Dokumentation ZHAW Orbitarium

# Hardware

## Auf-/Abbau

Folgend wird beschrieben, wie das Orbitarium auf- bzw. abgebaut werden kann. Da es im momentanen Zustand montiert ist wird schrittweise erklärt, wie es demontiert werden kann. Für den Aufbau sollte diese Anleitung einfach rückwärts nachvollzogen werden.

Am Schluss dieser Anleitung findet sich eine Auflistung des benötigten Materiales.

### 1. Kabel

Die Projektoren sind mit 2 Kabeln angeschlossen: 1x DVI für den grafischen Input und 1x Stromkabel. Die Stromkabel können an ihren Verteilersteckleisten eingesteckt bleiben. Die DVI Kabel hingegen sind zu entfernen und separat zu verschieben.

Mittig aus der Oberseite des Projektorgehäuses herausschauend findet man das Hauptstromkabel. Ebenfalls an dieser Oberseite befindet sich der Alarmmelder, der angibt, falls das Orbitarium zu stark pendelt. Für dieses Gerät findet sich ebenfalls ein Stromkabel. Beide Kabel sind nicht zu entfernen. Die Kabel sollten ebenfalls nicht von dem Projektorgehäuse geschraubt werden (siehe Abbildungen 1 und 2).



Abbildung 1: An Projektorgeäuse befestigtes Hauptstromkabel

Abbildung 2: An Projektorgehäuse befestigtes Verteilerstromkabel

### 2. Die Projektoren

Um die Projektoren aus dem Projektorgehäuse zu entfernen, sollte auf der Gegenseite des zu entfernenden Projektors das Projektorgehause stabilisiert werden. So kann verhindert werden, dass durch den Gewichtunterschied das Orbitarium zu pendeln beginnt.

**Achtung:** Die Projektoren sind schwer! Falls möglich sollte eine Hebebühne organisiert werden, um die Projektoren sicher auf den Boden zu bringen.

### 3. Spiegeloptik

Mit den entfernten Projektoren sieht man nun die Spiegeloptik. Diese ist auf einem Holzbrett festgemacht. Auf dem Brett sieht man ausserdem die Fassungen, welche die Projektoren in die richtige Position bringen.

Das ganze Brett ist zu entfernen ohne etwas darauf zu entfernen. Es ist dabei darauf zu achten, dass das Brett angehoben werden muss, da es vier Aussparungen für unterliegende Muttern besitzt. Da dies das Herzstück des Oribitariums ist, ist hierbei äusserste Vorsicht geboten.

### 4. Orbitariumhalsfassung

Mit der entfernten Spiegeloptik bietet sich nun die Sicht auf die Orbitarumhalsfassung, welche auf der Unterseite des Projektorgehäuses befestigt ist. Abbildung 3 zeigt diese Sicht von innerhalb des Projektorgeäuses.

Abbildung 3: Sicht auf die Orbitariumhalsfassung von innerhalb des Projektorgehäuses

**Wichtig:** Diese Öffnung führt direkt in das Innere des Orbitariums. Es ist mit äusserster Sorgfalt darauf zu achten, dass nichts in diese Öffnung fällt. Dieser Gegenstand wäre nicht nur so gut wie verloren, sondern er könnte auch den halbrunden Projektionsspiegel im Innern der Kugel beschädigen und somit das Orbitarum als Ganzes unbrauchbar machen!

Das komplette Gewicht der Kugel wird von diesen vier Schrauben bzw. den aufgeschraubten Muttern getragen. Um diese zu lösen muss das Gewicht der Kugel von etwas/jemandem getragen werden. Es ist ratsam dafür mehrere Personen zur Verfügung zu haben und/oder einen Gabelstapler mit dem passenden Lagerungsring zur Hand zu haben (siehe hierzu Abbildungen 4 & 5).

Abbildung 4: Geneigtes Orbitarium für den Transport

Abbildung 5: Gabelstapler mit Lagerungsring

Von der Fassung sollten jeweils nur die Muttern gelöst werden, nichts Weiteres! Die Muttern sind ausserdem so gewählt, dass sie nach ein paar wenigen Umdrehungen verkeilen. Es ist nicht gewollt, dass sie bis zu unterst runtergeschraubt werden können.

