# CODEX AI Sprintplan – Implementierung der Blender‑Simulation

## Kontext

Im Rahmen des Roadmap‑EPIC‑01 soll eine erweiterbare Blender‑Simulation entwickelt werden. Ausgangspunkt sind die vorhandene Deck‑Berechnung (deck\_calculations\_script.py) und diverse technische Dokumente der Sphere Space Station. Ziel ist es, aus den berechneten Deckdaten und ausgewählten Spezifikationen eine CSV mit allen relevanten 3D‑Konstruktionsmaßen zu erstellen und darauf basierend eine erste dreidimensionale Darstellung in Blender zu generieren. Der Code soll innerhalb des Git‑Repositories im Ordner simulations/blender\_simulation strukturiert abgelegt werden.

## Zielsetzung

* Erstellung eines neuen Ordners simulations/blender\_simulation innerhalb des bestehenden Repositories.
* Ableitung einer Datei deck\_3d\_construction\_data.csv aus den Ergebnissen der bestehenden Deck‑Berechnung (simulations/results/deck\_dimensions.csv) und Ergänzung um Ausstattungsmerkmale aus den technischen Dokumenten.
* Implementierung eines Blender‑Python‑Skripts, das die CSV einliest und daraus Deck‑Geometrien, den zentralen Zylinder („Wurmloch“) sowie Sockelringe erzeugt.
* Dokumentation der Simulation und Anleitung zur Ausführung in README.md.

## Aufgaben (User Stories)

| Nr. | Aufgabe | Akzeptanzkriterien |
| --- | --- | --- |
| **1** | **Neuen Ordner strukturieren** – erstelle im Repository den Pfad simulations/blender\_simulation mit Unterdateien | Ordner ist vorhanden; enthält zu Beginn eine .gitkeep oder ist leer. |
| **2** | **CSV generieren** – lies simulations/results/deck\_dimensions.csv ein und leite daraus deck\_3d\_construction\_data.csv ab. Füge zusätzliche Spalten hinzu: usage (Nutzung des Decks), radial\_thickness\_m, windows\_count, window\_material, window\_total\_thickness\_cm, structure\_material, Rotation Velocity, Centrifugal Acceleration. Nutze als Grundlage die Daten aus den technischen Dokumenten (z. B. Materialauswahl und Deck‑Nutzungstabelle). | Die Datei deck\_3d\_construction\_data.csv befindet sich im neuen Ordner. Sie enthält eine Zeile pro Deck (Deck 000 bis Deck 015) mit allen aufgeführten Spalten. Werte sind nachvollziehbar und stimmig zu den Quelldaten. |
| **3** | **Blender‑Skript schreiben** – implementiere blender\_deck\_simulation.py, das die oben genannte CSV einliest und folgende Schritte ausführt: 1. Szene bereinigen. 2. Pro Deck zwei Zylinder erzeugen (innerer und äußerer Radius), mittels Boolean zu einer ringförmigen Schale kombinieren und entlang der Z‑Achse stapeln. 3. Einen durchgängigen Zylinder als zentralen Durchgang generieren. 4. Oben und unten Sockelringe anbringen. 5. Alle Objekte in eine Collection verschieben. | Beim Ausführen des Skripts in Blender werden die Decks entsprechend gestapelt angezeigt. Die Anzahl der Decks entspricht der CSV. Der zentrale Zylinder und die Sockelringe sind vorhanden. Es treten keine Fehlermeldungen beim Skriptlauf auf. |
| **4** | **Dokumentation erstellen** – schreibe README.md im selben Ordner. Beschreibe Zweck, Struktur, Aufbau der CSV, Funktionsweise des Skripts und Anweisungen zur Ausführung in Blender (sowohl in der GUI als auch per Kommandozeile). | README.md liegt vor, ist in verständlichem Deutsch verfasst, beschreibt alle relevanten Schritte und verweist auf die notwendigen Dateien. |
| **5** | **Test & Validierung** – öffne Blender, importiere das Skript und die CSV und führe das Skript aus. Validieren, dass die Modelle erscheinen und die Maße plausibel wirken (z. B. Deck‑Höhen, Radien). | Die Simulation läuft in Blender ohne Fehlermeldungen. Die erzeugten Geometrien stimmen mit den eingelesenen Daten überein (Stichprobenprüfung). |

## Zeitplan / Aufwandsschätzung

| Aufgabe | Aufwand |
| --- | --- |
| Ordner anlegen | 0,5 h |
| CSV ableiten (inkl. Datenanalyse & Ergänzung) | 2 h |
| Blender‑Skript entwickeln & testen | 4 h |
| Dokumentation schreiben | 1 h |
| Integration und finale Tests | 1,5 h |
| **Gesamt** | **9 h** |

## Hinweise für die Umsetzung

* **Blender Version** – Die Entwicklung sollte mit Blender ≥ 2.9 erfolgen, da sich die API seit früheren Versionen geändert hat. Bei späteren Blender‑Versionen (3.x) sind nur geringe Anpassungen zu erwarten.
* **Boolean‑Operationen** – Der Boolen‑Modifier wird im Skript direkt angewendet. Sollte das in einer zukünftigen Blender‑Version Probleme bereiten, kann stattdessen die bpy.ops.mesh.boolean\_mod() API genutzt werden.
* **Fenster & Ausstattung** – Die generierten Decks enthalten bisher keine Fenster oder Inneneinbauten. Eine spätere Erweiterung kann die Spalten windows\_count nutzen, um mithilfe von Arrays oder Booleans Fensteröffnungen entlang der Umfangslinie zu platzieren.
* **Performance** – Bei großen Deckzahlen oder höherer Detailtreue kann die Generierung mehrere Sekunden dauern. Es empfiehlt sich, das Skript zunächst im Hintergrundmodus (blender --background --python …) zu testen.

## Übergabe

Nachdem alle Aufgaben umgesetzt sind, sollten folgende Dateien im Repository verfügbar sein:

* simulations/blender\_simulation/deck\_3d\_construction\_data.csv
* simulations/blender\_simulation/blender\_deck\_simulation.py
* simulations/blender\_simulation/README.md

Diese Dateien bilden die Grundlage für weitere, detaillierte Simulationen der Sphere Space Station und können in zukünftigen EPICs erweitert werden.