1. Einleitung

Hierbei handelt es sich um ein Projekt im Rahmen des SPP „Multimodal Media Madness“ an der RWTH Aachen im Sommersemester 2020

2. Software

Bei dem Projekt handelt es sich um ein Add-On für das Programm Blender. Dabei wird gefordert, dass Blender mind. Auf der Version 2.80 läuft.

Der Quellcode für das Projekt „Buildo“ wurde in Python geschrieben und arbeitet stark mit dem importierten pby-Paket von Blender selbst zusammen.
Verwendet wird auch das Add-on in Blender, welches die Speicherung von Dateien im OBJ-Format unterstützt. Dieses Add-on ist in Blenderversionen ab 2.80 auch schon vorher eingebettet und aktiviert. Hier ist also grundsätzlich keine zusätzliche Installation nötig.

Erklärungen der Funktionen auf der Codeebene ist dem kommentierten Quellcode zu entnehmen.

2.1 Coloring

Grundsätzlich ist es vorgesehen, die implementierten Farben zu verwenden. Diese sind in einem Reiter bei dem Feld „set co.:“ einzusehen und zu verwenden.
Im Code sind diese dennoch erweiterbar, wenn man in der Datei buildo_panel.py die Funktion *change_color* erweitert. Hierbei hält man sich an die Implementierung der bereits vorgefertigten Farbeinstellungen.
Hierbei darf nicht vergessen werden, dass unter der Methode *change_color* bpy.types.Object.setcolor angepasst werden muss, da die Farbe sonst nicht angezeigt wird.

2.2 Bausteine

Die Bausteine wurden extra einzeln angefertigt und bestehen aus Punkten (verts) und Flächen (faces) zwischen diesen Punkten.
Möchte man einen eigenen Stein einbauen, so muss eine neue Klasse erstellt werden und das Template eines bereits bestehenden Steins in die neue Klasse kopiert werden.
Danach sind die Punkte und Flächen selber nach Belieben in der Methode main zu definieren.
Anzupassen sind dann *bl_idname („myopy.add_STEINNAME“)* und *bl_label („Add STEINNAME“)*.
Die Klasse muss dann in __init__.py eingebunden werden. Das kann beispielhaft durch die anderen Klassen gesehen werden.
Anschließen muss in der Datei buildo_panel.py in der Klasse BuildoPanel die Methode *draw* angepasst werden.
Dabei wird unter dem letzten Knopf, der einen Stein hinzufügt,
row = layout.row()
row.operator("myops.add_STEINNAME")
hinzugefügt wie bei den bereits bestehenden Steinen.

2.3 Das Speichern von Modellen

Nach Erstellung eines Modells lassen sich die Bausteine einzeln als druckbare Wavefront (.obj)-Dateien speichern.
Dabei greift das Add-on indirekt auf das bereits eingebettete Add-on zu, um die Speicherung als .obj möglich zu machen.

Zunächst muss das Modell als .blend Datei abgespeichert werden. Der Ort der .blend Datei wird auch gleichzeitig der Ort, an dem sich die einzelnen Bausteindateien befinden werden. Wenn man vorher nicht die .blend Datei erstellt, kann es sein, dass sich die Dateien der Bausteine nicht bilden lassen (und eine Fehlermeldung ausgegeben wird) oder sie im Ordner des Add-ons landen.