Requirements Specification Document

1.1 Introduction

Parametrix is a web-based animation suite designed for functionally programmed animations, utilizing React and Three.js for the frontend and Firebase Cloud Storage for the backend. The application enables users to create, manipulate, and animate 3D objects using programmable functions. Parametrix provides an interface where users can define transformations and color dynamics through mathematical expressions, allowing for precise, customizable animations that can be quickly generated with a seamlessly integrated interface. Animations are small functions created by the user that loop once or forever using a loop control value over a range of input values that animate at a set sample rate.

1.2 CSCI Component Breakdown

The animation suite, Parametrix, is composed of the following CSCs divided into three main subsystems: User Interaction, Animation Programming, and Website Functionality and Backend. 1. User Interaction: Includes 3D object creation, camera control, and object editing. 2. Animation Programming: Includes functional programming for object and camera animations, loop control, and sample rate management.

3. Website Functionality and Backend: Includes Firebase integration, object storage, and animation rendering.

1.3.1 User Interaction Requirements

The following requirements are levied on the User Interaction subcategory of the Parametrix project.

1.3.1.1 The User Interaction Subsystem shall display the application in a web page.

1.3.1.2a User Interaction Subsystem shall be hosted with React for frontend rendering.

1.3.1.2b User Interaction Subsystem shall be hosted with Three.js for frontend rendering.

1.3.1.3 User Interaction Subsystem shall allow users to create 3D objects.

1.3.1.4a User Interaction Subsystem shall allow users to move the camera around the 3D object using mouse controls.

1.3.1.4b User Interaction Subsystem shall allow users to move the camera around the 3D object using keyboard controls.

1.3.1.5 User Interaction Subsystem shall allow users to right-click on an object to open an edit menu.

1.3.1.6a User Interaction Subsystem shall allow users to modify the object’s position through the edit menu.

1.3.1.6b User Interaction Subsystem shall allow users to modify the object’s rotation

through the edit menu.
1.3.1.6c User Interaction Subsystem shall allow users to modify the object’s scale through the edit menu.
1.3.1.7 User Interaction Subsystem shall allow users to specify an object's color using RGB values.
1.3.1.8 User Interaction Subsystem shall make selected objects glow yellow using a non-user-modifiable layer.
1.3.1.9a User Interaction Subsystem shall change the glow color of selected objects when hovering over delete buttons
1.3.1.9b User Interaction Subsystem shall change the glow color of selected objects when hovering over reset buttons.
1.3.1.10 User Interaction Subsystem shall allow users to delete selected objects.
1.3.1.11 User Interaction Subsystem shall allow users to reset the scene to its initial state.

1.3.2 Animation Programming Requirements

The following requirements are levied on the Animation Programming subcategory of the Parametrix project.
1.3.2.1 Animation Programming Subsystem shall allow users to program a function to control a dimension of an object's position.
1.3.2.2 Animation Programming Subsystem shall allow users to program a function to control a dimension of an object's rotation.
1.3.2.3 Animation Programming Subsystem shall allow users to program a function to control a dimension of an object's scale.
1.3.2.4a Animation Programming Subsystem shall allow RGB values of an object’s color to be mappable to function with a floor of 0.
1.3.2.4b Animation Programming Subsystem shall allow RGB values of an object’s color to be mappable to functions with a ceiling of 255.
1.3.2.5 Animation Programming Subsystem shall allow users to animate the camera's angle using a programmed function.
1.3.2.6 Animation Programming Subsystem shall allow users to animate the camera's position using a programmed function.
1.3.2.7Animation Programming Subsystem shall enable animation as small functions created by the user that loop once or forever.
1.3.2.8 Animation Programming Subsystem shall use a loop control value over a range of input values.
1.3.2.9 Animation Programming Subsystem shall animate at a set sample rate defined by the user.
1.3.2.10 Animation Programming Subsystem shall allow users to define the range of input values for animations.
1.3.2.11 Animation Programming Subsystem shall allow users to set the sample rate for animations.
1.3.2.12a Animation Programming Subsystem shall allow users to toggle animations to loop once.

1.3.2.12b Animation Programming Subsystem shall allow users to toggle animations to loop forever.
1.3.2.13 Animation Programming Subsystem shall provide a visual timeline for animations, showing keyframes and loop points.