### 5. Das Projektorgehäuse

Gleich wie beim Orbitarium selber, sollte auch das Gewicht des Projektorgehäuses gestützt werden, bevor dieses abgeschraubt werden kann. Hierfür empfiehlt sich ebenfalls ein Gabelstapler (allenfalls mit zusätzlichen Paletten für die fehlende Höhe), wie in Abbildung 6 gezeigt ist.

Abbildung 6: Gabelstapler zur Abstützung des Gewichtes

Um mit dem Gabelstapler über das Metallgerüst zu kommen sollte eine Rampe verwendet werden, um überhaupt über die Streben am Boden zu kommen und danach auf der Innenseite des Gerüstes weitere Holzbretter oder Riffelbleche um eine Ebene zu erreichen.

Sobald das Gewicht gestützt ist, können die Muttern auf der Oberseite der tragenden Metallplatte gelöst werden und das Gehäuse mittels des Gabelstaplers gesenkt werden.

### 6. Die Metallplatte

Die Metallplatte an sich befindet sich auf der Oberseite des kompletten Metallgerüstes. Sie ist nur daraufgelegt und nicht angeschraubt! Damit diese jedoch trotzdem gesichert ist, muss darauf geachtet werden, dass die vorderen 2 Schraublöcher (von der Vorderseite der Installation betrachtet) bzw. später die 2 vorderen Schrauben des Projektorgehäuses vor der äussersten Strebe des Gerüstes vorbei geht. Die hinteren 2 Schraublöcher bzw. Schrauben, sollten hinter der mittleren Strebe des Gerüstes vorbei gehen (vergleiche hierzu mit beigefügtem Konstruktionsplan (separates Dokument)).

### 7. Das Gerüst

Die Aufhängung bzw. das Gerüst als solches ist aus Stahl und daher sehr schwer. Es empfiehlt sich also das Gerüst zu kippen und liegend auseinander zu schrauben. Für den Aufbau natürlich im Umgekehrten Sinne: liegend zusammenschrauben und danach aufstellen.

Abbildung 7: Die 0.5m langen Zwischenstücke

Im Momentanen Zustand (nach der Beendigung des Projektes am 09.08.2020) ist das Gerüst ohne die Verlängerungsstücke zusammengebaut. Dies erleichtert die Konfiguration der Projektoren. Für die finale Installation sollten die Zwischenstücke (Abbildung 7) jedoch montiert werden.

## Materialliste für den Auf-/Abbau

|  |  |
| --- | --- |
| Material | Grund & Details |
| Hebebühne | Um die Installationsschritte in der Höhe ausführen zu können.  Höhe: min. 3m besser 4m. |
| Leiter | Für die Zweitperson um zu fixieren und kleinere Einstellungen vorzunehmen. Höhe: min. 1.3m |
| Gabelstapler | Zur Gewichtstützung des Orbitariums & Projektorgehäuses. |
| «Pallet-Rolli» | Um die einzelnen Teile auf Palletten zu bewegen. |
| Zusätzliche Palletten | Stapelbar auf dem Gabelstapler um zusätzliche Höhe zu gewinnen (siehe Abbildung 6). |
| Rampe | Eine Rampe um den Gabelstapler über die Metallstreben des Gerüstes am Boden fahren zu lassen. Höhe: min. 5cm |
| Riffelblech / Holzbretter | Um auf er Innenseite des Metallgerüstes am Boden eine ebene Fläche für den Gabelstapler zu bilden. Höhe: optimal 4-5cm |
| Lagerungsring | Ring um das Orbitarium zu lagern/transportieren, damit dieses nicht zu rollen beginnt |
| Spannsets | Zur Befestigung der Palletten auf dem Gabelstapler und zum Transport der Teile. Min. 6 Stk. à ca. 5m (je mehr desto besser). |
| Schrauben / Muttern | Benötigte Teile sind entweder bereits verbaut oder im/auf dem Projektorgehäuse zu finden. |
| Zwischenstücke | Die Zwischenstücke des Gerüstes um eine Verlängerung der Installation von 0.5m zu erreichen. |
| Werkzeuge | Allerlei Werkzeuge ist nötig. |
| Lieferwagen | Für den Transport ist ein genügend grosser Lieferwagen nötig.  Empfehlung: <https://mietauto.ch/de/Fahrzeuge/Reservieren?cmd=reserveVehicle&id=53> |
| Konstruktions-zeichnungen | Die Konstruktionszeichnungen für das Gerüst. Zusätzliches Dokument mit dem Namen: Orbitarium\_Geruest\_Zeichnung.pdf |

