Heading

System and Unit Test Report
EZ 3D
Nov. 30, 2021

System Test Scenarios

Sprint 1

User Story 1: As a 3D modeler, I need to have a full window viewport so I can model with ease.
User Story 2: As a 3D modeler, I need to be able to move in the scene so I can better see my models.
User Story 3: As a 3D modeler, I need to be able to bring meshes(cube, sphere, cylinder, etc.) into the scene so I can make assets and environments.
User Story 4: As a 3D modeler, I need all of my options in a nicely formatted dropdown or outliner menu, so I can work efficiently.
User Story 5: As a 3D modeler, I need to be able to clearly see the object and its relation to the ground.

Scenario for user stories 1 to 5:

1. Start the ez3d app by entering the app directory from terminal
    - npm install all dependencies
    - then npm start
    - Wait until a browser window pops up in Chrome at localhost:3000
    - User should see the full window viewport
2. To control the camera, scroll up/down, user should see camera zoom in and out
    - Hold down left mouse button and drag around, user should see the camera orbit around center of screen
    - Hold down right mouse button and drag around, user should see the camera move left/right and up/down the scene
3. Click on ‘add’ in the toolbar on top - user should see a drop down menu listing shapes to add
    - Click ‘cube’ - user should see a blue cube appear in the center of the screen
    - Click ‘sphere’ - user should see a purple sphere appear in the center of the screen
    - Click ‘cylinder’ - user should see a yellow cylinder appear in the center of the screen
4. On load, the toolbar is immediately visible to the user
    - Furthermore, a menu is also visible to the user on the right of the screen, displaying an object list and options to toggle the grid, shadows, and fog
    - Clicking on an existing object in the scene will select the object, and display the object information in the right menu (outliner) to the user.
    - Under ‘model’, the user can see three rows of textboxes with numbers containing the position, scale, and rotation, respectively. They should also see the color of

the object represented in a box, a toggle to change the render style between ‘wireframe’, ‘flat’, and ‘smooth’, and finally the visibility of the selected object.

5. The user should see a grid (which represents the ground) in the scene on load.

Sprint 2

User Story 1: As a 3D modeler, I need to be able to manipulate Translate/Rotate/Scale with click/drag and/or by units so I can be as precise as I want.
User Story 2: As a 3D modeler, I need to have hotkeys between Translate, Rotate, Scale so I can work efficiently.
User Story 3: As a 3D modeler, I need all of my options in a nicely formatted dropdown or outliner menu, so I can work efficiently.

Scenario for user stories 1 to 3:

1. Click on ‘add’ in the toolbar on top - user should see a drop down menu listing shapes to add
    - Click ‘cube’ - user should see a blue cube appear in the center of the screen
2. Click on the shape to select it
    - A control mechanism should appear on the shape
    - Click and hold on the mechanism to manipulate the selected shape
3. Click on one of the options in the toolbar or press a hotkey to change modes
    - The toolbar is on the top left with three buttons
        - Click the first button to change to ‘translate’ mode - this allows the user to move the object
        - Click the second button to change to ‘scale’ mode - this allows the user to scale the object
        - Click the third button to change to ‘rotate’ mode - this allows the user to rotate the object
    - The hotkeys corresponding to the modes are ‘w’, ‘e’, and ‘r’
        - Press the ‘w’ key to change to ‘translate’ mode
        - Press the ‘e’ key to change to ‘scale’ mode
        - Press the ‘r’ key to change to ‘rotate’ mode
4. Use the transforms UI to provide more precision on manipulating the shape
    - The transforms UI is on the right pane of the shape titled “Transformation”
    - Each mode is divided into the x, y, and z parameters which correlates to the axis in which the shape will be manipulated on
    - Use the numbers to change the values or write a number in the input to more precisely manipulate the shape
        - The first row represents the ‘translate’ mode
            - Hover over the input on the first row, first column titled ‘X’
            - Click on the arrows to change values - click and hold on the arrows to change values faster
            - Type ‘10’ in the same input
            - Doing either or both of these actions should move the object on the x-axis

        - The second row represents the ‘scale’ mode
            - Hover over the input on the second row, second column titled ‘Y’
            - Click on the arrows to change values - click and hold on the arrows to change values faster
            - Type ‘10’ in the same input
            - Doing either or both of these actions should scale the object on the y-axis
        - The third row represents the ‘rotate’ mode
            - Hover over the input on the third row, third column titled ‘Z’
            - Click on the arrows to change values - click and hold on the arrows to change values faster
            - Type ‘10’ in the same input
            - Doing either or both of these actions should rotate the object on the z-axis

Sprint 3

User Story 1: As a 3D modeler, I need to be able to manipulate the light sources so the shading looks the way I want.
User Story 3: As an artist, I need Wireframe/Faceted/Smooth shading options so I can specialize the aesthetic of my scene.
User Story 4: As a 3D modeler, I need to be able to change the color of each of my meshes so I can make distinct meshes in my scene.

