

Japanischer Garten

Programmentwurf

Dokumentation

Computergrafik

im Studiengang Informatik
an der DHBW Ravensburg
Campus Friedrichshafen

von
Tim Bader
Daniel Erhard
Xena Letters
Sebastian Weber

22.11.2021

Bearbeitungszeitraum: 06.10.2021 - 22.11.2021
Name, Matrikelnummer, Kurs: Bader 7322851 TIM-19
Erhard 1757926 TIM-19
Letters 3176202 TIM-19
Weber 4258652 TIM-19
Gutachter der Dualen Hochschule: Prof. Dr. Jürgen Schneider

Inhaltsverzeichnis

Abbildungsverzeichnis	III
1 Einleitung	1
2 Grundidee der Szene	2
3 Modellierung mit Blender	3
3.1 Gestaltung der Objekte	3
3.2 Modelle	3
3.2.1 Gebäude und Designelemente	4
3.2.2 Tiere	9
3.2.3 Vegetation	12
4 Erstellung der Szene	17
4.1 Unity Grundlagen	17
4.2 Gesamtaufbau der Szene	17
4.3 Umgebung	18
4.3.1 Menü	18
4.3.2 Terrain	19
4.3.3 Tag und Nacht Zyklus	21
4.3.4 Wasser	24
4.4 Einfügen der Modelle	25
4.4.1 Häufig platzierte Modelle	25
4.4.2 Kollisionserkennung mit Collider	26
4.4.3 Weitere Lichtquellen	27
4.4.4 Animationen	28
5 Player Controller	29
5.1 Bewegen des Spielers	29
5.2 Kamerabewegung	30
6 Fazit	31
Literatur	32

Abbildungsverzeichnis

3.1	Brücke	4
3.2	Klippe	4
3.3	Lampen	5
6figure.caption.6		
3.5	Japanisches Ramenrestaurant in Blender	6
3.6	Steg	7
3.7	Steine	8
3.8	Tempel nach Vorbild Seiganto-ji-Tempel	8
3.9	Torii	9
3.10	Koi	10
3.11	Koi Textur	11
3.12	Koi Rigidbody	11
3.13	Ahorn	12
3.14	Bonsai	13
3.15	Bambus	13
3.16	Quittenbusch	14
3.17	Farn	15
3.18	Japanischer Kirschbaum	15
3.19	Topfpflanze	16
3.20	Seerosenblätter	16
4.1	Japanischer Garten Skizze	17
4.2	Japanischer Garten Übersicht	18
4.3	Menüführung innerhalb der Szene	19
4.4	Übersicht über die Texturen des Terrains	20
4.5	Übersicht über die Details des Terrains	20
4.6	Tree Tool mit zuvor hinzugefügten Objekten	21
4.7	Umsetzung von Tag und Dämmerung als Skybox	22
4.8	Um die Szene rotierendes Directional Light für die Simulation einer Sonne	22
4.9	Objekt Tree des DL-Objekts, dem die Sonnen-Sphere untergeordnet ist . .	23
4.10	In-Engine Objektzuweisungen durch Parameter des DayCycle.cs Skripts für das Managen der Tag und Nacht Zyklen	23
4.11	Vertiefung im Terrain, bei der die Wasserebene zum Vorschein kommt. . .	24
4.12	Graph des Wasser Shaders. Ein Controller (rechts) verwaltet die Teile Reflection, Color, Displacement, Depth und Normals.	24
4.13	Aus Blender importiertes <i>Bush</i> Objekt und seine Bestandteile	25

4.14 LOD des <i>Bush</i> Objekts. Im Vordergrund werden die Büsche vollständig gerendert, während im Hintergrund nur einige Teile des Busches nach der LOD Vorgabe gerendert werden.	26
4.15 Die in dem Projekt verwendeten Collider: Box Collider (Quader) und Mesh Collider (angepasst auf das Modell)	27
4.16 Globale Lightmap (links) und statisches Lichtobjekt einer Lampe (rechts) . .	27
4.17 Ein Animationscontroller (links) hat einen Idle State, der einen vordefinierten Animationsclip (rechts) beinhaltet.	28
5.1 Spielerobjekt mit Kameraobjekt an der Startposition	29

1 Einleitung

Japanische Gärten sind ein Ort der Kultur und der Natur. Sie dienen dem Ausdruck der japanischen Geschichte und Philosophie. Durch asymmetrische Anordnungen gibt es für den Besucher viel zu entdecken, da sich der Eindruck der Anlage aus verschiedenen Blickwinkeln verändert. Durch dies wird die Achtsamkeit der Besucher geweckt und ein Ort der Erholung und der Ruhe geschaffen. Dabei steht der Symbolismus der Anlage im Vordergrund. So hat jedes Element eine tiefere Bedeutung in der japanischen Philosophie. Um dieses Stück japanisches Gedankengut zu würdigen, wird eine 3D-Szene eines solchen Gartens erstellt. Dies soll ermöglichen, dass jede Person die Möglichkeit hat, diesen Ort der Erholung ortsunabhängig zu genießen. Dabei kann so für wenige Minuten dem Alltag entsagt werden und der Schönheit der japanischen Kultur gehuldigt werden.

Aufbau der Arbeit

In dem hier vorliegenden Dokument werden die Ergebnisse des Projektes beschrieben. Dabei gliedert sich diese Arbeit in fünf Kapitel. Beginnend mit dieser Einleitung wird im folgenden Kapitel die Grundidee der Szene erläutert. Kapitel drei umfasst das Designkonzept, sowie die in Blender erstellten Modelle. Im vierten Kapitel wird die Erstellung der Szene in Unity näher beleuchtet, wobei ebenfalls auf Animationen und Anordnung eingegangen wird. Daraufhin wird im fünften Kapitel der Spieler behandelt. Abschließend werden im Fazit die Ergebnisse der Arbeit zusammengefasst und ein Ausblick gegeben.

2 Grundidee der Szene

Die Grundidee basiert auf traditionell angelegten japanischen Gärten. Dabei wird in drei Hauptgruppen unterschieden, welche im Folgenden umrissen werden.

- **Trocken- und Steingärten (karesansui teien):** Zeichnen sich durch Kieselsteine, Moos und größere Steine aus
- **Teichgärten (chisen teien):** Viele Wasser- und Pflanzelemente, wenig Stein
- **Moosgärten (koke no niwa):** hauptsächlich Moos, wenig Stein- und Wasserelemente (Christian Kaden, 2021)

Ebenfalls gibt es verschiedene Nebentypen, welche jedoch in dieser Szene keine Anwendung finden (Christian Kaden, 2021). Weiterhin wird zwischen den Betrachtungsformen des Gartens unterschieden.

- **Garten betrachten (Kansho Shiki):** Es erfolgt keine Interaktion mit der Gartenanlage, reine Betrachtung von vordefiniertem Standpunkt
- **Durch den Garten spazieren (Kaiyu Shiki):** zu deutsch Wandelgärten, es wird durch den Garten spaziert
- **Vom Boot aus betrachten (Shuyu Shiki):** Betrachtung des Gartens vom Boot aus (Christian Kaden, 2021)

In der Szene wurde ein Teichgarten erstellt, da die Vegetation und Wasserelemente im Vordergrund stehen sollen. Hiermit lässt sich die Szene in die oben beschriebenen Kategorien einordnen. Des Weiteren wurde die Betrachtungsweise Kaiyu Shiki gewählt, da hierdurch eine größtmögliche Interaktion mit dem Garten gewährleistet werden kann. Der Spieler kann sich frei durch die Gartenanlage bewegen und seinen Blickwinkel auf den Garten selbst bestimmen.

Weiterhin sind japanische Gärten meist symbolisch zu verstehen, was für den Aufbau der Szene relevant ist. Steine stehen in der herkömmlichen Interpretation für Tiere als Teil der Natur. Wasser dagegen symbolisiert Seen und Ozeane. Bambus steht für die Verbindung aus Biegsamkeit und die Standhaftigkeit. Ebenfalls schaffen Brücken und Stege neue Perspektiven und Felsen repräsentieren Landschaften. Das Motiv der Vergänglichkeit in der Natur erinnert an die eigene Vergänglichkeit und regen zum Nachdenken an.(Japanischer Garten, 2017)

Diese Elemente finden sich in der Szene wieder, um die Bedeutung von traditionellen japanischen Gärten widerzuspiegeln. Der Garten dient der Darstellung der japanischen Geschichte in diesem Bereich und lädt zur Achtsamkeit und Selbstreflektion ein.

3 Modellierung mit Blender

Die Software Blender ist eine freie Open Source Software für die Erstellung von 3D-Modellen. Sie unterliegt der GNU General Public Licence. Diese erlaubt das ausführen, verändern, studieren und verbreiten der Software. (Blender, 2021) Im Projekt wird die Software zur Erstellung der einzelnen Elemente der Szene verwendet. Dafür werden sie in Blender modelliert und texturiert. Weiterhin werden Einzelemente ebenfalls animiert, wenn es sich um eine Animation innerhalb des Objekts handelt.

3.1 Gestaltung der Objekte

Für das Designkonzept wurde ein Low Poly Ansatz gewählt. Dabei werden aus grafischen Primitiven simplifiziert Modelle erstellt. Dieser Stil zeichnet sich durch eine Modellierungstechnik aus, die eine möglichst geringe Anzahl an Polygonen zur Erstellung von Objekten verwendet. Die Ästhetik zielt auf eine hohe Abstraktion ab. Die Objekte bestehen aus einer reduzierten Anzahl an Vertices und Faces gegenüber High Poly. (Domestika, 2021)

Dieser Ansatz wurde gewählt, um trotz der reduzierten Komplexität der Objekte ein ästhetisches Gesamtbild zu schaffen. Der Anspruch dieser Arbeit ist es nicht, eine realitätsnahe Abbildung der Objekte zu schaffen, sondern diese in einer künstlerischen und abstrahierten Szene darzustellen.

Die Texturen wurden vorwiegend einfarbig gewählt, um dem Designkonzept Low Poly zu entsprechen. Dabei wurden ebenfalls verschiedene Modifikationen zur Veränderung der Oberfläche angewandt. Diese Anpassungen erfolgten über die Software Blender, die Einstellungen in Form von Roughness (Rauheit), Spectacular (Farbkraftigkeit) und Metallic (Metallisch) erlaubt. Hierdurch konnten die verschiedenen Details der Objekte durch unterschiedliche Oberflächen hervorgehoben werden.

Ebenfalls wurde als Akzentfarbe rot gewählt, um Details der verschiedenen Objekte hervorzuheben und eine stimmige Gesamtästhetik zu schaffen. Um den Charakter eines Gartens hervorzuheben, wird als Basisfarbe grün gewählt. Dabei werden jedoch anorganische Objekte wie Gebäude und Brücken bewusst durch eine andere Kolorierung hervorgehoben. Hierdurch erfolgt die Unterteilung in Garten (grün, braun, wenig rot) und Gebäude (mehrfarbig, rot, weiß).

3.2 Modelle

Das Kapitel Modelle umfasst alle Blendermodelle, welche eigenständig erstellt wurden. Dabei werden die im vorherigen Kapitel beschriebenen Prinzipien der Gestaltung der Objekte auf die Modelle angewandt. Das Kapitel wird in übergeordnete Gruppen untergliedert, wie Gebäude, Vegetation etc. um die Übersichtlichkeit zu wahren.

3.2.1 Gebäude und Designelemente

Das Unterkapitel umfasst sowohl die erstellten Gebäude, als auch künstliche Objekte, wie Torbögen, Brücken und Lampen.

Brücke

Um die Möglichkeit zu geben mit Pfaden über Wasserbereiche zu gelangen wurde eine Brücke modelliert, welche in der Abbildung 3.1 zu sehen ist.



Abb. 3.1: Brücke

In japanischen Gärten wird häufig die Farbe rot verwendet. Somit wurde dem hölzernen Untergrund ein rotes Geländer hinzugefügt. Dem hölzernen Weg würde des weiteren eine Wölbung hinzugefügt um eine Erhöhung über dem Wasser zu erzeugen.

Klippe

Um die Landschaft interessanter zu gestalten, wurde eine Klippe erstellt. Diese ist in Abbildung 3.2 zu sehen.



Abb. 3.2: Klippe

Dabei wurde zum eine grüne Textur gewählt, die das Gras darstellt. Für das Gestein wurden zwei graue Texturen erstellt, um die Unregelmäßigkeiten in Steinen darzustellen.

Lampen

Für die Beleuchtung der Szene wurden zwei Lampen gestaltet. Eine hohe Laterne und eine niedrigere Lampe. Beide sind in Abbildung 3.3 zu sehen.



Abb. 3.3: Lampen

Die große Laterne dient der allgemeinen Beleuchtung der Wege. Um diesem Zweck gerecht zu werden, wird eine hölzerne Halterung gewählt. Hierdurch passt sich das Element an die Natur an und sticht nicht heraus.

Die kleine, kunstvoll gestaltete Lampe dient einem dekorativen Zweck und wurde deswegen in der Akzentfarbe rot eingefärbt. Beide Texturen der Halterungen wurde eine hohe Roughness zugewiesen, wodurch das Objekt nicht spiegelt. Beiden Lampen haben ebenfalls eine metallische Lampenhalterung, welche mittels der Einstellung metallic in Blender umgesetzt wurde.

Ramenrestaurant

Ramen ist in Japan ein weit verbreitetes Gericht, weshalb ebenfalls ein Ramenrestaurant erstellt wurde. Dafür wurden Referenzbilder von traditionellen Ramenrestaurants verwendet, um die Szenerie einheitlich zu halten. In Abbildung 3.4 ist ein solches Restaurant dargestellt.



Abb. 3.4: Japanisches Ramenrestaurant¹

Ramen hat in der Geschichte Japans einen bedeutenden Teil. Was zuerst als Gericht für Arme galt, entwickelte sich ab den 1990ern zu einem Gourmetgericht in Japan. (Kizuki, 2019) Durch das Restaurant soll das Ambiente eines angelegten Besucherarks entstehen. Ebenfalls soll dieser Teil der japanischen Küche durch das Ramenrestaurantmodell repräsentiert werden. Hierfür wurde aus grafischen Primitiven ein Low Poly Modell entwickelt. Dieses ist in Abbildung 3.5 zu sehen.



Abb. 3.5: Japanisches Ramenrestaurant in Blender

Dabei setzt sich das Modell aus mehreren Hauptbestandteilen zusammen. Zum einen wur-

¹Quelle: „<https://i.pinimg.com/736x/70/31/b0/7031b01ca183dead48a992b85c330c96-ramen-shop-restaurant-design.jpg>“

de das Dach mit Wellblecherweiterungen erstellt. Des Weiteren wurden vier Holzstreben hinzugefügt, welche dieses stützen. An den Tresen wurden einzelne Holzlatte angebracht, wodurch dieser ebenfalls hölzern wirkt. Die Gestaltung der Küche setzt sich aus einer Dunstabzugshaube, einem Herd und einem Regal mit Schüsseln zusammen. Vor den Hölzern wurden Stühle platziert. Ebenfalls wurden fertige Ramengerichte mit Nudeln, Ei und Essstäbchen auf den Tresen platziert. Weiterhin wurden Fähnchen an das Dach befestigt, da diese ebenfalls häufig in Ramenrestaurants zu finden sind. Weiterhin wurde ein Schild neben den Imbiss platziert. Als Lichtquellen wurden zwei Lampions ergänzt.

Für die Texturen wurden als Primärfarben braun, rot und blau verwendet. Dabei dient rot der Hervorhebung einzelner Elemente, wie der Schüsseln auf dem Tisch, der Lampions und der Sitzflächen des Stuhls. Den Sitzflächen wurde ebenfalls durch das Reduzieren der Roughness innerhalb des Materials eine glänzende Optik hinzugefügt. Braun wurde für die hölzernen Elemente verwendet, wobei für große Objekte, wie die Holzplanken ein helleres Braun verwendet wurde als für die Streben. Hierdurch schließen die Streben das Modell durch die dunklere Farbe ab. Für die Holzmaterialien wurde die Roughness reduziert, da das Holz natürlich wirken soll. Für die Fähnchen wurde ein dunkles Blau verwendet, um hier ebenfalls den Abschluss des Modells zu verdeutlichen. Die metallischen Objekte wurden mithilfe eines hellen Silbertons und der Materialeinstellung metallic erstellt. Dieses Material wurde auf die Streben der Stühle, das Wellblechdach und den Herd verwendet. Durch die Veränderung der Texturen der verschiedenen Objekte wird eine realistischere Optik geschaffen.

Steg

Der modellierte Steg stellt einen Weg auf das Wasser dar. Durch die Zylinder Objekte werden Hölzer dargestellt auf denen gelaufen werden kann. Durch die braune Farbe sollen die Objekte Holz darstellen.

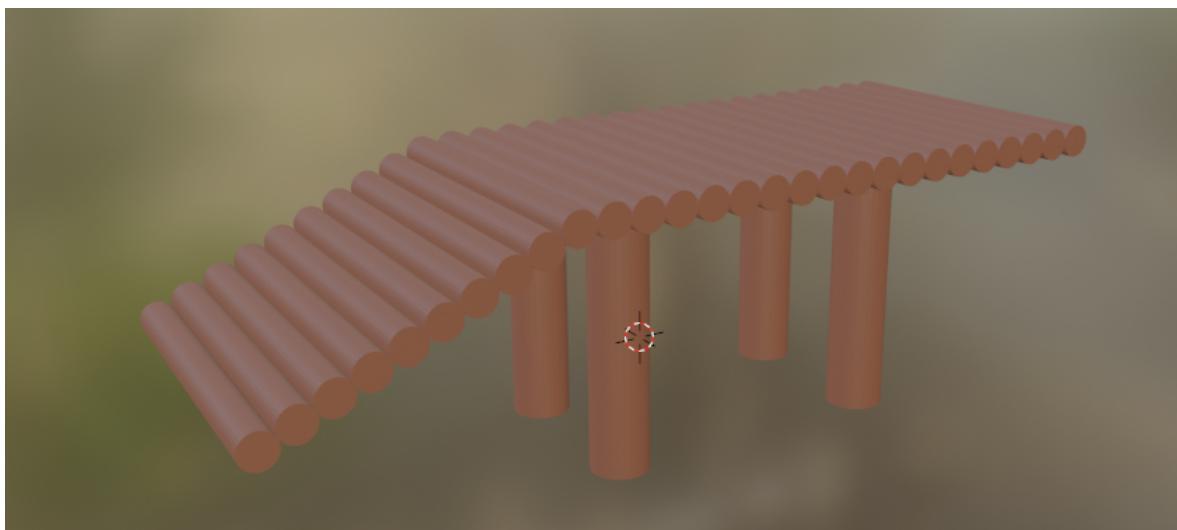


Abb. 3.6: Steg

Steine

Da in Teichgärten ebenfalls Steinelemente zu finden sind, wurden verschiedene Gesteine modelliert. Dabei wurde zum einen ein einzelner Stein, zum anderen ein Steinhaufen erstellt. Hierdurch kann Abwechslung in der Gestaltung der Landschaft erreicht werden. Die verschiedenen Modelle werden in Abbildung 3.7 dargestellt.