## Zusätzliches Inventar

* 2 PCs mit alter Projektionsmethode vom Technorama
* 2 PC-Stromkabel
* 1 DVI zu DVI Kabel (2x weiblicher Ausgang)
* 1 Entwicklungs-PC (inkl. Maus, Tastatur & Stromkabel)
* 2 DVI zu DisplayPort Converter (in Gebrauch)
* 1 DVI zu DisplayPort Adapter (ungebraucht)
* 2 Projektoren (alt), wobei einer davon defekt ist

# Software

Um das Orbitarium anzusteuern und mit Daten zu versorgen, laufen auf dem Orbitarium-PC verschiedene Programme. Zum einen wird Software von Drittanbietern verwenden, zum anderen kommt eine eigens entwickelte Softwarearchitektur zum Einsatz. Sämtliche Programme sind bereits auf dem Orbitarium-PC installiert und können verwendet werden.

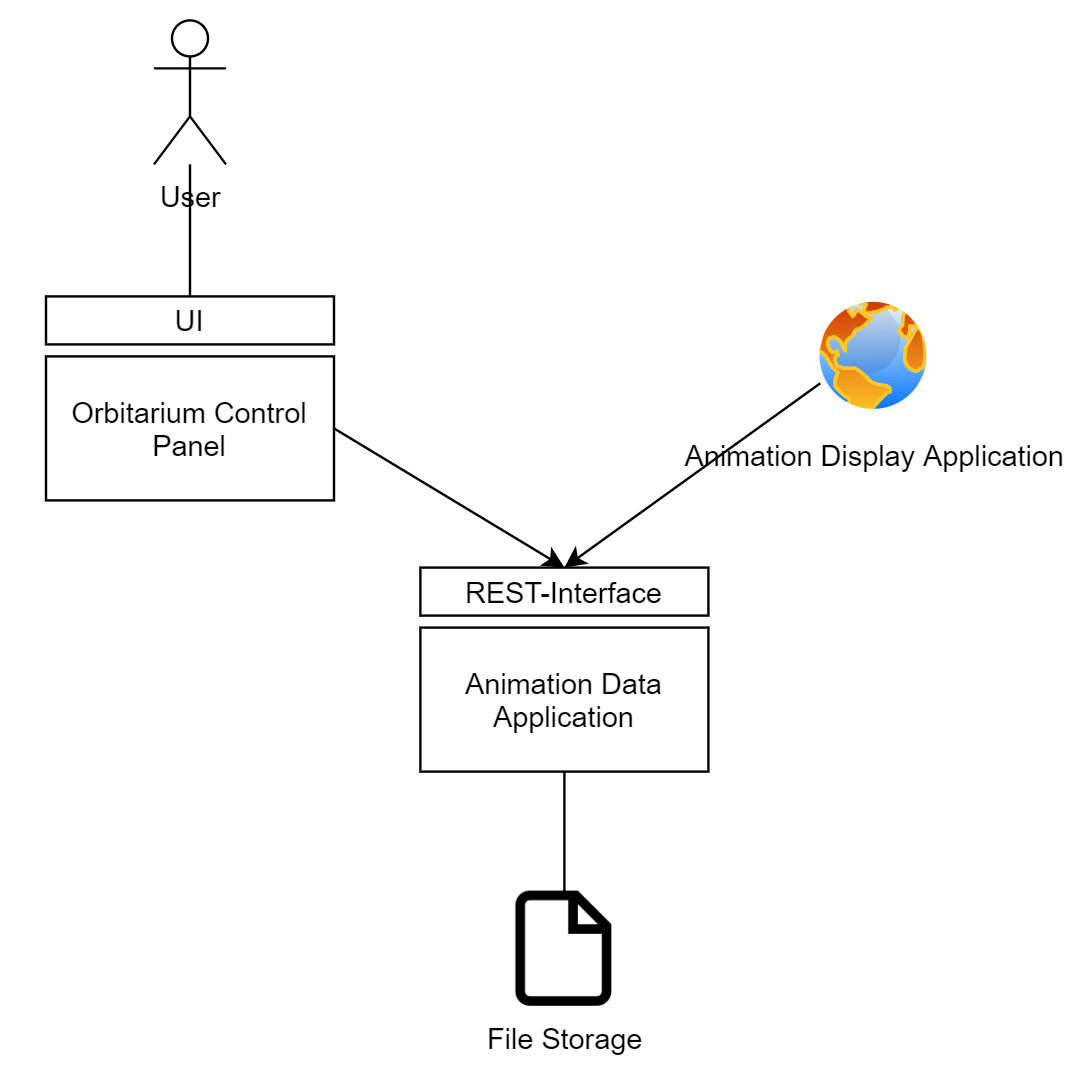
## Software von Drittanbietern

|  |  |
| --- | --- |
| Softwareprodukt | Einsatzzweck |
| IIS | Hosting der eigens entwickelten Software (siehe Folgekapitel) |
| spacedesk virtual display driver | Erweiterung der bestehenden Monitore durch Hinzufügen eines virtuellen Monitors. Hier läuft ein Browser im Vollbildmodus mit der Animation |
| ManyCam | ManyCam nimmt den virtuellen Bildschirm auf. Wir haben eine Lifetime-Vollversion (user: [karl.rege@zhaw.ch](mailto:karl.rege@zhaw.ch), pw: JrXKGW$Kw2g6469) |
| PToolset2.20.04 | Tool zum Konfigurieren und Anpassen der Projektoren vom Hersteller der Projektoren. |
| Postman | Um den REST-Service der Animation Data Application zu konsumieren (Erstellen neuer Animationen / Updaten bestehender Animationen) |