Scenario for user story 1:

1. Click on ‘add’ in the toolbar on top - user should see a drop-down menu listing types of lights they can add.
    - Click ‘Ambient Light’ - user should see a sphere that represents ambient light appears above the center of the screen
    - Click ‘Directional Light’ - user should see a sphere and a plane that represents directional light appears above the center of the screen
        - Change the position of the light source, user should see the shadow against the directional light
    - Click ‘Point Light’ - user should see a sphere that represents point light appears above the center of the screen
2. Click the sphere to modify the light
    - Click on the sphere to select it
        - A control mechanism should appear on the sphere
        - Click and drag the mechanism to manipulate position of the light.
3. Use UI on the right pane to modify the light
    - User should see the transforms UI on the right pane - this allow user to change the transformation.
        - Click the input on the first row, first column titled "X".
        - Click the arrow in this box to change the transformation on X-axis - click and hold the arrow to change the value faster.

        - Type 10 in the same input to set transform value directly.
    - User should see the Color UI on the right pane - this allow user to change the light color.
        - Click the white rectangle(default color) on the color UI, user should see a color pallet - this allows the user to change the color of the light.
    - User should see the intensity UI on the right pane - this slide bar allow user to change the intensity of the light.
        - Change the position of the button on the slide bar, user can change the intensity of the light.

Scenario for user stories 3 to 4:

1. Start the app, click the Add tag on the top toolbar
    - User should see a list of mesh, click “Cube”
2. Click the mesh in the scene, user should see all modification UI on the right pane.
    - User should see the Shader UI on the right pane - it allows user to change the shading option.
        - Click "Wireframe" to set the wireframe shading.
        - Click "Flat" to set the flat shading.
        - Click "Smooth" to set the smooth shading.
    - User should see the Color UI on the right pane - this allow user to change the mesh color.
        - Click the white rectangle(default color) on the color UI, user should see a color pallet - this allows the user to change the color of the mesh.
    - User should see the visibility UI on the right pane.
        - Click the "eye" in UI, user can make the mesh invisible. The mesh is still selectable when invisible.

Sprint 4

User Story 1: As a regular user of EZ 3D, I need to be able to export/import my work so I can easily pick up where I left off

Scenario for user story 1:

1. Start the app
    - In the upper toolbar click ‘add’, then click ‘cube’ to add a cube
    - The user should see a scene with a cube
    - In the upper toolbar click ‘file’, then click ‘save’ in the window that pops up
    - The scene will be saved and automatically download. The user can see the scene saved in a file marked ‘project.ez3d’ in their downloads folder.
2. To import any work, refresh the page to clear it
    - In the upper toolbar click ‘file’, then click ‘Choose File’ in the window that pops up.

    - A prompt will open up, showing the user's file system.
    - Select the project.ez3d file that was downloaded earlier, and click 'open'
    - The user will now see the saved scene loaded up into the editor

User Story 2: As a 3D modeler, I need to be able to bring more meshes (cone, tetrahedron, torus, etc.) into the scene so I can make assets and environments.

Scenario for user story 2:

1. Start the app
    - In the upper toolbar, click on 'add'
    - The user will see a window with a list of selections
    - Click on 'cone'
    - The user will see a cone in the center of the scene
    - Click on 'add' in the upper toolbar again and click 'torus'
    - The user will see a torus appear in the center of the scene
    - Click on 'add' in the upper toolbar again and click tetrahedron
    - The user will see a tetrahedron appear in the center of the scene
    - Repeat the above for the rest of the shapes in the 'add' list

User Story 3: As a 3D modeler, I need all of my meshes to be in the outliner and the ability to hide/name objects in the scene.

Scenario for user story 3:

1. Start the app
    - In the upper toolbar, click on 'add'
    - Click on 'cube'
    - The user will see a cube appear in the scene
    - The user will also see the cube appear as 'model 1' in the Object List panel on the right, with a text box next to 'model 1'
    - Type 'this is a name' into the text box to name it
    - Click on the cube to select it
    - The user will see a subsection in the right pane labeled 'Model'
    - Click on the eye icon next to the text that says 'Visibility'
    - The user can see that the selected shape is hidden

User Story 5: As a 3D modeler, I want to be able to toggle fog on/off.

