EECS 351 - Project B

SpongeBob SquarePants Toy

Hangbin Li Net ID: HLL932

Goals

The goal is to show different kinds of drawing and animating methods using WebGL language in 3D space. In my implementation, I will implement drawing and animating, different views control, roll, pitch, yaw control and quaternion control, etc.

User Guide

'/' Key: toggle projection.
F1 / Esc: show / hide help.
Left / Right Arrow Key: y-axis rotate of last view.
Up / Down Arrow Keys: x-axis rotate of last view.
[ ]: adjust nose length
A / S / D / W: move camera in plain.
F / R: move camera in altitude.

Code Guide

HangbinLi_ProjB.html
It mainly contains definition for 3 parts.

1. The canvas for drawing for WebGL content.
2. Interactive controls. There are several html input controls, such as buttons, range inside.
3. The area for help content.

Thera are some css code inside this file helped for formatting. No javascript code inside this file.

HangbinLi_ProjB_main.js
It contains GLSL code and javascript code that calls to draw and animate the picture, as well as javascript functions interact with html page and keyboard and mouse. The functions are:

1. main() - It initiates vertex and fragment shader then call to draw the object. It contains listener for mouse and keyboard event.
2. initVertexBuffers() - Initiates VBO.
3. draw() - Draw all views.
4. drawMyScene() - Draw all elements in one view.
5. animate() - Control the animation rules.
6. keydown() - Handle the action of key press event.
7. writeHelp2Html() - Write help content to webpage when needed.
8. winResize() - When the webpage is resized, the canvas should do the same.

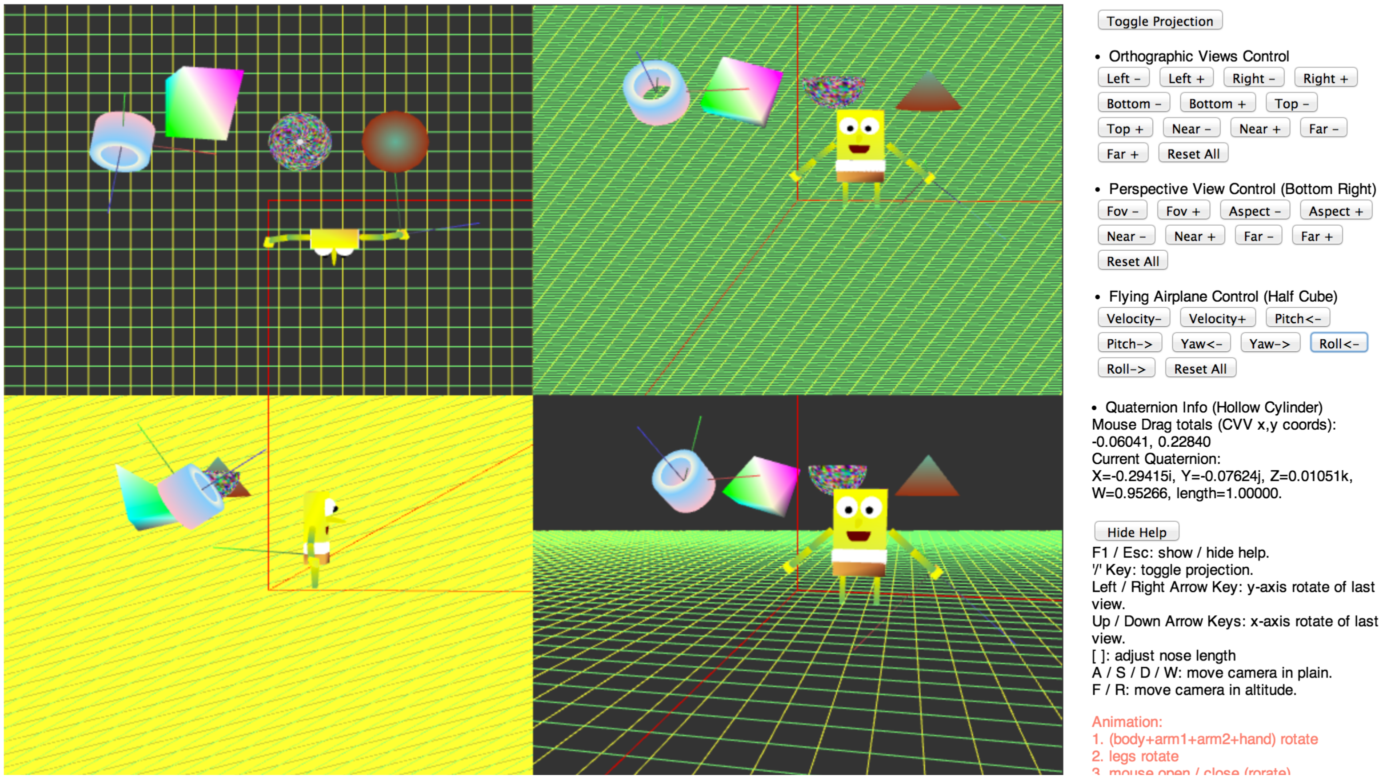
HangbinLi_ProjB_makeshape.js
Functions that make all kinds of shapes that is used in the other javascript file.

HangbinLi_ProjB_quaternion.js
I want to keep the code inside one file to be not too long, which can be hard to navigate when coding, so I put the functions that calculate the quaternion into one file.

Picture Illustration

The initial status of animation. All moving elements are showing in the help content above. Help content can be displayed or hidden by clicking button or keyboard shortcut keys.

The main interface


Toggle Projection
Left – Left + Right – Right +
Bottom – Bottom + Top –
Top + Near – Near + Far –
Far + Reset All
Fov – Fov + Aspect – Aspect +
Near – Near + Far – Far +
Reset All
Velocity– Velocity+ Pitch<–
Pitch–> Yaw<– Yaw–> Roll<–
Roll–> Reset All
Hide Help

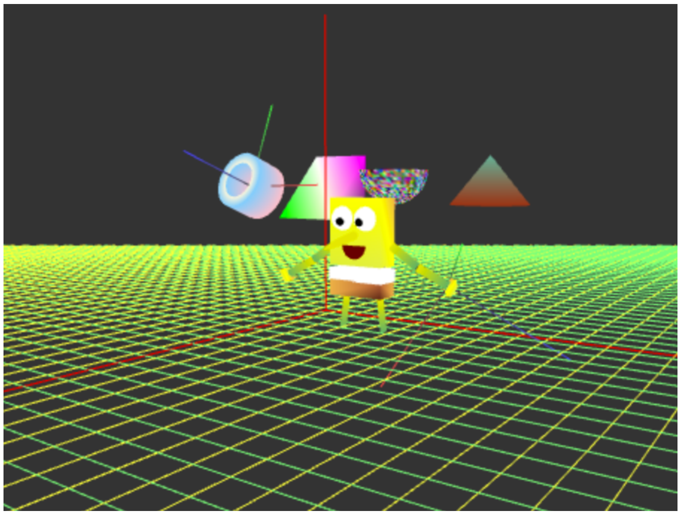
- Orthographic Views Control
- Perspective View Control (Bottom Right)
- Flying Airplane Control (Half Cube)
- Quaternion Info (Hollow Cylinder)

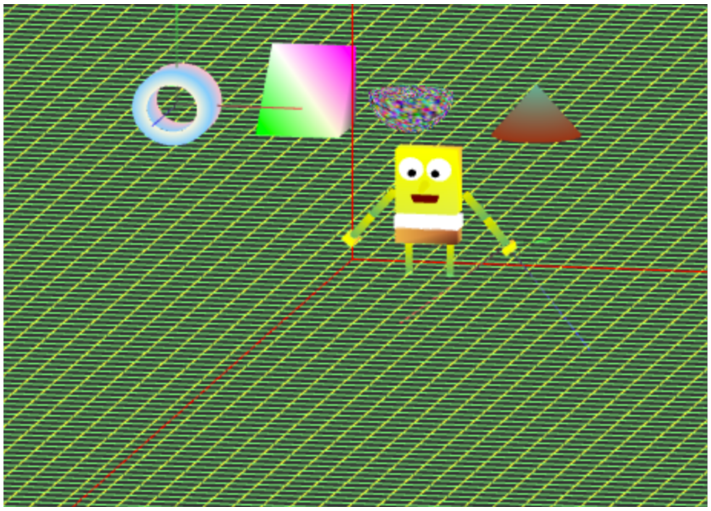
Mouse Drag totals (CVV x,y coords):
-0.06041, 0.22840
Current Quaternion:
X=-0.29415i, Y=-0.07624j, Z=0.01051k,
W=0.95266, length=1.00000.