| Visual Studio 2019 Community Edition | Programmierung der Animation Data Application, Programmierung der Unity-Skripte. |
| Unity Editor | Entwicklung des Orbitarium Control Panels |

## Eigenentwicklungen

|  |  |
| --- | --- |
| Softwareprodukt | Einsatzzweck |
| Animation Data Application | Eine Asp.Net Webapplikation mit einem REST-Service. Enthält die Animationen. Enthält Logik, um Animationen zu erstellen. Verwaltet den Zustand der aktuellen Animation. Kann beliebig erweitert werden, um weitere Animationen zu erstellen und zu verwalten |
| Animation Display Application | Eine Web-Applikation. Bindet Google Maps ein, um die Erde darzustellen. Konsumiert Daten von der Animation Data Application. |
| Orbitarium Control Panel | Unity-Applikation. Steuert über die Animation Data Application die Animationen. Startet den virtuellen Display Driver. Startet den Content-Browser und navigiert ihn zur selektierten Animation. Liest von ManyCam die auf dem virtuellen Display laufende Animation ein. Rechnet das Eingabebild in die korrekte Projektionsart (Azimutalprojektion) um und schickt die resultierende Umrechnung an die Projektoren. |

## Zusammenspiel der Eigenentwicklungen



### Animation Data Application

Die Animation Data Application läuft auf dem IIS unter localhost:12345. Der Source-Code findet sich hier: <https://github.zhaw.ch/rege/Orbitarium-Animation-Service> (private repository, Owner: rege).

