Progress report #5 - 04.15.2020

Main purpose / focus

Support multiple importing formats.
Support multiple exporting formats.
Optimization / improvement of three.js's raycasting.
3D modeling - for testing.

Planned activities

- Implement exporting support
  - BINVOX file format
  - XML file format (coloring)
  - 3D Array (coloring)
  - VOX file format (coloring
  - SCHMATIC file format
- Add importing support of binvox files for three-voxel-loader.
- Implement importing support.
  - STL file format
  - PRWM file format
  - MMD resources
  - glTF file format
  - MTL file format
  - OBJ file format
- Optimize three.js raycasting functionality.
  - Implement a spatial index.
- Create 3D model for testing Voxelizer library.

Deviations

The implementation of the importing support is dropped for the voxelizer library. Three.js provides several "examples" for importing various file formats. Previously, these examples were not included with the base three.js package. The old voxelizer library therefore implemented a wrapper around the OBJ file loader. However, all the loaders are now included in the three.js base package. This makes it much easier to access and use the various loaders. The voxelizer library will therefore leave the process of importing 3D model files into three.js objects up to the user. These loaders produce an Object3D object, which the voxelizer library then can consume.

An AutoLoader was started on, however due to the various loaders requiring a lot of different settings, assets, paths etc. this proved hard to implement properly. Although, this functionality can possibly be used in the voxelizer desktop application.

Completed work

- Several of the exporting formats are implemented.
    - BINVOX file format
    - XML file format (coloring)
    - 3D Array (coloring)
- Importing support of BINVOX files for three-voxel-loader is implemented.

NOTE: The BINVOX functionality has been extracted in a separate repository. See https://github.com/andstor/binvox
It can both parse and build BINVOX files according to the specification.

- The raycasting functionality of three.js is optimized with the three.js plugin https://github.com/gkjohnson/three-mesh-bvh. This utilizes Bounding Volume Hierarchies, effectively reducing the time complexity for raycasting from O(n) to O(log n). It does take some time to generate the tree structure. However, this is negligible for a normal use case of voxelizing a large mesh.
- Created a 3D model of an anvil with Blender.

Anvil 3D model render: