Design Specifications

Group 1: Alexander Ramirez, Vivian Casas, Jacky Lim
Nicholas Sisneros, Vince Wang

December 11, 2025

Packages

- **Assimp:** Parses imported .obj files into mesh data.
- **OpenTK:** C# wrapper for OpenGL, used for rendering graphics and viewport updates.
- **Blender (Python API):** Used for mesh simplification and unfolding/export to .svg.
- **Docker:** Containerization of frontend, backend, database, and NGINX server for deployment.

Dependencies

- **Front-End (React.js):**
  - Node.js
  - npm
  - React.js, react-dom
  - Bootstrap
- **Back-End (Node.js + Express):**
  - Node.js (v18-alpine)
  - Express
  - body-parser, cors, dotenv
  - pg (PostgreSQL client)
  - nodemon (development auto-restart)
  - Python 3.x runtime for Blender scripting
- **Database Management:**
  - PostgreSQL (v15-alpine)
  - Docker Volume (dbdata) for data persistence
- **Reverse Proxy:**
  - NGINX (v1.25-alpine)
  - Custom config routes /api requests to backend

Libraries

- Assimp (Open Asset Import Library)
- OpenTK (Open Toolkit Library for OpenGL rendering)
- React.js (JavaScript library for building user interfaces)

- Express (Node.js web application framework)
- pg (PostgreSQL client for Node.js)
- Bootstrap (CSS framework for responsive design)
- Blender Decimation Modifier
- Blender Paper Model Exporter

Pages and Screens

- **Splash Screen:** Displays the application logo and loading status during startup.
- **Viewport:**
  - Import button for uploading 3D models in .obj format.
  - Render region powered by OpenTK for displaying 3D models.
  - Simplify button with bounding-box and decimation options.
  - Export button:
    - Passes simplified model to Blender for unfolding.
    - Produces .svg file with cuts and folds marked.
    - Downloads .svg file to user.
    - Displays unfolded model preview in a new window.
    - Option to print directly from the preview window.
    - Option to save .svg file for later use.
  - UI polish: error handling, loading indicators, responsive design, and edge case support.

Workflow Summary

1. Application Startup → Splash Screen → Viewport
2. User imports .obj file → File parsed by Assimp → Rendered in OpenTK viewport
3. User simplifies model → Simplification options applied → Model updated in viewport
4. User exports model → Model sent to Blender via Python script → Model unfolded and exported as .svg
5. .svg file downloaded to user → Unfolded model preview displayed → User can print or save .svg file

Docker Compose Services

- **frontend:** React application served via Node.js and NGINX.
- **backend:** Express API for modeling logic, database interactions, and Blender Python scripting.
- **database:** PostgreSQL container with persistent volume.
- **nginx:** Reverse proxy and static file server for frontend and backend services.

Repository

- GitHub: UnBox3D