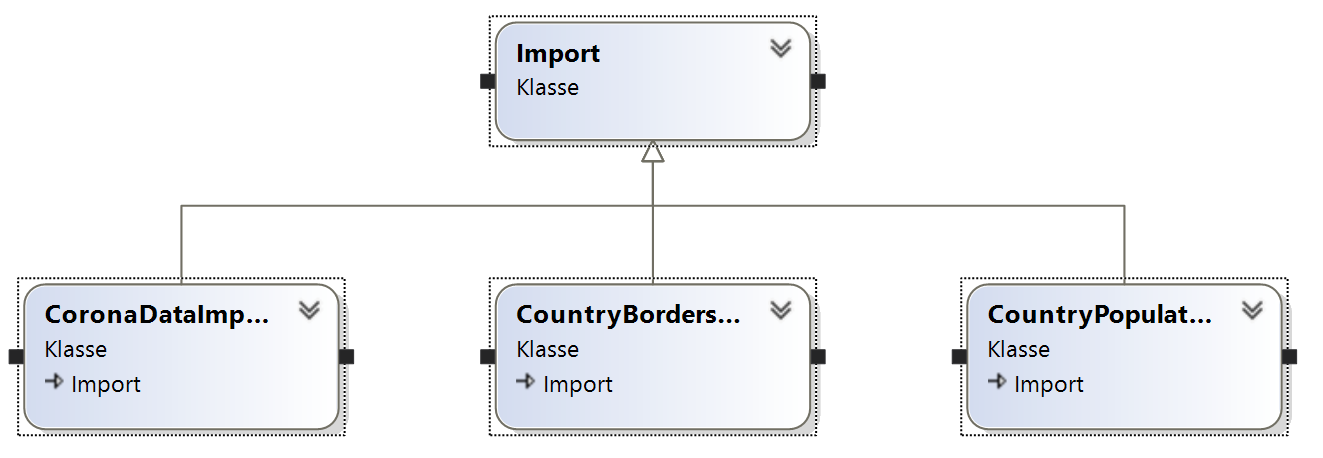
#### Animation Data

Die Daten für alle Animationen liegen auf dem Filesystem im JSON-Format vor. Die Daten sind im Ordner *«Orbitarium-Animation-Service\Animation\_Service\WebApplication1\Data»* abgelegt. Hier finden wir 3 Ordner:

|  |  |
| --- | --- |
| Ordnername | Zweck |
| source | Dieser Ordner enthält die source-Dateien, die für das Generieren der Animation benötigt werden. |
| datastore | Dieser Ordner enthält die importierten Rohdaten. Hier sind die Daten richtig formatiert (serialisierte .NET Datenstrukturen) und können für das zusammenbauen von Animationen verwendet werden |
| animation | Die fertig zusammengebauten Animationen. |

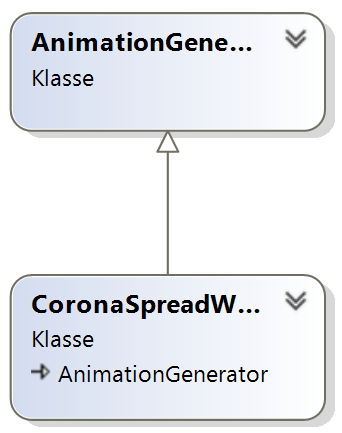
Für das Erstellen von den Animationen sind die Namespaces *Imports* und *AnimationGenerator* wichtig.

#### Namespace Imports

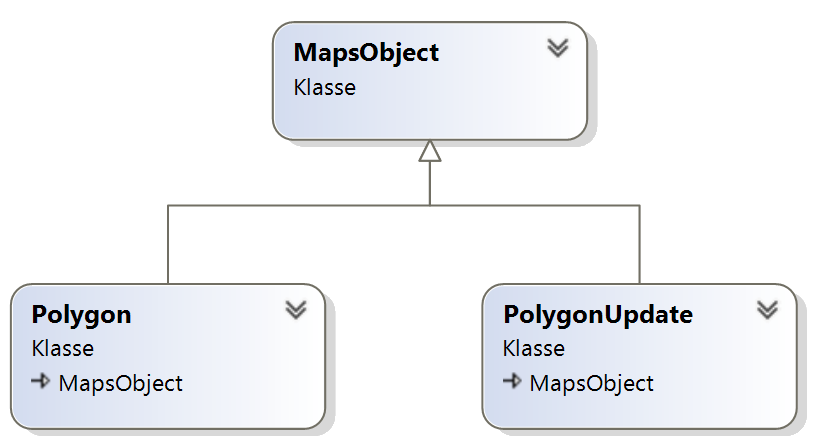


Der Namespace Import importiert verschiedenste Daten. Die Klasse CoronaDataImport beispielsweise liest die Corona-Verbreitungs-Daten ein und legt sie in geeigneter Form im Datastore ab. So kann beispielsweise einfach das Source-File für die Corona-Verbreitung mit aktuelleren Daten ersetzt und danach neu importiert werden. Es können auch einfach neue Import-Klassen geschrieben werden.

#### Namespace AnimationGenerator

Der Namespace AnimationGenerator enthält alle Klassen, die für das Generieren einer Animation nötig sind. Beispiel: Die Klasse *CoronaSpreadWorldwide* legt die Datenstrukturen für die weltweite Verbreitung des Coronavirus so an, dass sie von der Animation Display Application konsumiert werden können und legt die im Ordner Animations ab. Ausserdem schreibt sie einen Eintrag in der Datei *«animations\_information.json»* an, welche das Orbitarium Control Panel braucht, um die verfügbaren Animationen darzustellen.

