



# Absolute|0

## Voxc.js Requirements Specification

Name	Student Number
Chris Dreyer	15072623
HD Haasbroek	15046657
Cameron Trivella	14070970
Pearce Jackson	14044342
Idrian van der Westhuizen	15078729

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose . . . . .	2
1.2	Product Scope . . . . .	2
1.3	References . . . . .	2
<b>2</b>	<b>External Interface Requirements</b>	<b>3</b>
2.1	User Interfaces . . . . .	3
2.2	Software Interfaces . . . . .	3
2.3	Communications Interfaces . . . . .	3
<b>3</b>	<b>System Features</b>	<b>4</b>
3.1	System Feature 1 . . . . .	4
3.1.1	Description and Priority . . . . .	4
3.1.2	Stimulus/Response Sequences . . . . .	4
3.1.3	Functional Requirements . . . . .	4
<b>4</b>	<b>Other Nonfunctional Requirements</b>	<b>5</b>
4.1	Performance Requirements . . . . .	5
4.1.1	Time to respond to an uploaded file . . . . .	5
4.1.2	Respond to user interaction . . . . .	5
4.1.3	Reliability . . . . .	5
4.1.4	Maintainability . . . . .	5
4.1.5	Portability . . . . .	5
4.1.6	Scalibilty . . . . .	5
4.1.7	Usability . . . . .	5
4.2	Security Requirements . . . . .	6
4.3	Quality Requirements . . . . .	6

# 1 Introduction

## 1.1 Purpose

This SRS document aims to stipulate the requirements of the voxc.js library to aid in the development process and to ensure that a functional and usable product is delivered.

## 1.2 Product Scope

Voxc.js is meant to be an easy to use and easy to maintain JavaScript library, similar to how three.js is a library for WebGL. The purpose of the project is to provide users a way to import Voxel models into a webpage that would be using the Voxc.js library and ultimately texturing these Voxel models according to a rules file with a specified structure. The user will then be able to export the textured and rendered object.

## 1.3 References

<http://www.oskarstalberg.com/game/house/Index.html>

<https://voxel.codeplex.com/>

<https://pages.github.com/>

<http://threejs.org/>

<http://coffeescript.org/>

<http://es6-features.org/#Constants>

<http://www.typescriptlang.org/>

<http://www.codebelt.com/typescript/typescript-es6-modules-boilerplate/>

<http://giacomotag.io/typescript-webpack/>

## 2 External Interface Requirements

### 2.1 User Interfaces

The users should be able to interface with voxc.js through our web interface that will be hosted on Github Pages. They will mainly be using a laptop or a desktop to interface with the library. However the library itself should be able to interface with any website on any device with a web browser that has support for WebGL.

This library should not require user profiles as their voxel objects are stored locally on browser storage and manipulated by the library. This implies that the user should only be led to one screen where they will be allowed to upload a voxel object file and a rules file if they choose to do so.

All users should share the same priority as there is not a distinction between user types. All users should have access to all functions provided by the library.

### 2.2 Software Interfaces

The system should incorporate several different languages in order to function. For the rules file JSON objects should be used so that the users can manipulate the file in any text editor based on set conventions and structure.

MagicVoxel should be used for the creation and manipulation of voxel objects. The library should be able to handle the .obj filetype for the voxel objects.

The library should use TypeScript as the JavaScript variant to enable future developers and current developers to debug easier. The library should run on all major internet browsers that support WebGL.

### 2.3 Communications Interfaces

The system will have to handle frequent user uploads of their Voxel objects in order to manipulate them. \*This communication can be done through any persistent internet connection\*. HTTP will be used to handle GET and POST requests and FTP will be used for file uploads and downloads. This library should also allow multiple users to use the system simultaneously. \*This is to say that the system's performance should not be proportional to the number of active users.\*

### **3 System Features**

#### **3.1 System Feature 1**

##### **3.1.1 Description and Priority**

##### **3.1.2 Stimulus/Response Sequences**

##### **3.1.3 Functional Requirements**

## 4 Other Nonfunctional Requirements

### 4.1 Performance Requirements

#### 4.1.1 Time to respond to an uploaded file

A user is required to upload a voxel object to the library and then should be allowed to manipulate a rules file if they choose to do so or they should be able to use a predefined rules file such that the rules are applied to the object. The library must not delay once a file is uploaded, this means a delay of no more than \*2 seconds\* to respond to the upload should be allowed.

#### 4.1.2 Respond to user interaction

The system should respond to user interactions in real time.

#### 4.1.3 Reliability

The system should never cease working completely unless the error is caused by external systems outside our control (operating system, web APIs, etc). Ideally an entire system uptime (per month) of 99.9% must be reached, meaning that the system should have validation and error checking to prevent unwanted results.

#### 4.1.4 Maintainability

The library's code must be well documented, both by means of in-code comments and external documentation, to aid in maintaining the system. A user manual should also be provided to make it easier to understand and ultimately maintain the library.

#### 4.1.5 Portability

This library should be accessible across all major internet browsers that support WebGL. These include Chrome, Firefox etc. The library should also be able to be imported to any website.

#### 4.1.6 Scalability

\*It must be possible to scale the library backend in the event of an increase of users. Scaling must be possible both horizontally or vertically.\*

#### 4.1.7 Usability

The Voxel library should be easy to operate and understand. The core functions of the system shouldn't take the user more than a minute to access and understand. A user manual should be provided to aid in the use of the library and to explain some advanced functions or settings.

## **4.2 Security Requirements**

This library should be an open source project and so the code will be freely available to anyone visiting the Github repository. The actual web interface hosted on Github will act as a demo of the library and should be protected, the only editing that a user should be able to do is uploading of objects and editing their own rules file.

## **4.3 Quality Requirements**

\*\*\*\*\*