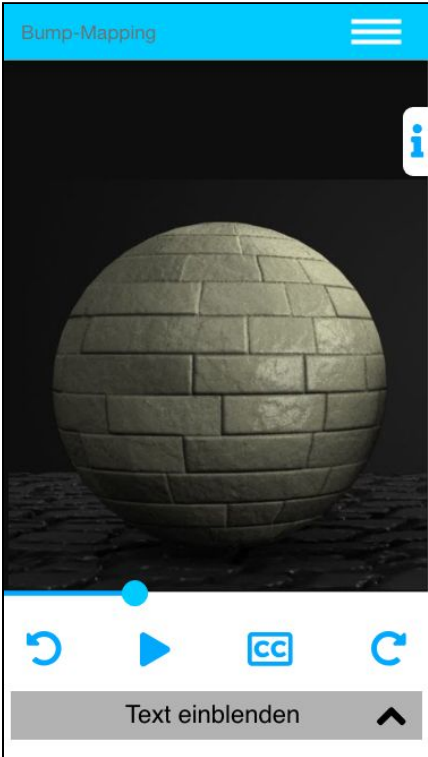
## 10.4 Mip-Mapping

Finaler Screen	Sprechertext	Screentext (i)	Regieanweisung
	<p>#100401 Das Mip-Mapping ist eine Anti-Aliasing-Methode. In der Computergrafik tritt der Alias-Effekt beim Abtasten von Bildern auf, wodurch Muster entstehen, die im Originalbild nicht enthalten sind.</p> <p>#100402 Beim Mip-Mapping werden von einer Textur mehrere vorberechnete Texture-Maps mit sinkender Auflösung berechnet. Dabei wird in jedem Schritt die Kantenlänge des Originals halbiert.</p> <p>#100403 Ist das texturierte Polygon nahe beim Betrachter, kommt eine große Texture-Map zum Einsatz. Ist es jedoch weiter entfernt, wird eine kleinere Texture-Map benutzt.</p> <p>#100404 Der große Vorteil des Mip-Mappings besteht darin, dass die verschiedenen Maps zum Zeitpunkt des Renderns bereits vorberechnet sind.</p>	<ul style="list-style-type: none"> <li>- Anti-Aliasing-Methode</li> <li>- Kantenlänge wird bei jedem Schritt halbiert</li> <li>- Objekt nahe beim Betrachter: große Texture-Map</li> <li>- Objekt weiter entfernt vom Betrachter: kleine Texture-Map</li> </ul>	<p>#100401 Einblenden einer großen Texture-Map</p> <p>#100402 Einblenden einer zweiten Map mit halbierten Kantenlängen</p> <p>#100403 Einblenden einer dritten Map mit nochmals halbierten Kantenlängen</p> <p>Einblenden des Fluchtpunktes als gestrichelte Linien</p>

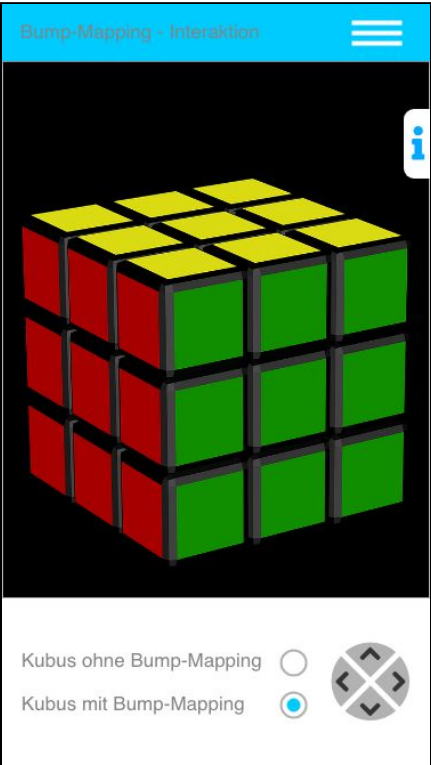
Grafikablauf:



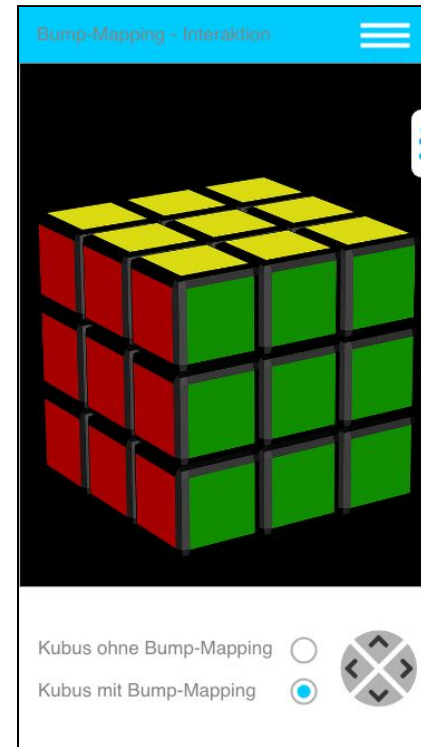
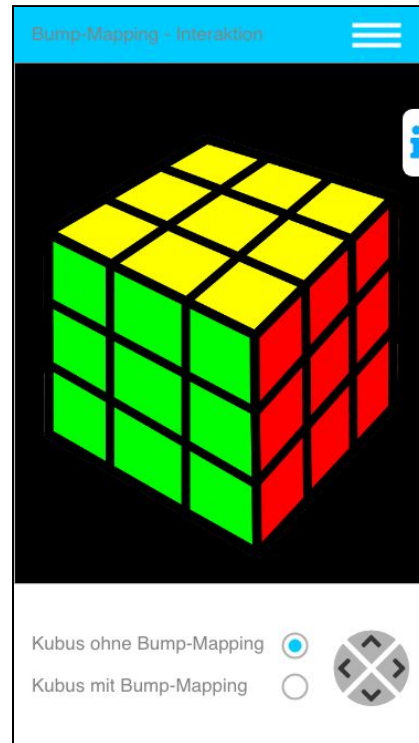
## 10.5 Bump-Mapping

Finaler Screen	Sprechertext	Screen text (i)	Regieanweisung
	<p>#100501 Eine 2D-Textur verleiht einem Objekt keine Oberflächeneigenschaften. Das Bump-Mapping dient dazu, einem 3D-Objekt diese Eigenschaften zu verleihen, ohne dabei die Geometrie des Objektes zu beeinflussen.</p> <p>#100502 Beim Bump-Mapping wird lediglich die Normale der Oberfläche verändert. Es wird mit Hilfe von Schattierung und Reflektion eine Illusion von Tiefe auf dem Objekt erzeugt.</p> <p>#100503 Der Vorteil des Bump-Mapping besteht darin, dass bei diesem Mapping-Verfahren weder der Speicherplatz, noch die Rendering-Zeit erhöht wird.</p>	<ul style="list-style-type: none"> <li>- Oberflächeneigenschaften ohne Beeinflussung der Geometrie</li> <li>- Veränderung der Normalen der Oberfläche</li> <li>- Illusion von Tiefe</li> <li>- keine Erhöhung des Speicherplatzes oder der Rendering-Zeit</li> </ul>	<p>#100501 Einblenden einer Sphäre mit Bump Mapping</p> <p>Sphäre dreht sich um sich selbst</p>