1.3.3 Website Functionality and Backend Requirements

The following requirements are levied on the Website Functionality and Backend subcategory of the Parametrix project.

1.3.3.1 Website Functionality Subsystem shall have a header above the 3D canvas.

1.3.3.2a Website Functionality Subsystem shall host the site logo in the header.

1.3.3.2b Website Functionality Subsystem shall host the sign-in button in the header.

1.3.3.2c Website Functionality Subsystem shall host the sign-out button in the header.

1.3.3.2d Website Functionality Subsystem shall host save options in the header.

1.3.3.2e Website Functionality Subsystem shall host load options in the header.

1.3.3.3a Website Functionality Subsystem shall only display the sign-out button when a user is signed in.

1.3.3.3b Website Functionality Subsystem shall only display save options when a user is signed in.

1.3.3.3c Website Functionality Subsystem shall only display load options when a user is signed in.
1.3.3.4 Backend Subsystem shall use Firebase Cloud Storage for saving and loading user projects.
1.3.3.5 Backend Subsystem shall allow users to save their current scene to Firebase Cloud Storage.
1.3.3.6 Backend Subsystem shall allow users to load a previously saved scene from Firebase Cloud Storage.

1.3.3.7 Backend Subsystem shall use Firebase Authentication for user sign-in and sign-out.

1.3.3.8 Backend Subsystem shall allow users to sign in with a Google account.
1.3.3.9 Backend Subsystem shall render animations in real-time using Three.js.
1.3.3.10 Website Functionality Subsystem shall support exporting animations as video files.
1.3.3.11 Website Functionality Subsystem shall support exporting animations as GIF files.
1.3.3.12 Website Functionality Subsystem shall support exporting animations as image sequences.

1.4 Performance Requirements

1.4.1 Parametrix Performance Subsystem shall render 3D objects and animations in real-time without significant lag.

1.4.2 Parametrix Performance Subsystem shall support at least 100 objects in a single scene without performance degradation.

1.4.3 Parametrix Performance Subsystem shall not require any special computing hardware to operate.

1.5 Project Environment Requirements

1.5.1 Parametrix Environment Subsystem shall be able to execute using any standard web browser.
1.5.2 Parametrix Environment Subsystem shall be able to be viewed with any media account.
1.5.3a Parametrix Environment Subsystem shall be able to save projects through Firebase with a Google account.
1.5.3b Parametrix Environment Subsystem shall be able to load projects through Firebase with a Google account.
1.5.4a Parametrix Environment Subsystem should be able to work on desktop devices.
1.5.4b Parametrix Environment Subsystem should be able to work on mobile devices.

1.6 Development Environment Requirements

1.6.1 Development Environment Subsystem should be able to be worked on in any code editor.
1.6.2a Development Environment Subsystem shall require Node.js for frontend code.
1.6.2b Development Environment Subsystem shall require Node.js for backend code.
1.6.2c Development Environment Subsystem shall require JavaScript for frontend code.
1.6.2d Development Environment Subsystem shall require JavaScript for backend code.
1.6.3 Development Environment Subsystem shall require Three.js for 3D rendering and animations.
1.6.4 Development Environment Subsystem shall require React for frontend design and hosting.
1.6.5a Development Environment Subsystem shall require Firebase for authentication.
1.6.5b Development Environment Subsystem shall require Firebase for cloud storage
1.6.5c Development Environment Subsystem shall require Firebase for project management.
1.6.6a Firebase Subsystem shall be used to allow for sign-in.
1.6.6b Firebase Subsystem shall be used to allow for sign-out.
1.6.6c Firebase Subsystem shall be used to allow for project saving.
1.6.6d Firebase Subsystem shall be used to allow for project loading.
1.6.7a Firebase Subsystem shall be used for cloud storage to allow for archiving user projects.
1.6.7b Firebase Subsystem shall be used for cloud storage to allow for hosting user projects.

1.7 Execution Environment Requirements

1.7.1 Execution Environment Subsystem will not need to be executed from a local device aside from hosting with React.

1.7.2 Execution Environment Subsystem will not need to be executed from a local device aside from deployment with Firebase.