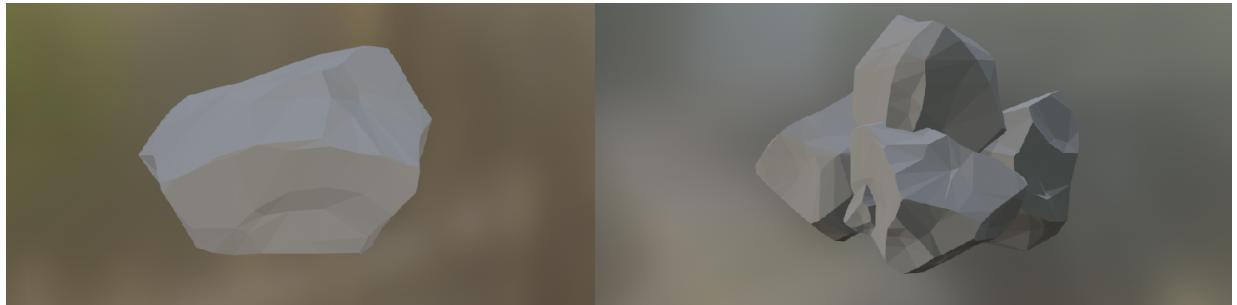


Abb. 3.7: Steine

Um eine natürliche und randomisierte Struktur der Steine zu ermöglichen, wurde auf ein Oval verschiedene Modifier angewandt. Mit dem Displace Modifier aus Blender wurde die Oberflächenstruktur erstellt und im Folgenden wurde die Anzahl der Faces mittels des Decimate Modifiers reduziert. Als letztes wurden die Steine grau eingefärbt.

Tempel

Der Shintoismus ist einer der führenden Religionen Japans. Zu Ehren der Götter werden Tempelanlagen errichtet, welche als Ort der Religion dienen. (Gaël, 2021)

Nach Beispiel des Seiganto-ji-Tempels, wurde ein japanischer Tempel modelliert. Dieses Modell ist in Abbildung 3.8 zu sehen.



Abb. 3.8: Tempel nach Vorbild Seiganto-ji-Tempel

Dabei wurde der Tempel ausschließlich aus geometrischen Primitiven erstellt. Es wurden keine Modifier angewandt.

Für die Textur wurden die verschiedenen Faces nach Vorbild des Referenztempels eingefärbt. Die Akzentfarbe ist hier ebenfalls wie zuvor beschrieben die Farbe rot. Diese steht in Japan für eine heilige Atmosphäre und Lebenskraft (Miho Shimizu, 2021). Des Weiteren wurde die Textur je nach Oberfläche des Faces abgewandelt. Dabei wurden die Wände eher rau gewählt und die Dächer spiegelnd.

Torii

Das Torii ist Teil japanischer Geschichte. Ein Torii Tor steht zum einen für das Trennen von Tempelgrund an Shinto Schreinen als heiliger Ort und der menschlichen Welt. Zum anderen können diese ebenfalls in der Natur aufgefunden werden, wobei Torii in diesem Umfeld Gottheiten verkörpert. Auffällig ist bei Torii die typische japanische Bauart und die meist rote Farbe. Wie bereits bei dem Unterpunkt Tempel erwähnt, steht die Farbe für eine heilige Atmosphäre und Lebenskraft. (Miho Shimizu, 2021) Ein solches Tor wurde ebenfalls für die Szene erstellt und wird in Abbildung 3.9 dargestellt.



Abb. 3.9: Torii

Inspiration waren hierbei die traditionellen Torii-Tore Japans. Um das Holz zu repräsentieren wurden die Balken des Tors uneben gestaltet.

3.2.2 Tiere

Tiere und tierähnliche Wesen haben eine Tradition in der japanischen Kultur. Sie können sowohl als göttliche Boten oder freundliche Helfer, aber auch als listiger Trickser gelten. (Diana Casanova, 2021)

Koi Fisch

Koi Fische stehen für verschiedene Symbole, dazu zählen Werte wie:

- Erfolg
- Ehrgeiz
- Glück
- Reichtum
- Stärke

Koi Fische wurden lediglich als Speisefische gezüchtet und waren ursprünglich nur einfarbig. Erst durch ein paar Mutationen wurde es möglich Kois in verschiedenen Farben zu züchten.

Kois können lange Stecken und Höhenunterschiede überwinden und sind somit eine sehr kraftvolle Fischart. (Asian Lifestyle, 2020)

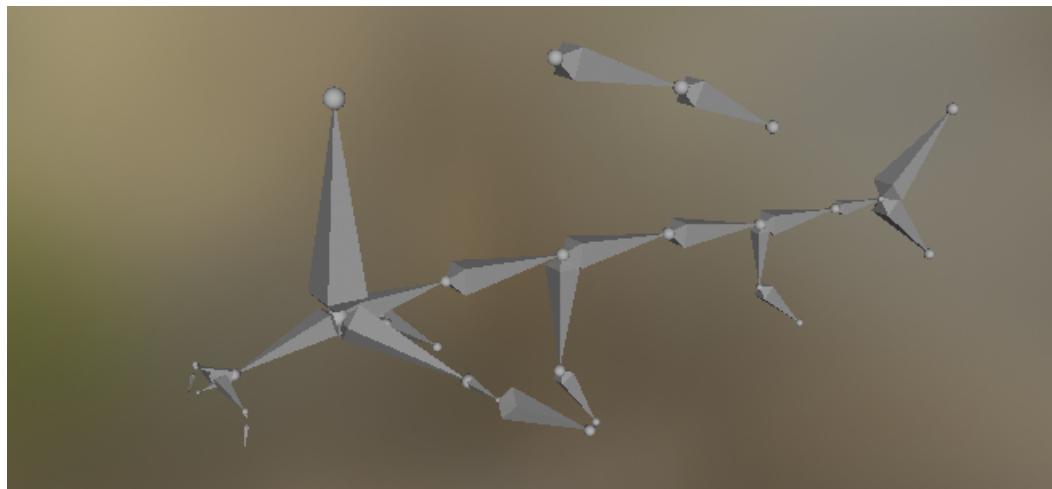


Abb. 3.10: Koi

Der Koi wurde modelliert in dem immer wieder verschiedene Seiten des ursprünglichen Würfels verlängert und neu skaliert werden. Des Weiteren wurde der Koi an der Y-Achse gespiegelt um eine gleichbleibende Symmetrie zu erzeugen. Bei der Textur des Kois handelt es sich um eine selbst erstellte Textur. Hierbei wird das Modell aufgeklappt um jede Fläche bemalen zu können.

**Abb. 3.11:** Koi Textur

Um dem Koi Fisch Leben zu verleihen, wurde diesem eine Bewegungsanimation hinzugefügt. Diese wird erzeugt, wenn dem Objekt ein sogenannter Rigidbody hinzugefügt wird. Dieser Rigidbody stellt für das Modell Knochen dar. Den Knochen wird das Objekt mit verschiedenen Gewichtungen zugewiesen. Somit ist es möglich, den Fisch über verschiedene Gelenke zu bewegen.

**Abb. 3.12:** Koi Rigidbody

Daraufhin kann eine Animation erstellt werden. Hierzu werden sogenannte Keyframes gesetzt. An diesen Keyframes wird die aktuelle Position und Rotation zu einem bestimmten Zeitpunkt gemessen. An einem zweiten Keyframe werden neue Rotationen und Positionen gespeichert. Somit wird zwischen den Keyframes in linearer Geschwindigkeit die Veränderung der Werte ausgeführt, wodurch man eine Animation bekommt.

3.2.3 Vegetation

Dieses Kapitel umfasst die Vegetation, welche in Blender für den japanischen Garten erstellt wurde. Dies umfasst die Gesamtheit der Pflanzen, die in die Szene eingefügt wurden.

Typische Pflanzen in japanischen Gärten sind beispielsweise der Ahorn, der Efeu, der Zwergbambus und die Schwertlilie (Garten-Blog, o.J.).

Ahorn

Eine typische Baumart in Japan ist der japanische Ahorn. Dieser zeichnet sich durch die rote Färbung der Blätter, sowie einen kurzen Stamm aus. Das Modell in Abbildung 3.13 stellt diesen Wachstum nach.



Abb. 3.13: Ahorn

Dabei wurde das Modell nach Referenzbildern japanischer Ahorns modelliert.

Bonsai

„Das Wort ‘Bon-Sai‘ ist japanischer Herkunft und heißt wörtlich übersetzt ‘Baum in der Schale‘. Diese Kunstform ist aus der alten chinesischen Gartenpraxis entstanden und in Teilen unter dem Einfluss des japanischen Zen-Buddhismus weiterentwickelt worden“ (Bonsai Empire, o.J.).

Bei dieser Kunstform werden Bäume mittels verschiedener Techniken kunstvoll zugeschnitten. Für das erstellte Modell wurde die Größenordnung Imperiale gewählt, welche 152 - 203 cm groß sind. (Bonsai Empire, o.J.)

Das Modell ist in Abbildung 3.14 zu sehen.



Abb. 3.14: Bonsai

Für die Modellierung wurden verschiedene Bonsaipflanzen als Vorbild genommen, um an die traditionelle Kunstform Japans zu erinnern.

Bambus

Der Bambus gehört zu den Süßgräsern und zeichnet sich durch hohle oder vollmarkige Halme aus. Die Äste sind rispig. Das Gewächs ist in allen Kontinenten bis auf Europa und die Antarktis beheimatet und ist ein typisches Gewächs für asiatische Länder. Ebenfalls ist es häufig in japanischen Gärten aufzufinden. Das Modell ist in Abbildung 3.15 zu sehen.