Scenario for user story 5:

1. Start the app by following the same path as sprint 1 user story 1
    - After startup, in the menu on the right side of the screen, the user should see in the area marked 'Scene' a switch marked 'Fog'.
    - Click the switch
    - The user should see that the fog is gone
    - Click the switch again

- The user should see that the fog is toggled back on

User Story 7: As a 3D modeler, I need to be able to undo or redo my actions

Scenario for user story 7:

1. Start the app by following the same path as sprint 1 user story 1
    - After startup, the user should see the toolbar in top half of the screen
    - Click ‘Add’
    - Click ‘Cube’
    - The user should see a scene with a cube
    - Click on the cube
    - Type ‘CTRL’ and ‘z’ to undo the action
    - The user should see an empty scene, reverting the what it was like on startup
    - Type ‘CTRL’ and ‘y’ to redo the action
    - The user should see a scene with a cube
    - Select the cube and press ‘CTRL’ and ‘x’
    - The user should see that the cube is deleted
    - Type ‘CTRL’ and ‘x’
    - The user will see that the delete action has been undone
    - Type ‘CTRL’ and ‘y’
    - The user will see that the delete action has been redone, resulting in an empty scene
2. With the cube still in the scene, click on the cube to select it
    - The user should see arrows coming out of the center of the cube
    - Drag the upward facing green arrow upwards and let it go
    - The user will see that the cube is in a different position
    - Type ‘CTRL’ and ‘z’
    - The user will see that the cube has returned to its original position
    - Type ‘CTRL’ and ‘y’
    - The user will see that the cube has returned to its modified position, nullifying the undo action
3. Using the same scene, add another cube and click on it to select it
    - In the right pane marked ‘model’, and under the section ‘transformation’, type ‘1’ into the cell in the first row and first column to set the x position of the object to ‘1’
    - The user will see that the cube has shifted one unit along the x axis
    - In the same transformation section, type ‘0’ into the cell in the second row, first column, resulting in the number 10 (1 is already inside the box). This will set the size of the object along the x axis to 10.
    - The user will see that the object has been stretched along the x axis.
    - In the same transformation section, type ‘45’ into the cell in the third row
    - The user will see that the object has been rotated by 45 degrees along the x axis
    - Type ‘CTRL’ and ‘z’ twice
    - The user will see that the rotation has been undone

- Type ‘CTRL’ and ‘z’
- The user will see that the stretch has been undone
- Type ‘CTRL’ and ‘z’
- The user will see that the move has been undone, reverting the square back to its original state completely
- Type ‘CTRL’ and ‘y’
- The user will see that the object movement has been re-done
- Type ‘CTRL’ and ‘y’
- The user will see that the stretch has been re-done
- Type ‘CTRL’ and ‘y’
- The user will see that the rotation has been re-done

Unit Tests

Module: Undo

Function: If CTRL + z is pressed, determine whether or not there is anything to undo, and if there is, undo it by reverting to the last state

Equ. Class	Description	Possible Values
$EC_{z,notCTRLZ}$	These keys do not trigger the running of the execution	A,b,c,d,e,f,g,h, CTRL,z,CTRL+y,del
$EC_{z,CTRLZ}$	This key combination triggers the execution	‘CTRL+z’
$EC_{z,notStored}$	These states result in the function being unable to fully execute	The scene is completely blank, there are no stored states
$EC_{z,stored}$	This state allows the function to fully execute	There is a stored last state

Test Cases:

Equ. Class Combos	Input	Expected Output
$EC_{z,notCTRLZ} + EC_{z,notStored}$	‘c’ + no saves	No undo
$EC_{z,notCTRLZ} + EC_{z,stored}$	‘f’ + saved state	No undo

$EC_{z,CTRLZ}$ + $EC_{z,notStored}$	‘CTRL + z’ + no saves	No undo
$EC_{z,CTRLZ}$ + $EC_{z,stored}$	‘CTRL + z’ + saved state	Undo executes, reverts to last saved state

Module: Color Selector

Function: If Color is changed in the object bar, determine whether or not there is an object selected. If there is, change the color of that object to the selected value.

Equ. Class	Description	Possible Values
$EC_{col,valid}$	Valid red, green, and blue inputs	0 <= red <= 1 & 0 <= green <= 1 & 0 <= blue <= 1
$EC_{col,error}$	Invalid rgb values	(Red < 0 || Red > 1) || (Blue < 0 || Blue > 1) || (Green < 0 || Green > 1)
$EC_{obj,\ selected}$	If a valid object is selected	Mesh object, light object selected
$EC_{obj,\ notSelected}$	If no valid object is selected	No selection, no objects in scene

Test Cases:

Equ. Class Combos	Input	Expected Output
$EC_{col,valid}$+ $EC_{obj,\ selected}$	red=0.5, green=0.5, blue=0.5, obj=mesh1	Mesh1 color changes to 0.5,0.5,0.5
$EC_{col,\ error}$ + $EC_{obj,selected}$	red=0.5, green=0.5, blue=-0.5, obj=mesh1	Error
$EC_{col,valid}$+ $EC_{obj,\ notSelected}$	red=0.5, green=0.5, blue=0.5, obj=not selected	No change
$EC_{col,error}$+ $EC_{obj,\ notSelected}$	red=0.5, green=0.5, blue=-0.5, obj=not selected	Error

Module: Shade Selection

Function: If a shading option is selected, determine whether or not there is a mesh selected, and if there is, change that meshes material properties

Equ. Class	Description	Possible Values
$EC_{wireframe,Change}$	Changes meshes material property to wireframe	Clicks Wireframe Button
$EC_{flatShading,Change}$	Changes meshes material property to flatShading	Clicks Flat Button
$EC_{smoothShading,Change}$	Changes meshes material property to smoothShading	Clicks Smooth Button

Test Cases:

Equ. Class Combos	Input	Expected Output
$EC_{wireframe,Change}$ + $EC_{flatShading,Change}$	Wireframe button click + flat button click	Mesh's shade changes from wireframe to flat
$EC_{wireframe,Change}$ + $EC_{smoothShading,Change}$	Wireframe button click + smooth button click	Mesh's shade changes from wireframe to smooth
$EC_{flatShading,Change}$ + $EC_{smoothShading,Change}$	Flat button click + smooth button click	Mesh's shade changes from flat to smooth
$EC_{flatShading,Change}$ + $EC_{wireframe,Change}$	Flat button click + wireframe button click	Mesh's shade changes from flat to wireframe
$EC_{smoothShading,Change}$ + $EC_{flatShading,Change}$	Smooth button click + flat button click	Mesh's shade changes from smooth to flat
$EC_{smoothShading,Change}$ + $EC_{wireframe,Change}$	Smooth button click + wireframe button click	Mesh's shade changes from smooth to wireframe

Module: Intensity

Function: If the user clicks and drags, check to see if the action is done at the proper location. If it is, change the intensity of the currently selected light.

Equ. Class	Description	Possible Values
$EC_{intensity,null}$	There is no intensity slider because no light has been selected	Light doesn't exist Light exists but is not selected Selected model exists but it is not a light
$EC_{intensity,1}$	Intensity slider exists	Light is selected
$EC_{intensity,valid}$	Valid inputs that trigger a change in a light's intensity	Click and drag on intensity slider
$EC_{intensity,invalid}$	Invalid inputs that do not trigger a change in a light's intensity	Click and drag outside the intensity slider Click off the light to deselect it

Test Cases:

Equ. Class Combos	Input	Expected Output
$EC_{intensity,null}$ + $EC_{intensity,valid}$	Light doesn't exist + Click and drag on intensity slider	Impossible action - No change in light's intensity
$EC_{intensity,null}$ + $EC_{intensity,invalid}$	Light exists but is not selected + Click and drag outside the intensity slider	No change in light's intensity
$EC_{intensity,1}$ + $EC_{intensity,valid}$	Light is selected + Click and drag on intensity slider	Selected light's intensity changes
$EC_{intensity,1}$ + $EC_{intensity,invalid}$	Light is selected + Click off the light to deselect it	No change in light's intensity

Module: Object List

Function: If there is at least one mesh in the object list, when the user clicks the object tag, determine whether the object is selected or not. If selected, the transformation mechanism on the object will be visible.

Equ. Class	Description	Possible Values
$EC_{\text{object list,empty}}$	There is no object in the object list. The user has not added mesh yet.	Empty object list
$EC_{\text{object list,non-empty}}$	There is at least one object in the object list.	A list with object name(s) Click object tag
$EC_{\text{object list,click object tag}}$	There is at least one object tag and click on it.	The object is selected Transformation UI appears on the object
$EC_{\text{object list,click name tag}}$	Click on the name tag on the right side of object tag.	"Type mesh name" on name tag
$EC_{\text{object list,define name}}$	Type the name in the name tag	Defined name on name tag

Test Cases:

Equ. Class Combos	Input	Expected Output
$EC_{\text{object list,empty}}$ + $EC_{\text{object list,click object tag}}$	Empty object list + Click on the object tag	Impossible action - No object tag in the list
$EC_{\text{object list,non-empty}}$ + $EC_{\text{object list,click object tag}}$	Non empty object list + Click on the object tag	Object is selected, and transformation UI appears on the object.
$EC_{\text{object list,non-empty}}$ + $EC_{\text{object list,click name tag}}$	Non empty object list + Click on and off the name tag	The name tag does not change.

$EC_{\text{object list,non-empty}}$ + $EC_{\text{object list,define name}}$	Non empty object list + Type the name in the name tag	Input name in the name tag