## 10.6 Bump-Mapping - Interaktion

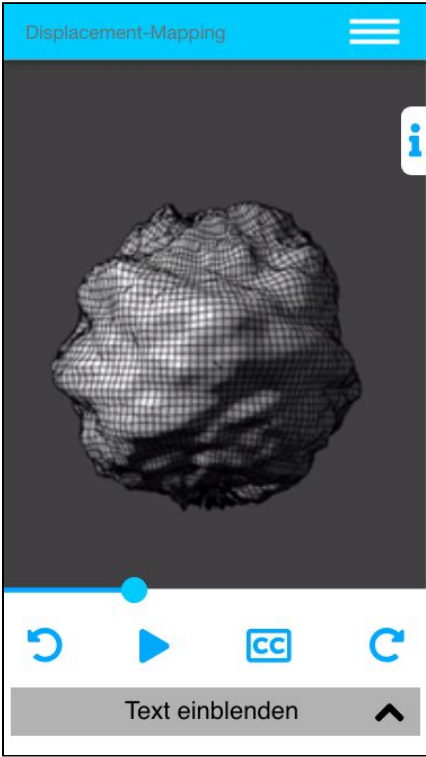
Finaler Screen	Sprechertext	Screentext (i)	Regieanweisung
	<p>#100601 Lasse dir den Unterschied zwischen einem 3D-Objekt mit Bump-Mapping und einem ohne anzeigen.</p>	<p>Lasse dir den Unterschied zwischen einem Objekt mit Bump-Mapping und einem ohne anzeigen.</p>	<p>#100601 Aufgabe wird gesprochen (über das i kann sich der Nutzer die Aufgabe anzeigen lassen)</p> <p>Auswahlmöglichkeiten "Kubus mit Bump Mapping" und "Kubus ohne Bump Mapping"</p> <p>Kubus kann über das Steuerkreuz gedreht werden</p>

Grafikablauf:

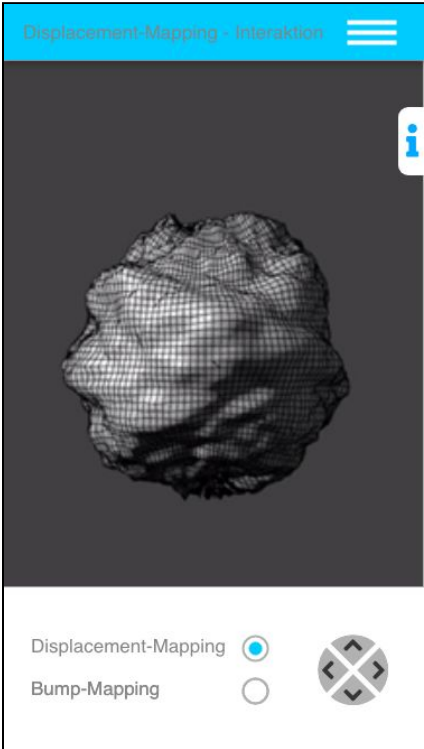




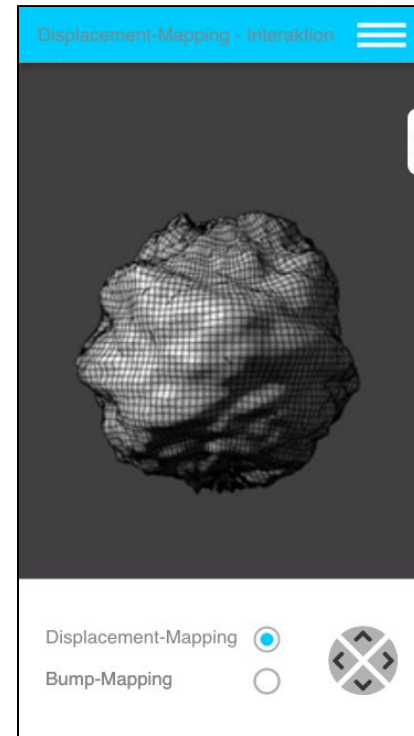
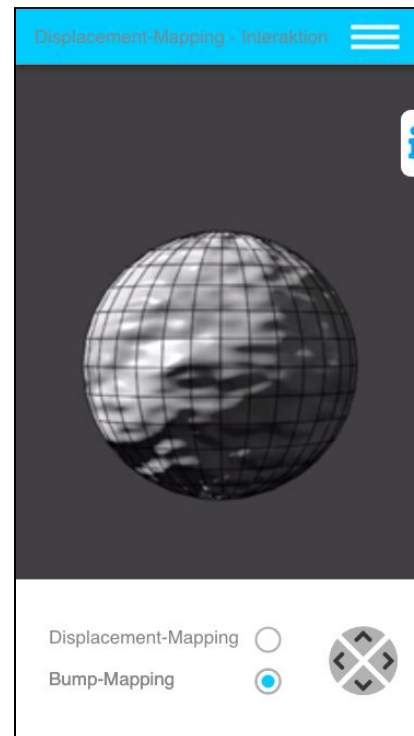
## 10.7 Displacement-Mapping