Abb. 3.15: Bambus

Im Modell wurde die typische Vegetationsweise erstellt, indem der Bambus zum einen in Stauden wächst. Zum anderen wurden die vollmarkigen Halme durch einzelne geometrische Primitive als Zylinder umgesetzt. Ebenfalls wurden verschiedene Grüntöne zur

Texturierung verwendet, wodurch ein Kontrast innerhalb des Modells geschaffen wird.

Quitten Busch

Der Quittenbusch ist eine japanische Zierpflanze, welche zu den Rosengewächsen. Der Busch hat intensiv rote Blüten.

In der Abbildung 3.16 ist das hierzu erstellte Modell zu sehen. Dabei wurden die roten Blüten mittels roter Kreise dargestellt.

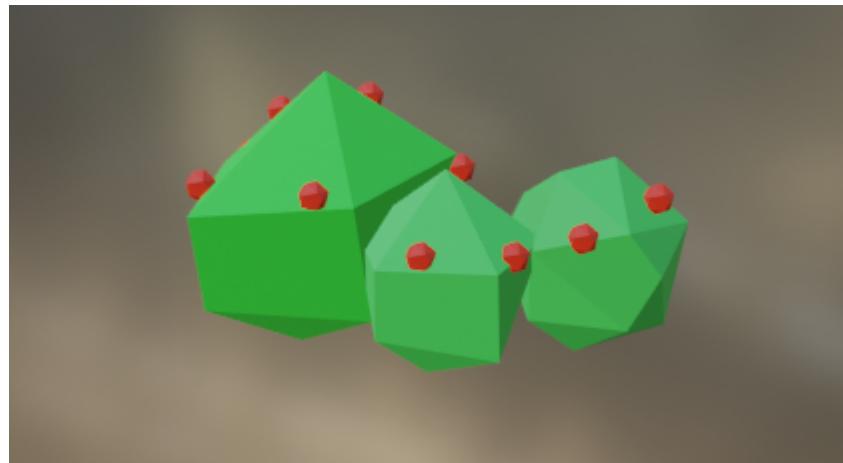


Abb. 3.16: Quittenbusch

Farn

Farne sind in Japan ebenfalls heimisch, weshalb eine solches Modell für die Szene entwickelt wurde. Dabei dient der Farn als Ergänzung der Szene, ist also kein Schlüsselement. Es wurden zwei dunkle Grüntöne verwendet, um diesen Zweck zu erfüllen. Das Modell ist in Abbildung 3.17 zu sehen.



Abb. 3.17: Farn

Kirschbaum

Ebenfalls ist der japanische Kirschbaum ein typisches Gewächs Japans. Das hierfür erstellte Modell ist in Abbildung 3.18 zu sehen.

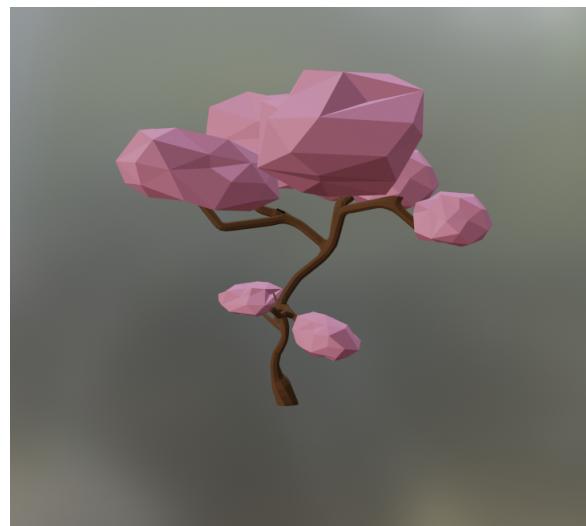


Abb. 3.18: Japanischer Kirschbaum

Dabei wurde auf die typische rosa Farbe der Kirschblüten geachtet. Ebenfalls wurde mittels Referenzbildern der Wachstum des Stamms nachgestellt.

Topfpflanze

Das Modell Topfpflanze ist ein fiktives Gewächs, wobei keine Referenz genommen wurde. Das Modell ist in Abbildung 3.19 zu sehen.



Abb. 3.19: Topfpflanze

Es dient als Fülllement in der Landschaft, weshalb die Farben grün und braun gewählt wurden. Hierdurch sticht das Modell nicht heraus und fällt nur dem genauen Betrachter auf.

Seerosen

Ebenfalls wurden Seerosenblätter zur Verschönerung der Wasseroberfläche erstellt. Hierfür wurde die Form eines Seerosenbattes in Blender ausgeschnitten und im Folgenden die Höhe angepasst. Abschließend wurde das Blatt dupliziert und grün eingefärbt. Das Modell ist in Abbildung 3.20 zu sehen.

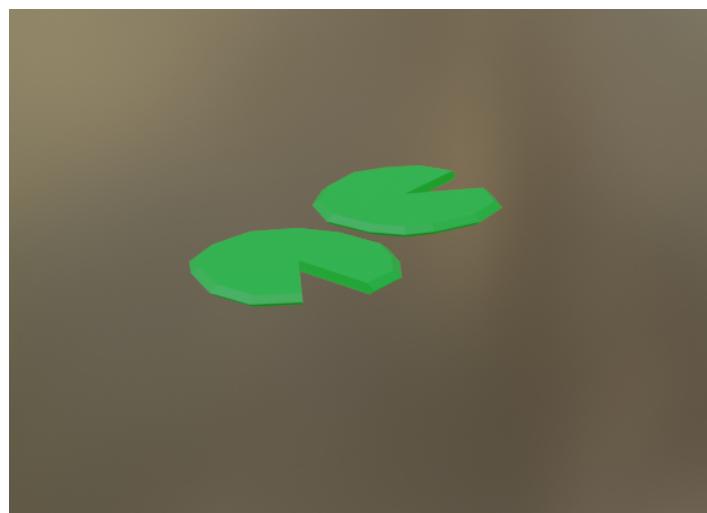


Abb. 3.20: Seerosenblätter

4 Erstellung der Szene

Die Szene wurde mithilfe der Unity Engine zusammengestellt. Die Objekte wurden nach einem zuvor definierten Grundriss angeordnet und anschließend durch Hinzufügen von Terrain, einer Sonne und Wasser abgerundet.

4.1 Unity Grundlagen

Unity ist eine populäre Echtzeit Entwicklungsplattform aus der Gaming Branche, die für non-profit Zwecke völlig kostenlos verwendet werden kann. Zusammen mit einer großen Community bildet die Engine ein robustes Ökosystem zur Entwicklung jeglicher 2D und 3D Applikationen. (Unity Technologies, 2021)

Unity wird für dieses Projekt verwendet, da einige Grundlagen der Computergrafik bereits 'out of the box' umgesetzt werden und der Fokus somit auf der Szene selbst liegt. Auf diesen mathematischen Grundlagen wird in diesem Projekt aufgebaut, um in kurzer Zeit eine beeindruckende Szene zu kreieren.

4.2 Gesamtaufbau der Szene

Der Gesamtaufbau der Szene ist die Anordnung der Objekte. In Abbildung 4.1 ist diese vereinfacht dargestellt.

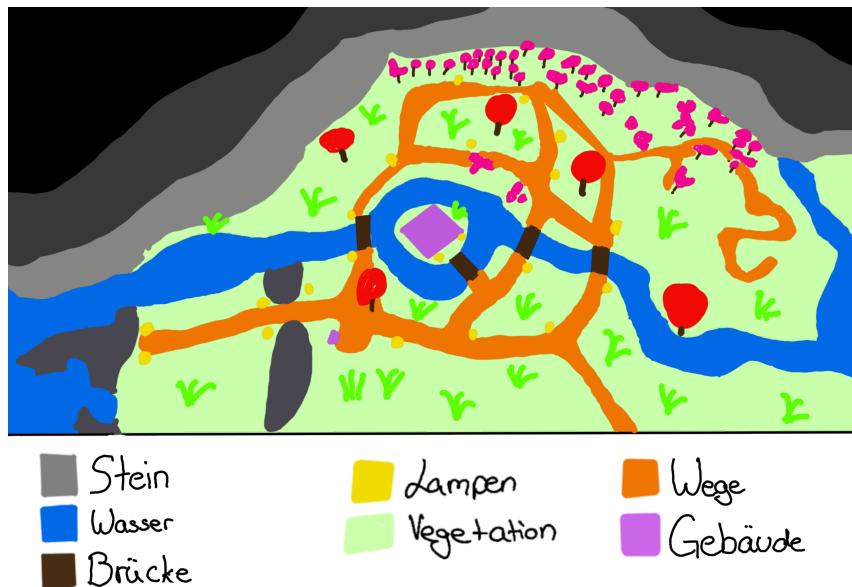


Abb. 4.1: Japanischer Garten Skizze

Dabei sind die verschiedenen Elemente in der Legende beschrieben. Das Zentrum bildet ein Tempel, welcher von Wasser umgeben ist. Die Karte wird mittels Fels eingegrenzt. Die Wege werden in Orange dargestellt, jedoch ist der komplette Bereich vom Spieler begehbar. Ebenfalls ist der Karte der Fokus auf Vegetation und Wasser zu entnehmen,

was einem Teichgarten entspricht.

In Abbildung 4.2 wird die Szene nochmals in Unity dargestellt.



Abb. 4.2: Japanischer Garten Übersicht

4.3 Umgebung

Die Umgebung bildet die Basis für die in Blender erstellten Modelle. Dazu gehören sowohl das Terrain um den Garten herum, als auch Tag und Nacht Zyklen und Wasser. Auch das Menü gehört dazu, da es die Nutzererfahrung positiv abrundet trotz keiner direkten Zugehörigkeit zum japanischen Garten.