#### Namespace MapsObjects

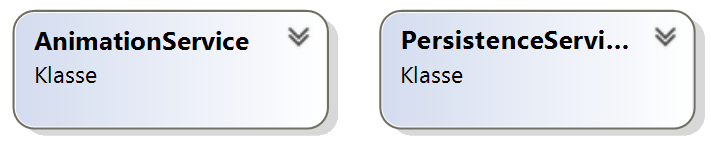
Der Namespace MapsObjects repräsentiert Objekte, wie sie die Google Maps Javascript Api benötigt. Die Klasse Polygon ist eine direkte Abbildung von Polygonen, die Klasse PolygonUpdate eine abgespeckte Variante davon, da für das Update eines Polygons nur spezifische Felder benötigt werden und so der Datentransfer reduziert werden kann.

#### Namespace Models



Der Namespace Models enthält die Repräsentationen verschiedenster Objekte, welche benötigt werden. Üblicherweise werden diese Objekte serialisiert und als JSON abgelegt.

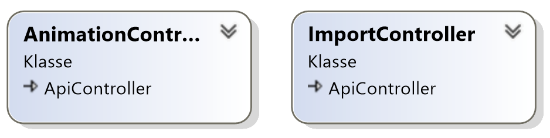
#### Namespace Services

Der AnimationService liefert das JSON mit allen verfügbaren Animationen, der PersistenceService ist zuständig für I/O Operationen auf dem Filesystem.

#### Namespace AnimationDisplay

Die statische Klasse bildet die jetzt gerade laufende Klasse ab. Die Methode «LoadAnimation()» lädt die Corona-Spread-Worldwide-Animation (respektive die Updates) und schreibt alle Steuerungsinformationen in die ensprechenden Klassenvariabeln. Die Methode Start() startet einen neuen Thread, welcher jede Sekunde die nächsten Polygonupdates in die Step-Variable einliest. Sofern diese noch nicht von der Animation Display Application konsumiert wurden, sendet sie die Updates, ansonsten ein «noop».

#### Namespace Controller

Der Namespace Controller stellt die REST-Schnittstelle zur Verfügung und steuert seinerseits wieder das Verhalten der anderen Objekte. Folgende Methoden stellt der REST-Service zur Verfügung:

|  |  |  |
| --- | --- | --- |
| Pfad | Methode | Params (body, as json) |
| http://localhost:12345/api/import | POST | String, z.B. “country\_borders” für den import der country-borders |
| http://localhost:50330/api/animation | POST | String, kommasepariert, “action,animation\_name” z.B. “generate, corona\_spread” zum generieren der Corona-Animation oder “start, corona\_spread” zum starten der Corona-Animation |

### Animation Display Application

Die Animation Display Application läuft auf dem IIS unter localhost:12346. Der Source-Code findet sich hier: <https://github.zhaw.ch/rege/Orbitarium-Animations> (private repository, Owner: rege).

Die Animation Display Application ist dafür zuständig, eine Webseite mit Google Maps darzustellen. Sie ist in Javascript geschrieben und nicht allzu komplex. Sobald die Applikation gestartet wurde und im Orbitarium Control Panel gestartet wurde, macht die Applikation kontinuierliche Abfragen auf die REST-Schnittstelle, um Updates für die darzustellenden Objekte zu erhalten. Die konkrete Implementierung wurde mittels der Google Maps Javascript API (<https://developers.google.com/maps/documentation/javascript/overview>) vorgenommen und wird hier nicht weiter diskutiert.

### Orbitarium Control Panel

Das Orbitarium Control Panel läuft als Standalone-EXE auf dem Orbitarium-PC. Der Source-Code findet sich hier: <https://github.zhaw.ch/rege/Orbitarium-Control-Panel> (private repository, owner: rege).

Das Orbitarium Control Panel wurde mit Unity erstellt. Es ist für das Handling der Applikation bestimmt. Siehe hierzu das Demonstrations-Video im Github-Repository. Ausserdem ist das Orbitarium Control Panel in unserer Bachelor-Arbeit dokumentiert (auch im Github-Repository).