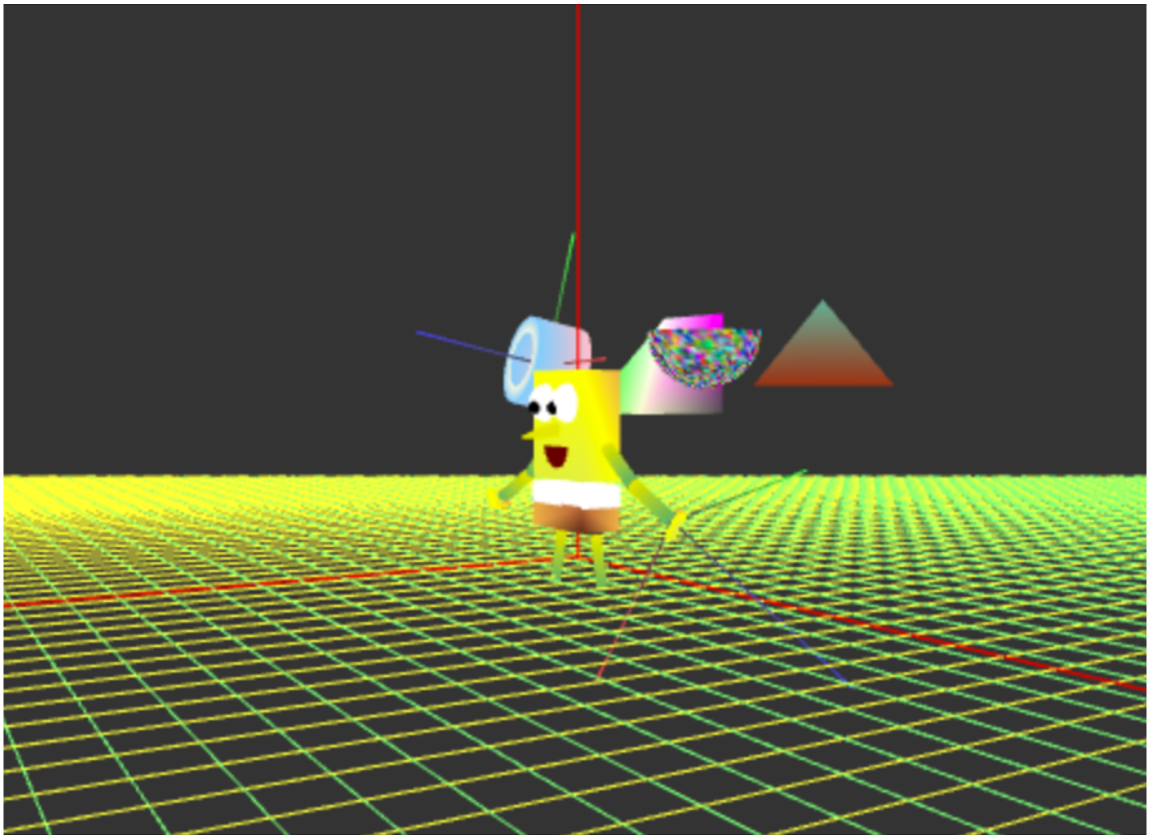
F1 / Esc: show / hide help.
'/' Key: toggle projection.
Left / Right Arrow Key: y-axis rotate of last view.
Up / Down Arrow Keys: x-axis rotate of last view.
[ ]: adjust nose length
A / S / D / W: move camera in plain.
F / R: move camera in altitude.

Animation:
1. (body+arm1+arm2+hand) rotate
2. legs rotate
3. mouse open / close (rorate)

Animation includes:
1. arm 3-joints (body+arm1+arm2+hand) & legs rotate
2. toy mouse open / close (rorate)
3. toy nose swing and length manipulation
4. eyeball (black) rotate



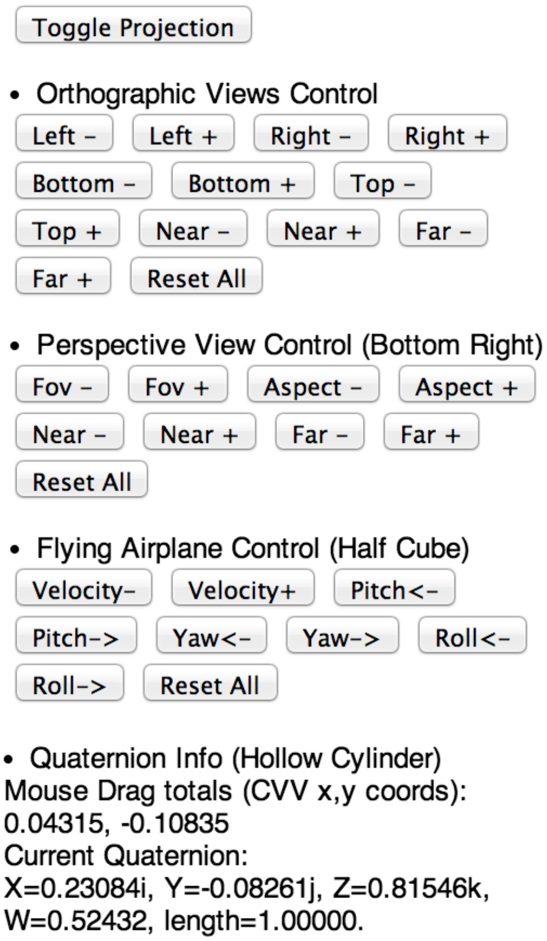




The camera type can be changed.

Viewing angle and viewing position can all be changed in all dimensions.

You can move eye point and direction to any point in the space. And it’s direct movement towards the position, not following axis fold lines.



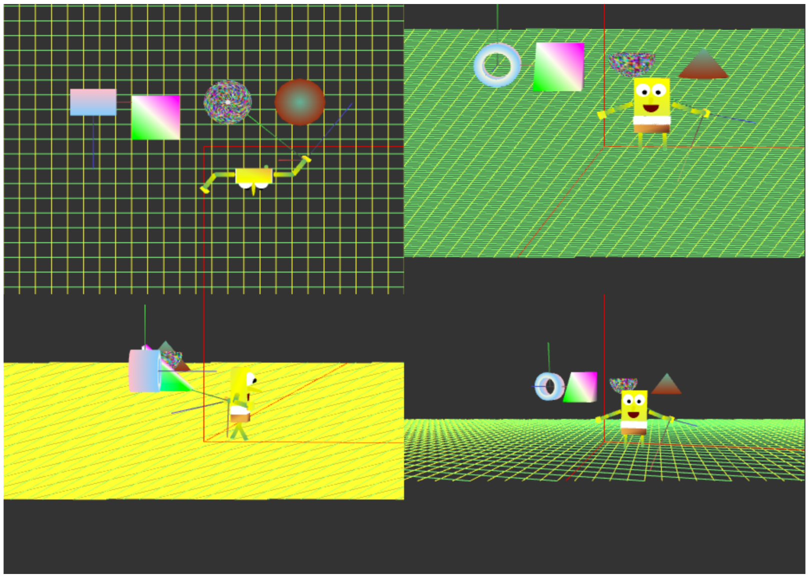
- Orthographic Views Control

- Perspective View Control (Bottom Right)

- Flying Airplane Control (Half Cube)

- Quaternion Info (Hollow Cylinder)

Mouse Drag totals (CVV x,y coords):
0.04315, -0.10835
Current Quaternion:
X=0.23084i, Y=-0.08261j, Z=0.81546k, W=0.52432, length=1.00000.



Extra 1. Orthographic and Perspective View Controls (6 parameters)

Extra 3. Flying Airplane Control (half cube, back 2nd left)

Extra 4. Quaternion Control (hollow cylinder, back 1st left)

Reference

1. In order to save a little time, I started my code from professor’s source files as references:
   BasicShapes.js - for drawing simple shapes
   JTLookAtSceneWithKeys_ViewVolume.js - for drawing different scenes
   ControlQuaternion.js - for making quaternion
   HelloCube_Resize.js - for resize the canvas when the windows is resized.

2. SpongeBob SquarePants Cartoon is a famous cartoon figure, the reference images that I use are found using Google Image Search which spread across the Internet.