4.3.1 Menü

Sobald die Szene geladen wurde lässt sich das Menü mit der ESC-Taste öffnen und schließen. Während das Menü geöffnet ist, wird die Zeit innerhalb der Szene pausiert, d.h. Animationen werden für diesen Zeitraum ebenfalls pausiert. Der Resume-Button dient dazu das Menü wieder zu schließen und in die Szene zurückzukehren. Durch einen Klick auf den Quit-Button kann das Programm geschlossen werden. Der Mousespeed Schiebebe-regler ermöglicht es dem Benutzer die Mausempfindlichkeit individuell einzustellen.



Abb. 4.3: Menüführung innerhalb der Szene

4.3.2 Terrain

Das Terrain wird in Unity erstellt und bildet die Grundlage für den weiteren Aufbau der Szene. Jedes Terrain besteht aus mehreren Komponenten, die getrennt voneinander bearbeitet werden können und gemeinsam das endgültige Terrain formen. Zunächst werden Höhen und Tiefen modelliert um die Höhenunterschiede innerhalb der Landschaft darzustellen. Dazu gehören die Berge, Flüsse, Wege und alle anderen Unebenheiten. Dafür werden in Unity verschiedene Werkzeuge zur Verfügung gestellt. Ein Glätten der Oberfläche ist ebenfalls möglich. Im nächsten Schritt wird mithilfe des Brush Tools das Terrain mit Texturen versehen. Das Terrain besteht aus mehreren Texturschichten, wie Gras, Moos, Dreck, Fels und Stein, die wiederum miteinander kombiniert werden können. In Abbildung 4.4 werden die verwendeten Texturen gezeigt.

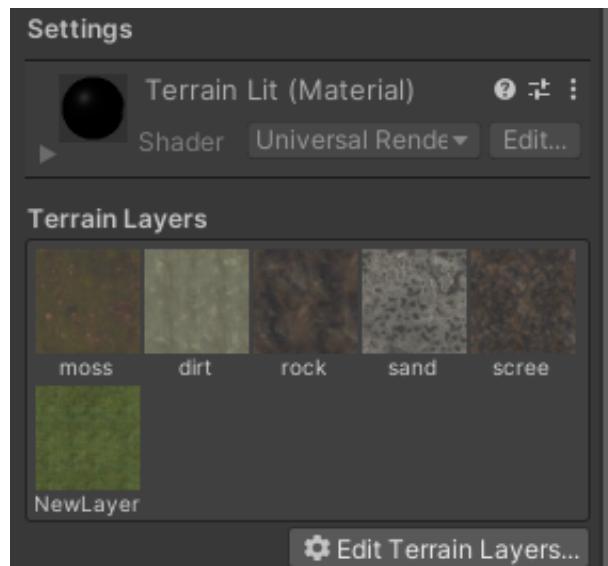


Abb. 4.4: Übersicht über die Texturen des Terrains

Nachdem der Untergrund modelliert wurde und auch die Texturen hinzugefügt wurden, also alle Landschaftsmerkmale, wie Wege oder Flussbetten und Berge geformt wurden, besteht die Möglichkeit Details, wie beispielsweise Gräser, Blumen und Bäume einzuziehen.

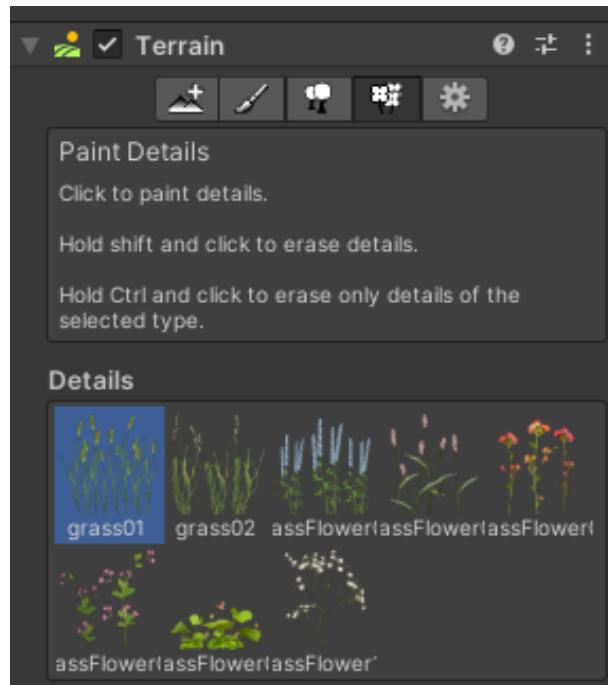


Abb. 4.5: Übersicht über die Details des Terrains

Die *Tree* Funktion ermöglicht das *Zeichnen* von Bäumen oder generell häufig vorkommenden Objekten in großen Massen. Objekte, die mit dem Tree Tool platziert wurden, werden in der Renderpipeline gesondert behandelt, um Rechenleistung zu sparen.

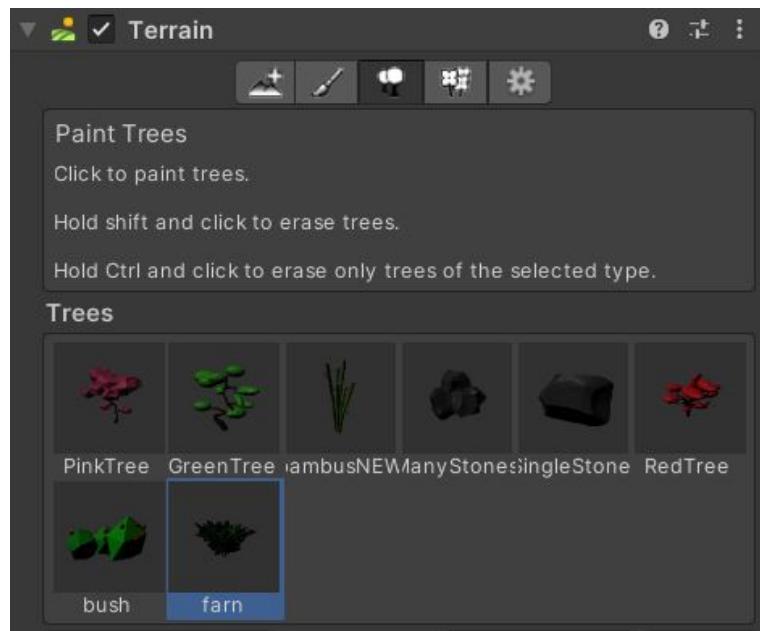


Abb. 4.6: Tree Tool mit zuvor hinzugefügten Objekten

Wenn ein Objekt aber immer wieder kopiert wird, wirkt die Szene oft einseitig und langweilig. Um das vorzubeugen, wird in den Zeicheneinstellungen die Größe und Rotation der Objekte in einem vorgegebenen Intervall zufällig gewählt. Dadurch entsteht eine große Variation desselben Objekttyps ohne großen manuellen Aufwand.

4.3.3 Tag und Nacht Zyklus

Der Tag und Nacht Zyklus bringt ein unauffällig auffälliges Ambiente mit sich und trägt signifikant zum Realismus der Szene bei. Die Grundlage bietet eine selbst erstellte innere Uhr, die je nach Einstellung der Geschwindigkeit eines Zyklus die aktuelle Uhrzeit in Stunden als Variable in *DayCycle.cs* bereitstellt.

Die Skybox besteht aus einem Tag und einem Nacht (hier Dämmerung) Setting, die grundlegend durch zwei verschiedene Materialen definiert sind. Die Dynamik der Skyboxen selbst entsteht durch weitere Skripte und Shader.



Abb. 4.7: Umsetzung von Tag und Dämmerung als Skybox

Zwischen 7 Uhr morgens und 16 Uhr abends wird ein langsamer Wechsel durchgeführt. Der Übergang wird mittels einer linearen Interpolation zwischen den zwei Skybox Materialen durchgeführt und kann in seiner Länge frei definiert werden.

Das Sonnenlicht wird von einem *Directional Light* Objekt ausgestrahlt. Die Position des Directional Lights ist irrelevant, da es außerhalb des Scopes von einer unendlich großen Ebene aus Lichtstrahlen in die Szene emittiert.

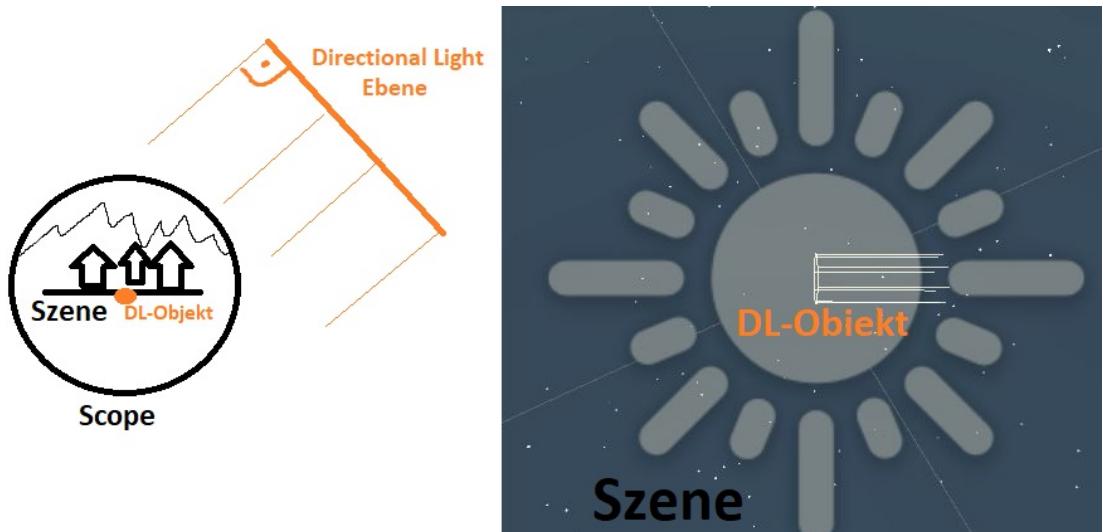


Abb. 4.8: Um die Szene rotierendes Directional Light für die Simulation einer Sonne

Durch die Rotation des DL-Objekts wird die DL-Ebene um das DL-Objekt mit rotiert. Somit kann der Effekt einer Sonne simuliert werden.

Sowohl das DL-Objekt als auch die Ebene haben beide keine Textur und sind transparent. Das heißt, dass aktuell nur die Lichtverhältnisse einer Sonne auf die Szene wirken, aber keine Sonne selbst zu erkennen ist.

Deshalb wird ein funktionsloses Sphärenobjekt mit gelbem lichtdurchlässigen Material

verwendet, um die eigentliche Sonne darzustellen. Die Sphere befindet sich am Rand der Szene, aber noch immer innerhalb des Scopes, und rotiert um die restlichen Szenenobjekte.

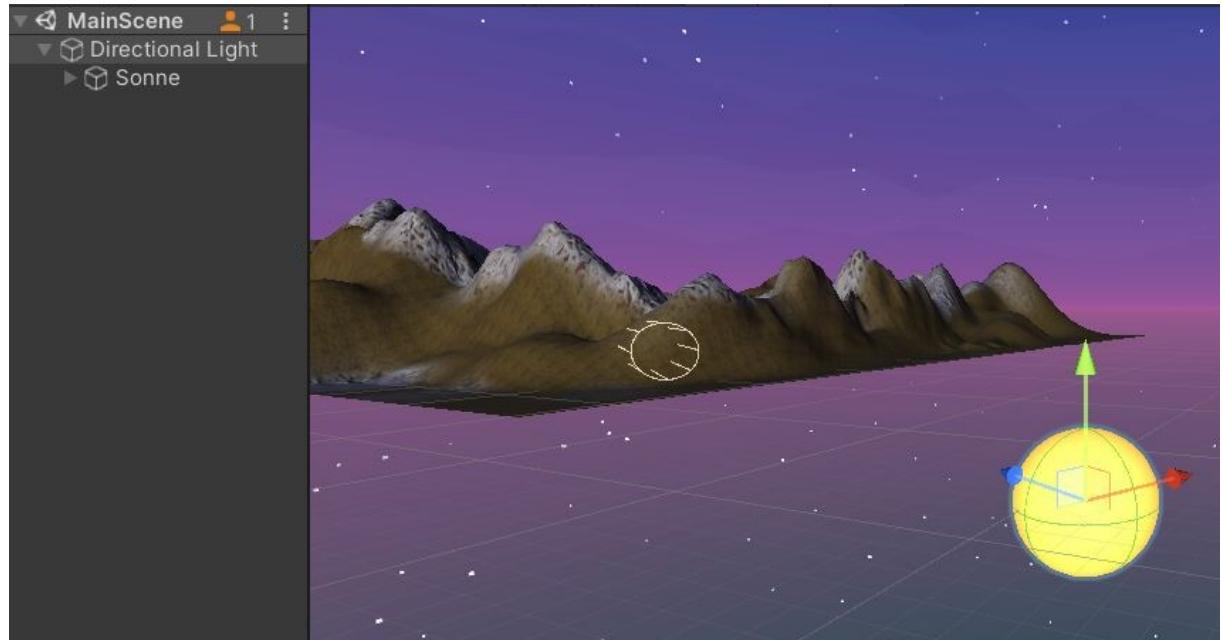


Abb. 4.9: Objekt Tree des DL-Objekts, dem die Sonnen-Sphere untergeordnet ist

Die Rotation muss synchron zu der DL-Ebene sein, weshalb die Sphere in den Objekt Tree des DL-Objekts als untergeordnetes Objekt hinzugefügt wird. Dadurch befindet sich die Sphere im selben lokalen Koordinatensystem wie das DL-Objekt und erbt dieselben Positionsdaten im globalen Koordinatensystem. Wird das Root DL-Objekt nun rotiert, so rotiert die Sphere synchron mit der DL-Ebene um das DL-Objekt mit.

Die Rotation des DL-Objekts wird in dem *DayCycle.cs* Skript relativ zu der Geschwindigkeit der Zykluszeit in Minuten durchgeführt. Diese und andere Parameter bzw. Objektzuweisungen können im Editor selbst dem Skript zugewiesen werden.

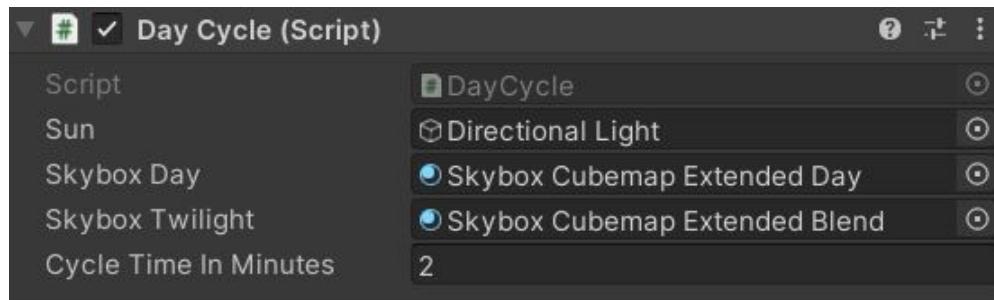


Abb. 4.10: In-Engine Objektzuweisungen durch Parameter des DayCycle.cs Skripts für das Managen der Tag und Nacht Zyklen

Zusammenfassend lässt sich also sagen, dass das *DayCycle.cs* Skript zusammen mit Objekten, Skyboxen und Materialen für einen umfangreichen Tag und Nacht Zyklus sorgt.

4.3.4 Wasser

Das Wasser in der Szene ist eine rechteckige Ebene, auf die ein Shader angewandt wird. Die Ebene liegt nur wenige Höheneinheiten unter dem Gartenboden und erweckt dadurch den Anschein, dass sich in den Flussbetten auch wirklich Wasser befindet.



Abb. 4.11: Vertiefung im Terrain, bei der die Wasserebene zum Vorschein kommt.

Durch den dynamischen Shader nimmt die Ebene die Eigenschaften von Wasser an. Dazu gehören beispielsweise die animierten Wellen, die bläulich-durchsichtige Farbe und realistische Reflexionen.

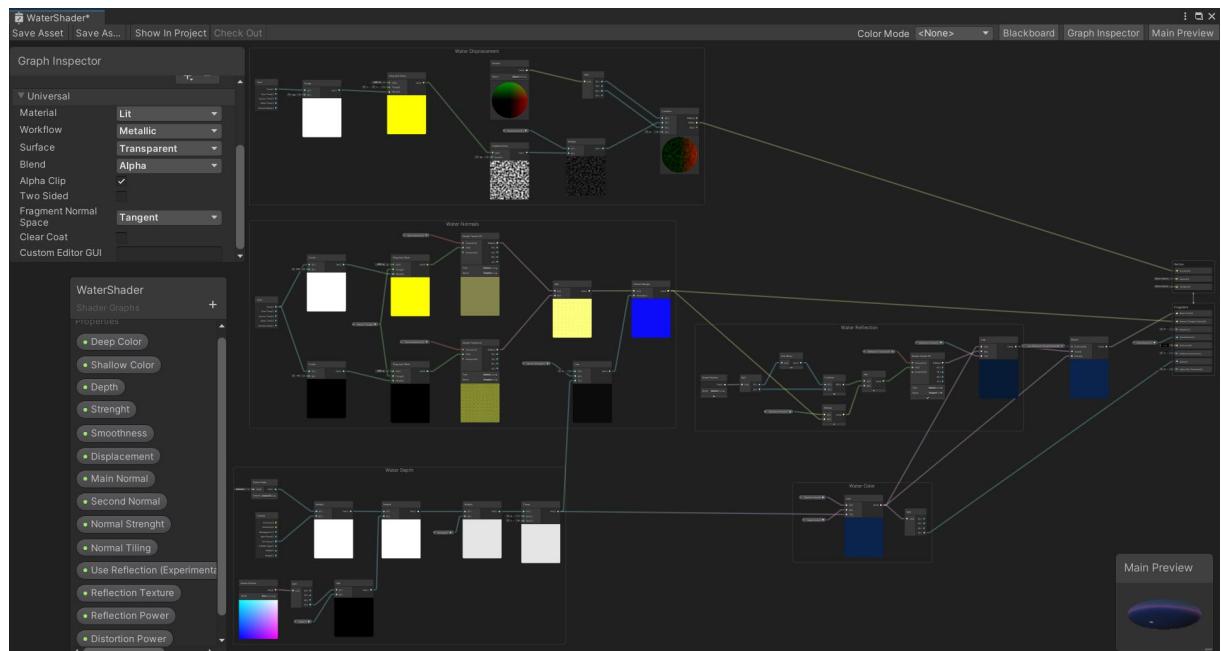


Abb. 4.12: Graph des Wasser Shaders. Ein Controller (rechts) verwaltet die Teile Reflection, Color, Displacement, Depth und Normals.

Der Shader selbst besteht aus einem komplexen Zusammenspiel fünf verschiedener Teile: der Reflektion, die Farbe, die Tiefe, die Normalen und das Displacement. Der Shader-Graph veranschaulicht den Zusammenhang zwischen den verschiedenen Teilen.

4.4 Einfügen der Modelle

Das Einfügen der Modelle ist nicht so trivial, wie es zunächst wirkt. Bereits bei dem Export aus Blender muss einiges beachtet werden, wie die korrekte Positionierung des Modells auf dem Ursprungspunkt und einen übersichtlichen Objektbaum.

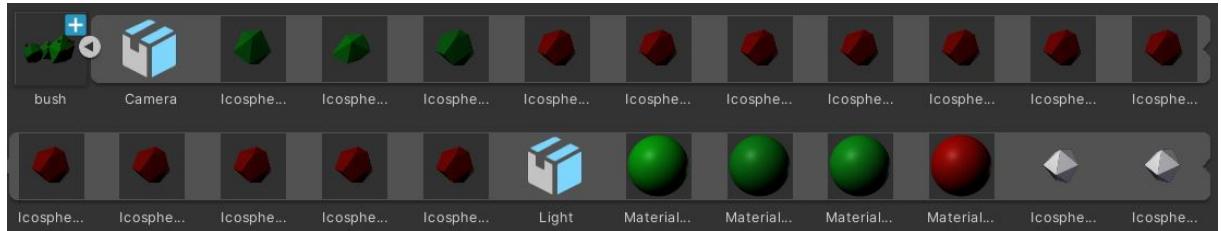


Abb. 4.13: Aus Blender importiertes *Bush* Objekt und seine Bestandteile

In Unity müssen nach dem Import der Objekte einige Anpassungen vorgenommen werden. Dazu gehört die Skalierung, oftmals Änderungen des Materials, das Hinzufügen von Lichtquellen und Collidern sowie das Einrichten von Animationen und Skripten.

4.4.1 Häufig platzierte Modelle

Wird ein Modell mehrmals in der Szene verwendet, dann sollte besonders auf die Komplexität und das Rendering des Modells geachtet werden. Unity bietet einige Funktionen, um viele (komplexe) Modelle desselben Typs in der Szene zu haben, ohne große Performanceeinbrüche zu erleiden.

Eine der wichtigsten Funktionen ist die LOD-Komponente, wobei LOD für *Level of Detail* steht. Durch Sie werden Objekte ab einer gewissen Distanz sofern möglich nicht mehr vollständig gerendert. Welche Teile des Objekts vollständig gerendert werden, kann in Unity manuell festgelegt werden.

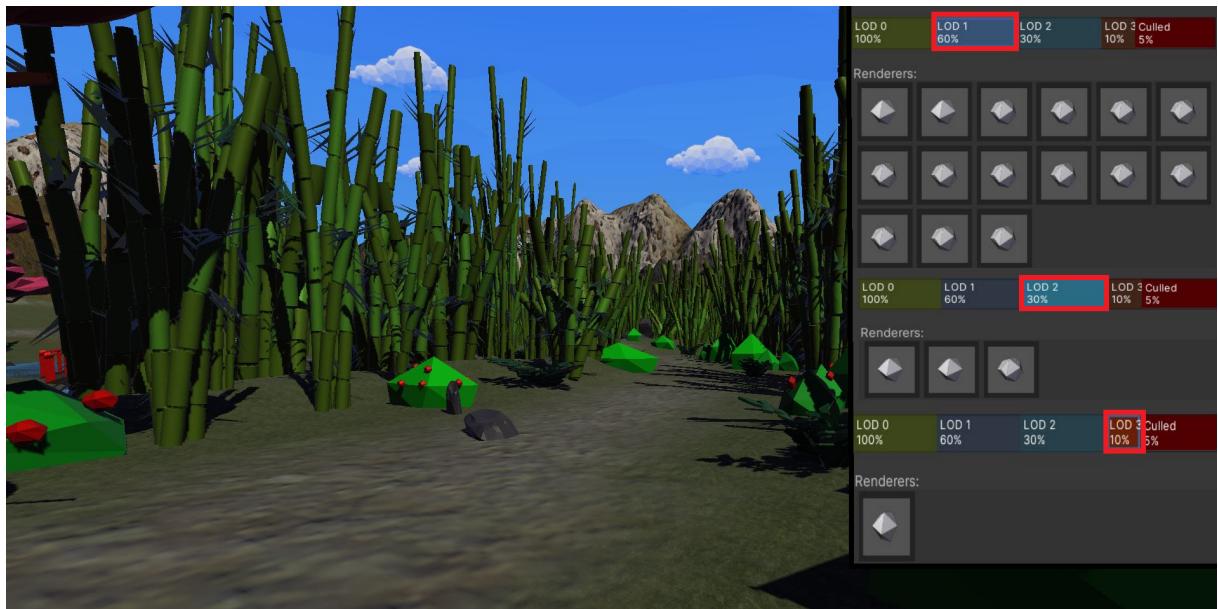


Abb. 4.14: LOD des *Bush* Objekts. Im Vordergrund werden die Büsche vollständig gerendert, während im Hintergrund nur einige Teile des Busches nach der LOD Vorgabe gerendert werden.

Dieses Prinzip wird auf fast jedes Objekt im Projekt angewandt. So werden beispielsweise bei Bambus die Blätter nur in absoluter Nähe gerendert und verschwinden weiter weg, genauso wie die Blattsphären bei Bäumen, die bei zunehmender Entfernung immer weniger werden.

Anschließend werden die Objekte in das zuvor erwähnte Tree Tool hinzugefügt, mit dem man die Objekte auf das Terrain *zeichnen* kann.

4.4.2 Kollisionserkennung mit Collider

In dem Projekt werden zwei verschiedene Arten an Collidern verwendet, um das Nutzererlebnis so angenehm und realistisch wie möglich zu machen.

Collider sorgen dafür, dass Objekte sich nicht ineinander bewegen können. In diesem Fall ist unser sich bewegendes Objekt das Playerobjekt, welches mit einer *RigidBody* Komponente versehen ist. Diese greift auf die Physik der Unity Engine zurück und kann dementsprechend mit Collidern interagieren.

Je nach Objekttyp und auch Häufigkeit ist es sinnvoll, einen eher simplen oder einen komplexen Collider zu verwenden.

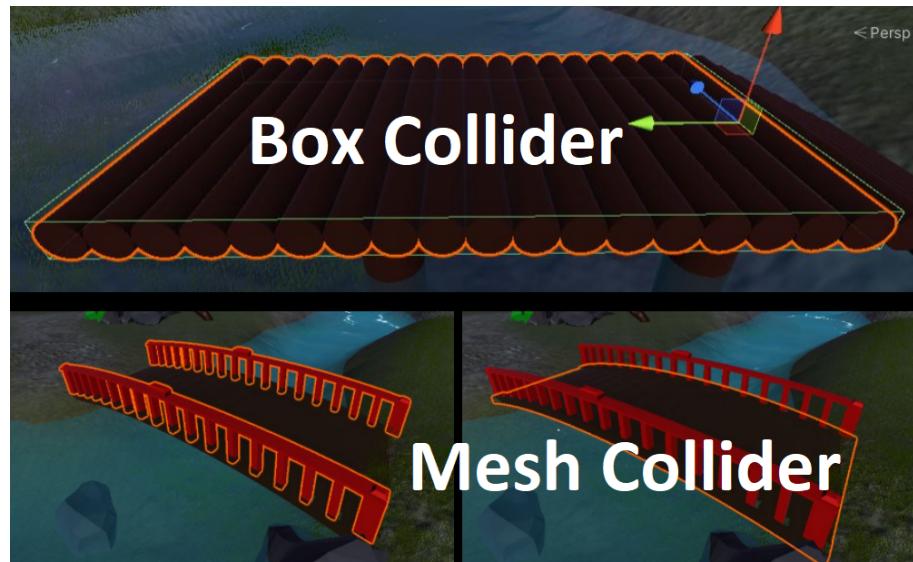


Abb. 4.15: Die in dem Projekt verwendeten Collider: Box Collider (Quader) und Mesh Collider (angepasst auf das Modell)

Ein Box Collider eignet sich für simple und häufig vorkommende Objekte, da dieser lediglich ein Quader ist, der nach Belieben skaliert werden kann. Ein Mesh Collider passt sich genau an die Form des Objekts an, wodurch detaillierte Interaktionen möglich sind. Der Rechenaufwand ist hierfür allerdings deutlich größer, weshalb Mesh Collider in diesem Projekt nur sehr selten benutzt werden.

4.4.3 Weitere Lichtquellen

Neben der Sonne existieren noch weitere Lichtquellen in der Szene: die Lampen. Mit den verschiedenen Lampentypen wird die Szene vor allem bei Nacht lebendig.

Damit die vielen Lichtquellen nicht für eine signifikante Reduktion der Performance sorgen, wird das Licht der statischen Lampen *gebaked*. Das bedeutet, dass vor dem Build der Anwendung eine Lightmap aus gebaketen Objekten erstellt wird.



Abb. 4.16: Globale Lightmap (links) und statisches Lichtobjekt einer Lampe (rechts)

In der Lightmap werden die Berechnungen von zu bakenden Lichtobjekten und deren

Auswirkung auf die Szene festgehalten. Dadurch wird während dem Ausführen der Anwendung keinerlei Lichtberechnung bei gebaketen Lichtern durchgeführt, da dies bereits vorher durchgeführt und in der Lightmap gespeichert wurde.

4.4.4 Animationen

In der Szene befinden sich neben des Tag und Nacht Zyklus auch eine konkretere Animation: die des Fisches.

In Blender wurde für den Fisch ein Animationsclip erstellt, in dem er seine Heckflosse bewegt und Bewegungen mit seinem Mund macht. Der Clip hat eine Dauer von 2.5 Sekunden und endet wieder in der Ausgangsposition des Fisches, wodurch ein nicht auffallender Loop möglich wird.

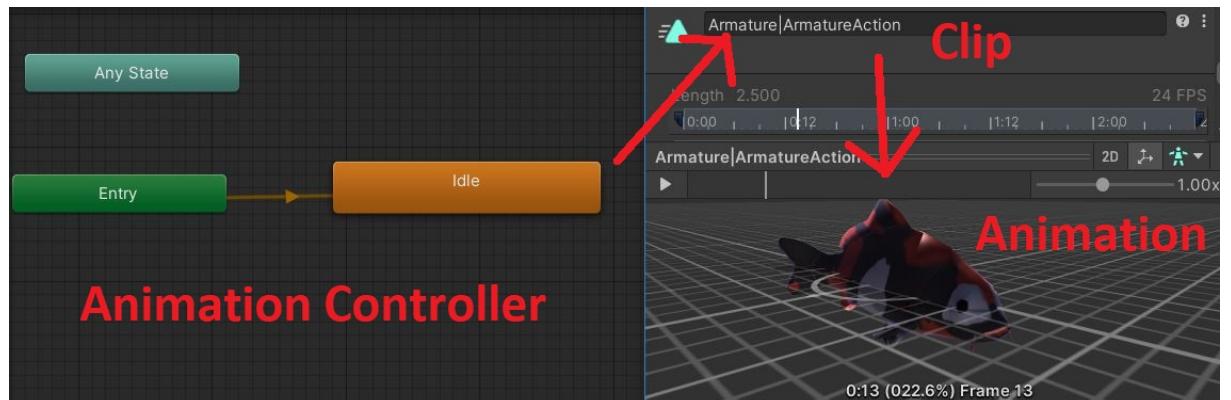


Abb. 4.17: Ein Animationscontroller (links) hat einen Idle State, der einen vordefinierten Animationsclip (rechts) beinhaltet.

In Unity wird der Clip durch einen Animation Controller verwaltet. Dieser kann durch States die Animationen eines Objekts steuern. Da nur ein Clip vorhanden ist, wird nur ein State benötigt, der in diesem Fall *Idle* genannt wird. Der Idle state beinhaltet den Animationsclip. Der Animationsclip selbst besteht aus mehreren Keyframes, die eine zeitliche Abfolge der einzelnen Teile des Fisch- bzw. Avatar-Skelettes repräsentieren.

Der Animation Controller wird anschließend in einer *Animate* Komponente des Fisch-Objekts zusammen mit dem Fisch-Avatar referenziert, um den Fisch zu animieren.

5 Player Controller

Damit ein freies Bewegen innerhalb der Szene möglich ist, wird ein Objekt als Spieler eingesetzt. Die Kamera wird auf dem Spielerobjekt platziert. Im Folgenden wird sowohl das Bewegen des Spielers (laufen) als auch der Kamera (umschauen) beschrieben. Die Funktionen werden mittels Skripte umgesetzt.



Abb. 5.1: Spielerobjekt mit Kameraobjekt an der Startposition

5.1 Bewegen des Spielers

Das Skript *PlayerMove.cs* ermöglicht die Steuerung des Spielerobjektes, das auch die Kamera enthält durch den Nutzer. Für die Umsetzung kommt ein CharacterController zum Einsatz. Die Steuerung erfolgt über die Tasten *WASD*, wie es in vielen Videospielen ebenfalls der Fall ist. Mittels der Funktion *GetAxis()* der Input Klasse lassen sich so die Bewegungen der beiden Achsen abrufen. Zusätzlich wird die Schwerkraft beachtet, sodass der Spieler sich durchgehend auf dem Boden befindet, auch, wenn der Untergrund uneben ist. Falls der Spieler sich aufgrund von Unebenheiten nicht mehr auf dem Boden befinden sollte, wird ein Richtungsvektor der nach unten zeigt auf den Bewegungsvektor addiert. Der entstandene Vektor wird noch mit der Konstante *speed* und der Variable *deltaTime* multipliziert, wodurch die Geschwindigkeit der Bewegung gesteuert und zusätzlich unabhängig von der Framerate durchgeführt wird. Dies ist notwendig, da die Update Funktion

einmal pro Frame aufgerufen wird.

Letztendlich wird die Bewegungs mit der Funktion *Move()* des CharacterControllers umgesetzt, welcher der Vektor *moveVector* übergeben wird, der die Informationen über die Bewegung enthält.

5.2 Kamerabewegung

Das Skript *MouseLook* ist verantwortlich für die Kamerabewegung. Dafür werden die X und Y-Bewegungen des Mauszeigers abgerufen. Dies wird ebenfalls über die *GetAxis()* der Input Klasse durchgeführt. Für die Bewegung wird ein Float genutzt, der das Produkt aus den Werten der Mausbewegung, der Variable *mouseSpeed* und der Variable *deltaTime* bildet. Um nach links und rechts schauen zu können, wird der Input der X-Achse der Maus genutzt um eine Rotation um die Z-Achse durchzuführen.

Damit nach oben und unten geschaut werden kann, wird der Input der Y-Achse der Maus genutzt um eine lokale Rotation um die X-Achse durchzuführen. Dabei wird diese Rotation auf die Maximalwerte von 90 und -90 Grad begrenzt, um es realistischer zu machen. Es ist also nur möglich den Blick um 180 Grad von oben nach unten zu richten.

6 Fazit

Das erstellte Projekt *Japanischer Garten* erfüllt alle Anforderungen der Aufgabenstellung. Idee, Aufbau und die Objekte des japanischen Teichgartens orientieren sich stark an der japanischen Kultur. Alle verwendeten Objekte wurden selbstständig modelliert, wobei das Designkonzept Low Poly genutzt wurde. Die Umsetzung der Szene erfolgt mit der Gameengine Unity, wobei viele Features, wie Tag/Nacht Zyklus, Terraingestaltung, Belichtung, Kameraführung, Kollisionserkennung, Menüführung und Animationen verwendet werden. Die finale Version ist auf einem herkömmlichen Windows-Computer bei guter Performance ausführbar.

In späteren Versionen des Projektes wäre es denkbar, für das gesamte Terrain, also auch für Details und Texturen ausschließlich eigene Materialien zu verwenden, wie es bei Objekten, Bäumen und Sträuchern bereits der Fall ist. Der Garten kann erweitert und zusätzlich mit neuen Objekten versehen werden. Mit Ausweiten des Animation Controllers kann der Garten noch lebendiger gestaltet werden. Anstatt der freien Betrachtung kann eine Spielmechanik verwendet werden, um die Geschichte hinter den Objekten darzustellen und weitere Interaktionsmöglichkeiten für den Nutzer zu schaffen.

Zusätzlich besteht die Möglichkeit, die Szene dank der WebGL Technologie auf Webseiten einzubinden, wodurch die Anwendung in Zukunft auch für mobile Endgeräte optimiert werden könnte. Die eben erwähnten Ideen sind nicht Teil der aktuellen Version, sondern sollen einen Ausblick darstellen.

Literatur

- [Garten-Blog] GARTEN-BLOG: *Bäume und Pflanzen für den japanischen Garten* [Asian Lifestyle 2020] ASIAN LIFESTYLE: *Koi Fisch - Mythos und Bedeutung.* April 2020. – URL <https://www.asienlifestyle.de/ueber-uns/unser-blog/koi-fisch-mythos-und-bedeutung>. – Zugriff am: 21 November 2021
- [Blender 2021] BLENDER: *The Freedom to Create.* November 2021. – URL <https://www.blender.org>
- [Bonsai Empire o.J.] BONSAI EMPIRE: *Was bedeutet Bonsai?* September o.J.. – URL <https://www.bonsaiempire.de/ursprung/bonsai-definition>. – [Zugriff am: 21. November 2021]
- [Christian Kaden 2021] CHRISTIAN KADEN: *Japanische Gärten – Typen und Unterschiede.* März 2021. – URL <https://japan-kyoto.de/japanische-gaerten-typen/>. – Zugriff am: 15 November 2021
- [Diana Casanova 2021] DIANA CASANOVA: *Tiere in der japanischen Mythologie und Folklore.* Juni 2021. – URL <https://www.japandigest.de/kulturerbe/religion/tiere-japanische-mythologie/>. – Zugriff am: 21 November 2021
- [Domestika 2021] DOMESTIKA: *What is Low Poly? Learn about Polygon Art for Video Games and More.* Januar 2021. – URL <https://www.domestika.org/en/blog/6206-what-is-low-poly-learn-about-polygon-art-for-video-games-and-more>. – Zugriff am: 15 November 2021
- [Gaël 2021] GAËL: *Der Unterschied zwischen Tempeln und Schreinen in Japan.* September 2021. – URL <https://voyapon.com/de/unterschied-tempel-und-schreine-in-japan/>. – [Zugriff am: 21. November 2021]
- [Japanischer Garten 2017] JAPANISCHER GARTEN: *Bedeutung des Japan Gartens.* Juni 2017. – URL <https://www.japanischer-garten.eu/japanischer-garten-bedeutung/>. – Zugriff am: 18 November 2021
- [Kizuki 2019] KIZUKI: *Was ist Ramen Geschichte von Ramen.* Oktober 2019. – URL <https://www.kizuki.com/whats-ramen-history-of-ramen/>. – Zugriff am: 15 November 2021
- [Miho Shimizu 2021] MIHO SHIMIZU: *What is Torii Gate? Meaning, Types and Facts.* November 2021. – URL <https://japan-kyoto.de/japanische-gaerten-typen/>. – Zugriff am: 19 November 2021

[Unity Technologies 2021] UNITY TECHNOLOGIES: *Sie fragen sich was Unity ist? Entdecken Sie, wer wir sind, wo wir angefangen haben und wohin wir uns entwickeln / Unity.* 2021. – URL <https://unity.com/de/our-company>. – Zugriff am: 19. November 2021]