Finaler Screen	Sprechertext	Screen text (i)	Regieanweisung
	<p>#100701 Das Displacement-Mapping löst das selbe Problem wie das Bump-Mapping, es verleiht dem 3D-Objekt ebenfalls Oberflächeneigenschaften.</p> <p>#100702 Der Unterschied zwischen den beiden Mapping-Verfahren ist, dass beim Displacement Mapping die Geometrie des 3D-Objektes verändert wird.</p> <p>#100703 Durch "verschieben" beziehungsweise "verdrängen" des Materials werden dem 3D-Objekt die gewünschten Oberflächeneigenschaften verliehen.</p>	<ul style="list-style-type: none"> <li>- Oberflächeneigenschaften durch Veränderung der Geometrie des 3D-Objektes</li> <li>- displacement = "verschieben" bzw. "verdrängen" des Materials</li> </ul>	<p>#100701 Einblenden einer Sphäre mit Displacement-Mapping</p> <p>Sphäre dreht sich um sich selbst</p>

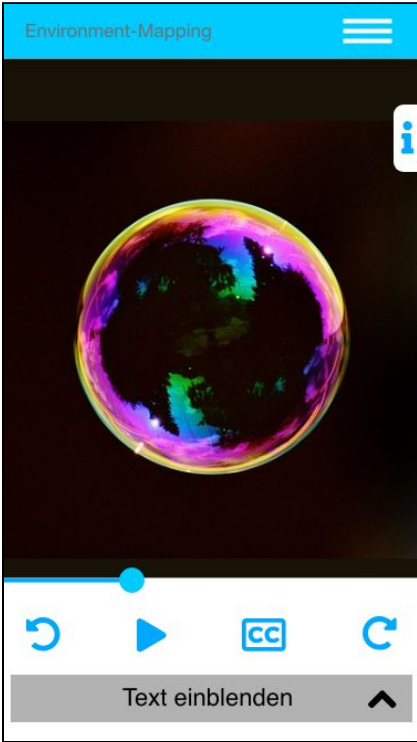
## 10.8 Displacement-Mapping - Interaktion

Finaler Screen	Sprechertext	Screentext (i)	Regieanweisung
	<p>#100801 Lasse dir den Unterschied zwischen dem Bump- und dem Displacement-Mapping anzeigen.</p>	<p>Lasse dir den Unterschied zwischen dem Bump- und dem Displacement-Mapping anzeigen.</p>	<p>#100801 Aufgabe wird gesprochen (über das i kann sich der Nutzer die Aufgabe anzeigen lassen)</p> <p>Auswahlmöglichkeiten "Displacement-Mapping" und "Bump-Mapping"</p> <p>Der Nutzer kann sich Objekt über das Steuerkreuz von allen Seiten ansehen</p>

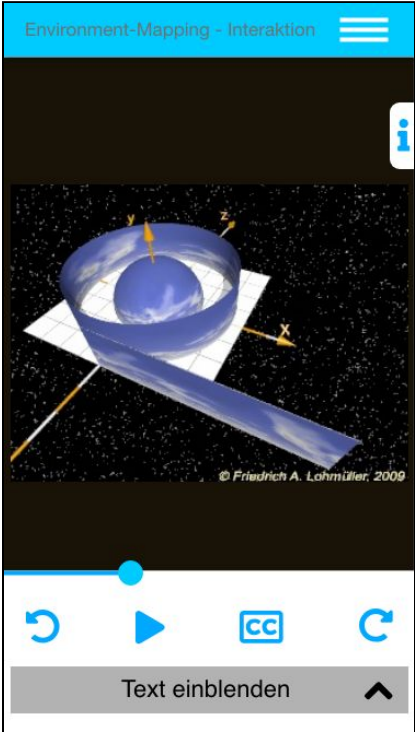
Grafikablauf:



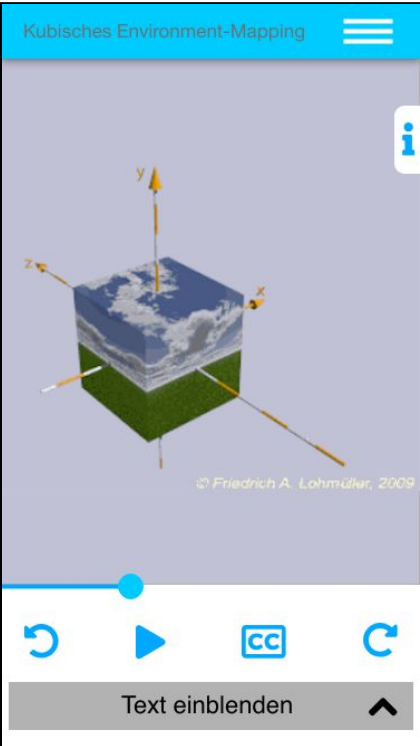
## 10.9 Environment-Mapping

Finaler Screen	Sprechertext	Screen text (i)	Regieanweisung
	<p>#100901 Mit dem Environment-Mapping-Verfahren werden in der Computergrafik spiegelnde 3D-Objekte simuliert.</p> <p>#100902 Dabei wird die Umgebung, bestehend aus Objekten und Lichtquellen, in der sich das Objekt befinden soll, als eine 2D-Textur gespeichert und auf das 3D-Objekt projiziert. Das 3D-Objekt muss im Verhältnis zu seiner Umgebung eher klein sein, damit die Spiegelung realistisch wirkt.</p> <p>#100903 Beim Environment-Mapping wird in zwei Arten unterschieden: das sphärische und das kubische Environment-Mapping.</p>	<ul style="list-style-type: none"> <li>- Simulation von spiegelnden Objekten</li> <li>- Sphärisches Environment-Mapping</li> <li>- Kubisches Environment-Mapping</li> </ul>	<p>#100901 Einblenden eines spiegelnden 3D-Objektes (hier: Seifenblase)</p>

### 10.9.1 Sphärisches Environment-Mapping

Finaler Screen	Sprechertext	Screentext (i)	Regieanweisung
	<p>#10090101 Beim sphärischen Environment-Mapping, wird die Umgebung auf das Innere einer hohlen Sphäre abgebildet. Dabei kann es zu Verzerrungen kommen.</p>	<ul style="list-style-type: none"> <li>- Sphärisches Environment Mapping: Abbildung der Umgebung auf das Innere einer hohlen Sphäre</li> </ul>	<p>#10090101 Eine 2D-Textur wird zylindrisch auf eine Sphäre gemappt</p> <p>Quelle und Code-Beispiel: Friedrich A. Lohmüller, <a href="http://www.f-lohmueller.de/pov_tut/backgrnd/p_sky8d.htm">http://www.f-lohmueller.de/pov_tut/backgrnd/p_sky8d.htm</a></p>

## 10.9.2 Kubisches Environment-Mapping

Finaler Screen	Sprechertext	Screentext (i)	Regieanweisung
	<p>#10090201 Bei dem kubischen Environment-Mapping, wird die Umgebung auf einem Kubus abgebildet. Dieses Verfahren hat den Vorteil, dass keine Verzerrungen entstehen, da die Fläche eben ist.</p>	<ul style="list-style-type: none"> <li>- Kubisches Environment-Mapping: Abbildung der Umgebung auf einen Kubus</li> </ul>	<p>#10090201 Eine 2D-Textur legt sich um den Kubus</p> <p>2D-Textur klappt wieder auf Vorgang wiederholt sich, solange der Sprecher spricht</p> <p>Quelle und Code-Beispiel: Friedrich A. Lohmüller, <a href="http://www.f-lohmueller.de/pov_tut/background/p_sky9d.htm">http://www.f-lohmueller.de/pov_tut/background/p_sky9d.htm</a></p>

Grafikablauf:

