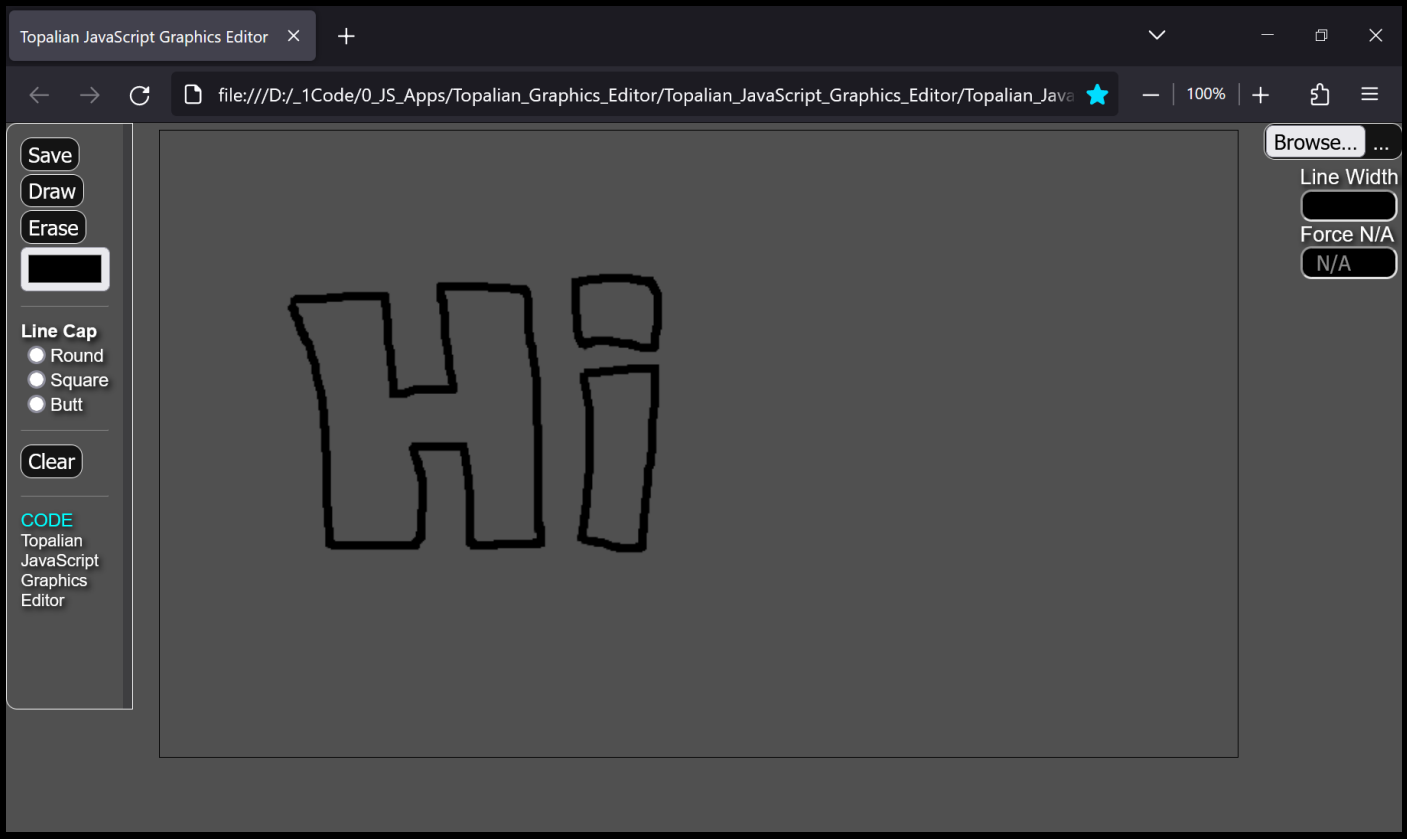
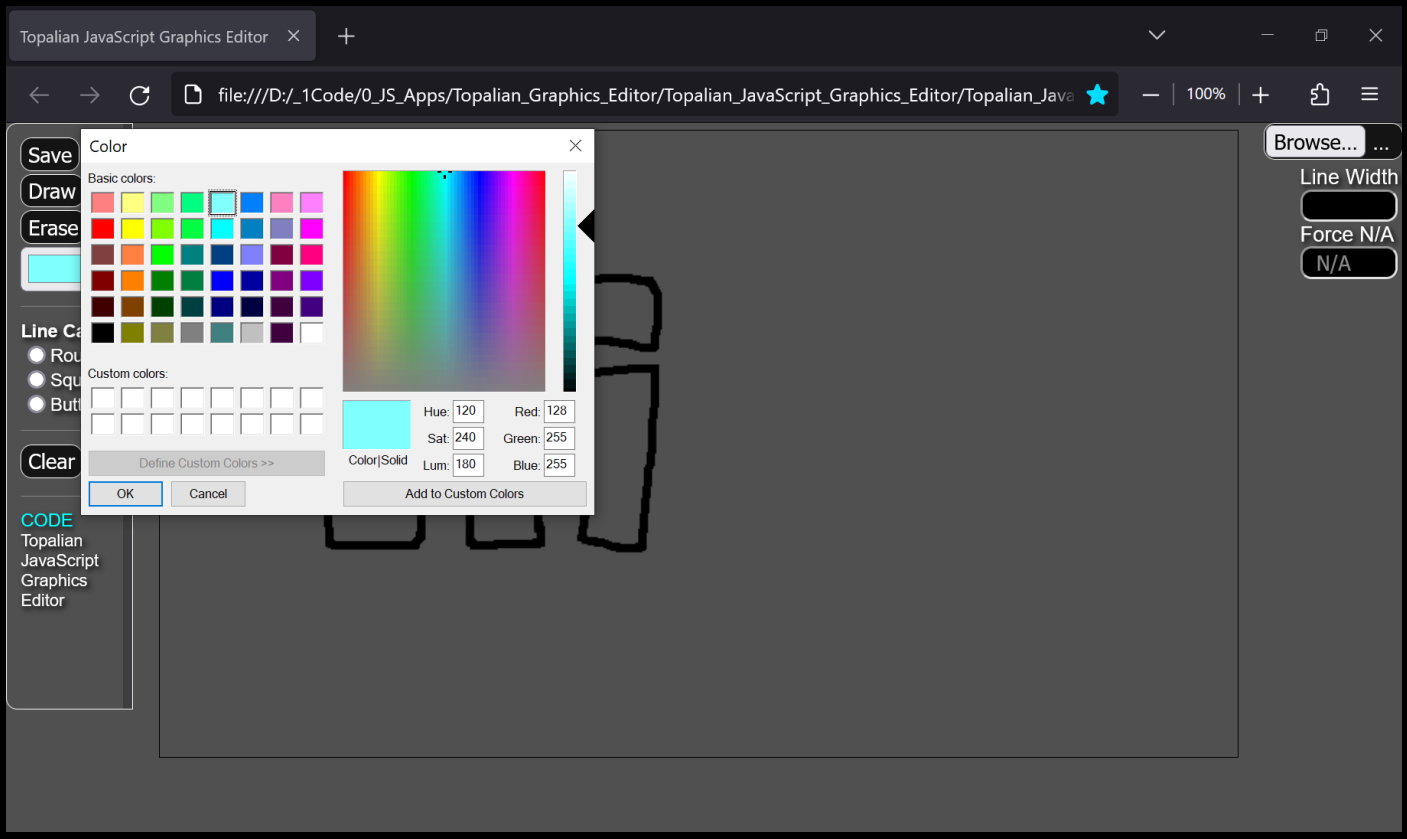
Topalian JavaScript Graphics Editor

by

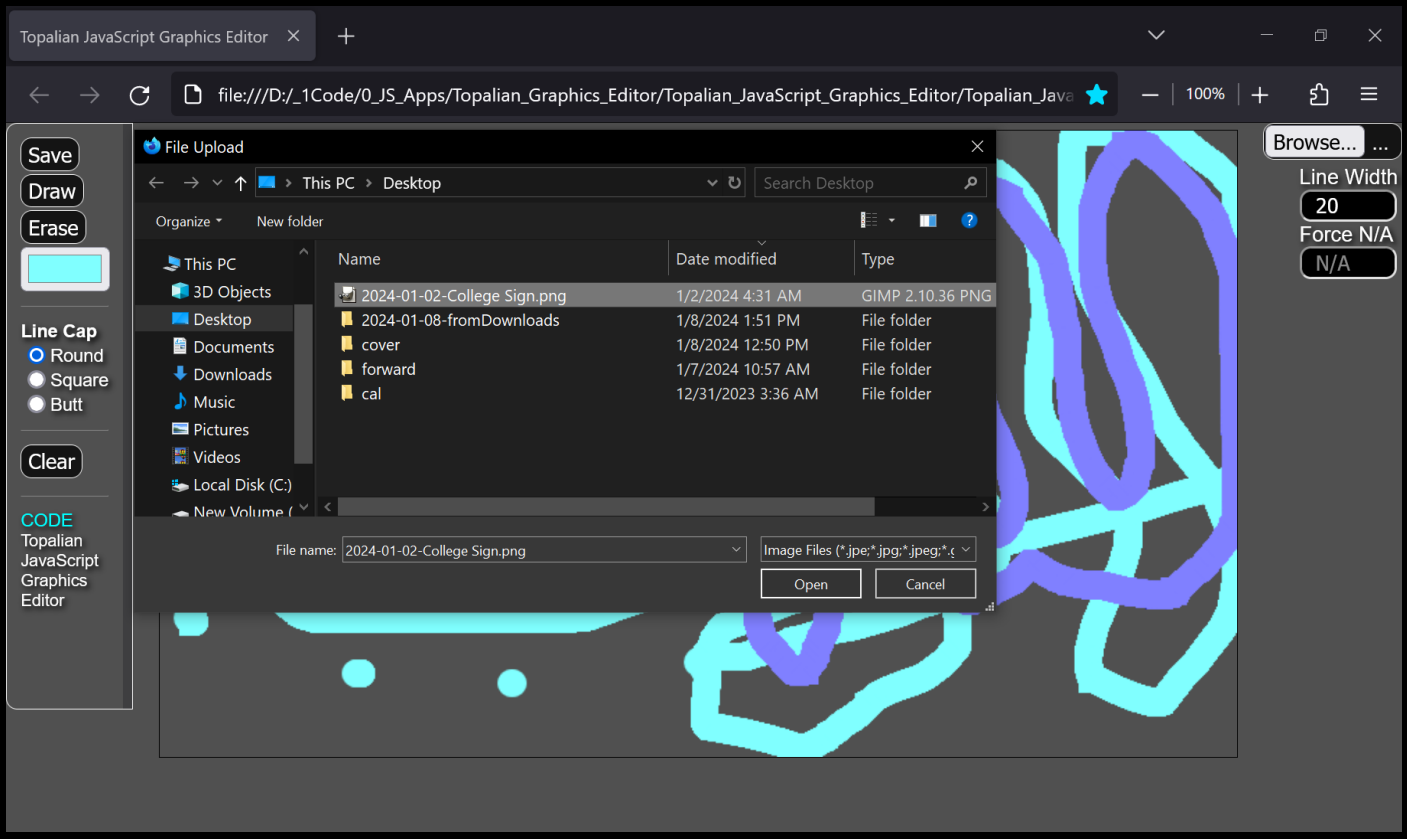
Christopher Andrew Topalian

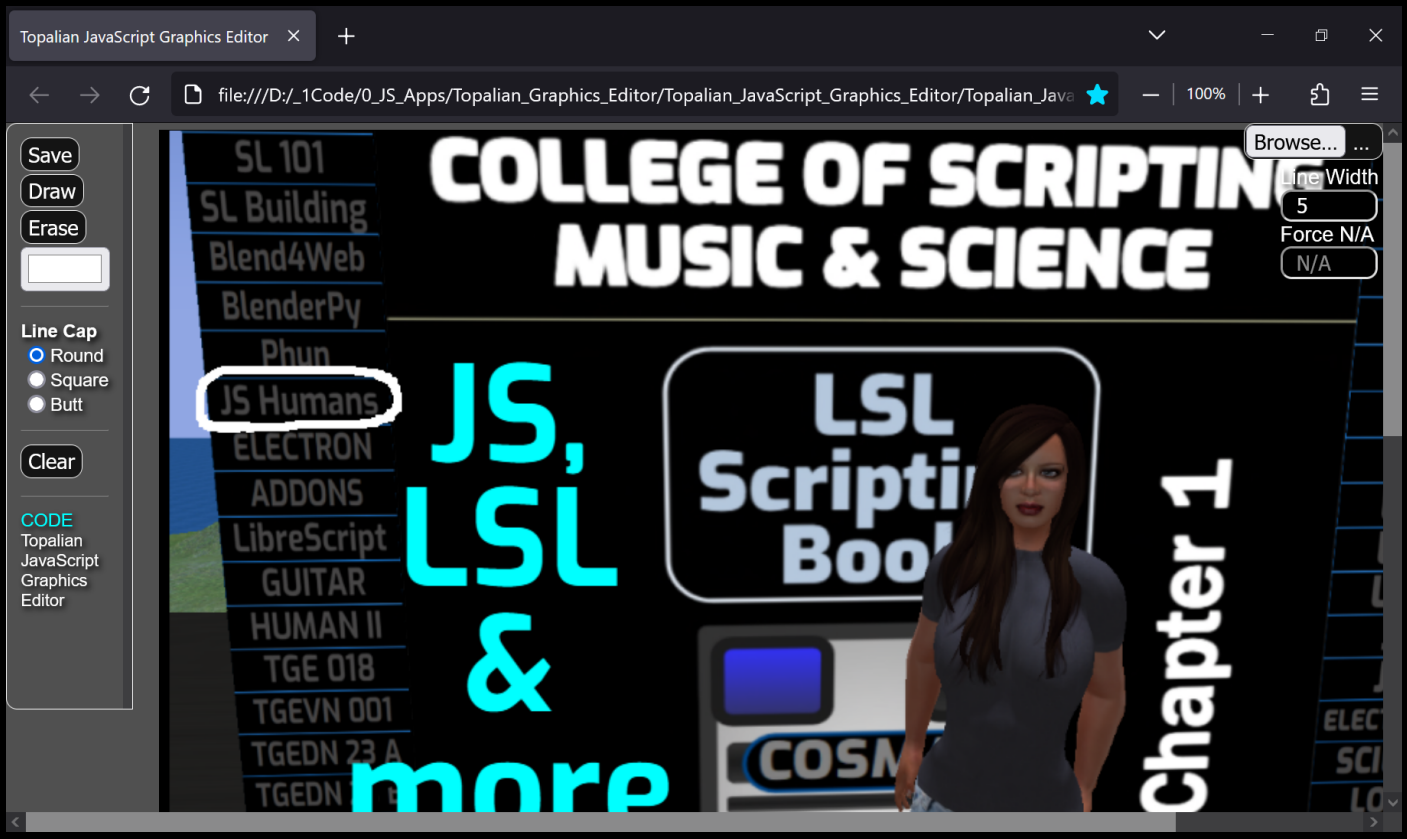
Dedicated to God the Father


Topalian JavaScript Graphics Editor
file:///D:/_1Code/0_JS_Apps/Topalian_Graphics_Editor/Topalian_JavaScript_Graphics_Editor/Topalian_Java
100%
Save
Draw
Erase
Line Cap
Round
Square
Butt
Clear
CODE
Topalian JavaScript Graphics Editor
Hi
Browse...
Line Width
Force N/A
N/A


Topalian JavaScript Graphics Editor
file:///D:/_1Code/0_JS_Apps/Topalian_Graphics_Editor/Topalian_JavaScript_Graphics_Editor/Topalian_Java
100%
Save
Draw
Erase
Clear
CODE
Topalian
JavaScript
Graphics
Editor
Color
Basic colors:
Custom colors:
Define Custom Colors >>
OK
Cancel
Color|Solid
Hue: 120
Sat: 240
Lum: 180
Red: 128
Green: 255
Blue: 255
Add to Custom Colors
Browse...
Line Width
Force N/A
N/A


Topalian JavaScript Graphics Editor
file:///D:/_1Code/0_JS_Apps/Topalian_Graphics_Editor/Topalian_JavaScript_Graphics_Editor/Topalian_Java
100%
Save
Draw
Erase
Line Cap
Round
Square
Butt
Clear
CODE
Topalian
JavaScript
Graphics
Editor
Hi
Browse...
Line Width
20
Force N/A
N/A


Topalian JavaScript Graphics Editor
file:///D:/_1Code/0_JS_Apps/Topalian_Graphics_Editor/Topalian_JavaScript_Graphics_Editor/Topalian_Java
100%
Save
Draw
Erase
Line Cap
Round
Square
Butt
Clear
CODE
Topalian JavaScript Graphics Editor
Browse...
Line Width
20
Force N/A
N/A
File Upload
This PC
Desktop
Search Desktop
Organize
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Name
Date modified
Type
2024-01-02-College Sign.png
1/2/2024 4:31 AM
GIMP 2.10.36 PNG
2024-01-08-fromDownloads
1/8/2024 1:51 PM
File folder
cover
1/8/2024 12:50 PM
File folder
forward
1/7/2024 10:57 AM
File folder
cal
12/31/2023 3:36 AM
File folder
File name:
2024-01-02-College Sign.png
Image Files (*.jpe;*.jpg;*.jpeg;*.
Open
Cancel


Topalian JavaScript Graphics Editor
file:///D:/_1Code/0_JS_Apps/Topalian_Graphics_Editor/Topalian_JavaScript_Graphics_Editor/Topalian_Java
100%
Save
Draw
Erase
Line Cap
Round
Square
Butt
Clear
CODE
Topalian JavaScript Graphics Editor
Browse...
Line Width
5
Force N/A
N/A
COLLEGE OF SCRIPTING
MUSIC & SCIENCE
SL 101
SL Building
Blend4Web
BlenderPy
Phun
JS Humans
ELECTRON
ADDONS
LibreScript
GUITAR
HUMAN II
TGE 018
TGEVN 001
TGEDN 23 A
JS, LSL & more
LSL Scripting Book
Chapter 1

```
<!-- Dedicated to God the Father -->

<!-- All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024 -->

<!-- https://github.com/ChristopherTopalian --
>

<!--
https://github.com/ChristopherAndrewTopalia
n -->

<!--
Topalian_JavaScript_Graphics_Editor.html --
>

<!-- Version 001 - (2024-01-09) -->
```

```
<html>
<head>
<title> Topalian JavaScript Graphics Editor
</title>

<link rel = 'stylesheet' href =
'css/style001.css'>

<script>

function ge(whichId)
{
    let result =
document.getElementById(whichId);

    return result;
}
```

```
function ce(whichType)
{
    let result =
document.createElement(whichType);

    return result;
}

function ba(whichElement)
{
    let result =
document.body.append(whichElement);

    return result;
}

function menuBrushOptions()
{
```

```
if (ge('mainLineWidthDiv'))
{
    ge('mainLineWidthDiv').remove();
}

let mainDiv = ce('div');
mainDiv.id = 'mainLineWidthDiv';
mainDiv.style.position = 'fixed';
mainDiv.style.right = '3px';
mainDiv.style.top = '30px';
ba(mainDiv);

//-//

let brushOptions = ce('div');
mainDiv.append(brushOptions);

//-//
```

```
    let lineWidthText = ce('span');
    lineWidthText.innerHTML = 'Line Width';
    lineWidthText.style.fontSize = '15px';
    brushOptions.append(lineWidthText);

    //-//

    let linebreakUnderLineWidthText = ce('br');

brushOptions.append(linebreakUnderLineWi
dthText);

    //-//

    let lineWidthTextbox = ce('input');
    lineWidthTextbox.type = 'text';
    lineWidthTextbox.id = 'lineWidthTextbox';
```

```
    lineWidthTextbox.className =
'inputStyle001';
    lineWidthTextbox.onkeyup = function()
    {
        lineWidth = lineWidthTextbox.value;
    };
    brushOptions.append(lineWidthTextbox);

    //-//

    let brushForceLabel = ce('div');
    brushForceLabel.innerHTML = 'Force N/A';
    brushForceLabel.id = 'brushForceLabel';
    brushForceLabel.style.fontSize = '15px';
    brushOptions.append(brushForceLabel);

    //-//
```

```
    let brushForceTextbox = ce("input");
    brushForceTextbox.type = "text";
    brushForceTextbox.id =
"brushForceTextbox";
    brushForceTextbox.className =
'inputStyle001';
    brushForceTextbox.title =
"brushForceTextbox";
    brushForceTextbox.placeholder = 'N/A';
    brushForceTextbox.onkeyup = function()
    {
        brushForce = brushForceTextbox.value;
    };
    brushOptions.append(brushForceTextbox);

    //-//

    brushOptions.append(ce('br'));
```

```
}

function updateLineCap()
{
    if (ge('roundCap').checked)
    {
        ctx.lineCap = 'round';
    }
    else if (ge('squareCap').checked)
    {
        ctx.lineCap = 'square';
    }
    else if (ge('buttCap').checked)
    {
        ctx.lineCap = 'butt';
    }
}
```

```
function clearCanvas()
{
    ctx.clearRect(0, 0, canvas.width,
canvas.height);
}

function toolboxCreate()
{
    let mainDivToolBox = ce('div');
    mainDivToolBox.id = 'toolbox';
    mainDivToolBox.className =
'mainDivToolBox';
    ba(mainDivToolBox);

    //-//

    // saveButton
    let saveButton = ce('button');
```

```
    saveButton.textContent = 'Save';
    saveButton.id = 'saveButton';
    saveButton.className = 'toolboxButton';
    saveButton.onclick = function()
    {
        saveDrawing();
    };
    mainDivToolBox.append(saveButton);

    //-//

    let linebreakUnderSaveButton = ce('br');

mainDivToolBox.append(linebreakUnderSave
Button);

    //-//
```

```
    // drawButton
    let drawButton = ce('button');
    drawButton.textContent = 'Draw';
    drawButton.id = 'drawButton';
    drawButton.className = 'toolboxButton';
    drawButton.onclick = function()
    {
        selectTool('draw');
    };
    mainDivToolBox.append(drawButton);

    //-//

    let linebreakUnderDrawButton = ce('br');

mainDivToolBox.append(linebreakUnderDraw
Button);
```

```
//-//

// eraseButton
let eraseButton = ce('button');
eraseButton.textContent = 'Erase';
eraseButton.id = 'eraseButton';
eraseButton.className = 'toolboxButton';
eraseButton.onclick = function()
{
    selectTool('erase');
};
mainDivToolBox.append(eraseButton);

//-//

let linebreakUnderEraseButton = ce('br');
```

```
mainDivToolBox.append(linebreakUnderEraseButton);

    //-//

    let colorChooser = ce('input');
    colorChooser.id = 'colorChooser';
    colorChooser.type = 'color';
    colorChooser.oninput = function()
    {
        lineColor = colorChooser.value;
    };
    mainDivToolBox.append(colorChooser);

    //-//

    mainDivToolBox.append(ce('hr'));
```

```
    //-//

    let lineCapContainer = ce('div');

mainDivToolBox.append(lineCapContainer);

    //-//

    let capText = ce('label');
    capText.textContent = 'Line Cap';
    capText.style.fontSize = '13px';
    capText.style.fontWeight = 'bold';
    lineCapContainer.append(capText);

    //-//

    lineCapContainer.append(ce('br'));
```

```
    //-//

    let roundCapCheckbox = ce('input');
    roundCapCheckbox.type = 'radio';
    roundCapCheckbox.id = 'roundCap';
    roundCapCheckbox.name =
'lineCapChoice';
    roundCapCheckbox.onchange = function()
    {
        updateLineCap();
    };

lineCapContainer.append(roundCapCheckbo
x);

    //-//
```

```
    let labelRound = ce('label');
    labelRound.id = 'labelRound';
    labelRound.textContent = 'Round';
    labelRound.className = 'labels';
    lineCapContainer.append(labelRound);

    //-//

    lineCapContainer.append(ce('br'));

    //-//

    let squareCapCheckbox = ce('input');
    squareCapCheckbox.type = 'radio';
    squareCapCheckbox.textContent =
'Square';
    squareCapCheckbox.id = 'squareCap';
```

```
    squareCapCheckbox.name =
'lineCapChoice';
    squareCapCheckbox.className = 'labels';
    squareCapCheckbox.onchange = function()
    {
        updateLineCap();
    };

lineCapContainer.append(squareCapCheckb
ox);

    //-//

    let labelSquare = ce('label');
    labelSquare.id = 'labelSquare';
    labelSquare.textContent = 'Square';
    labelSquare.className = 'labels';
    lineCapContainer.append(labelSquare);
```

```
//-//

lineCapContainer.append(ce('br'));

//-//

let buttCapCheckbox = ce('input');
buttCapCheckbox.type = 'radio';
buttCapCheckbox.textContent = 'Butt';
buttCapCheckbox.id = 'buttCap';
buttCapCheckbox.name = 'lineCapChoice';
buttCapCheckbox.className = 'labels';
buttCapCheckbox.onchange = function()
{
    updateLineCap();
};
```

```
lineCapContainer.append(buttCapCheckbox);

    //-//

    let labelButt = ce("label");
    labelButt.textContent = 'Butt';
    labelButt.id = 'labelButt';
    labelButt.className = 'labels';
    lineCapContainer.append(labelButt);

    //-//

    lineCapContainer.append(ce('hr'));

    //-//

    // clear button
```

```
let clearButton = ce('button');
clearButton.textContent = 'Clear';
clearButton.className = 'toolboxButton';
clearButton.onclick = function()
{
    let choice = confirm('Erase All');

    if (choice)
    {
        clearCanvas();
    }
    else
    {
        return;
    }
};

mainDivToolBox.append(clearButton);
```

```
    //-//

    mainDivToolBox.append(ce('hr'));

    //-//

    let githubLink = ce('a');
    githubLink.textContent = 'CODE';
    githubLink.href =
'https://github.com/ChristopherTopalian/Topal
ian_JavaScript_Graphics_Editor';
    githubLink.target = '_blank';
    githubLink.style.fontSize = 13 + 'px';
    mainDivToolBox.append(githubLink);

    //-//
```

```
    let info = ce('div');
    info.textContent = 'Topalian JavaScript
Graphics Editor';
    info.style.fontSize = 12 + 'px';
    mainDivToolBox.append(info);
}

function whenLoaded()
{
    toolboxCreate();
    menuBrushOptions();
}

</script>

</head>

<body onload = 'whenLoaded();'>
```

```
<input type = 'file' id = 'imageInput' class =
'imageInput' accept = 'image/*'>

<canvas id = 'drawingCanvas'></canvas>

<script>

let canvas = ge('drawingCanvas');

canvas.width = 775;
canvas.height = 450;

let ctx = canvas.getContext('2d');

let lineWidth = 5;

let brushForce = 1.0;
```

```
let lineColor = `rgba(0, 0, 0, ${brushForce})`;

let colorPicker = ge('colorPicker');

let paths = [];

let selectedTool = 'draw';

let drawing = false;

let erasing = false;

let lastPos;

canvas.addEventListener('mousedown',
handleMouseDown);
```

```
canvas.addEventListener('mouseup',
handleMouseUp);

canvas.addEventListener('mousemove',
handleMouseMove);

function handleImageUpload(event)
{
    let file = event.target.files[0];

    if (file)
    {
        let reader = new FileReader();

        reader.onload = function(e)
        {
            let img = new Image();
```

```
            img.onload = function()
            {
                let canvas = ge('drawingCanvas');

                canvas.width = img.naturalWidth;

                canvas.height = img.naturalHeight;

                ctx.clearRect(0, 0, canvas.width,
canvas.height);

                ctx.drawImage(img, 0, 0,
canvas.width, canvas.height);
            };

            img.src = e.target.result;
        };
```

```
        reader.readAsDataURL(file);
    }
}

let imageInput = ge('imageInput');

if (imageInput)
{
    imageInput.addEventListener('change',
handleImageUpload);
}
else
{
    console.error("Element with id 'imageInput'
not found.");
}

function handleMouseDown(e)
```

```
{
    if (selectedTool === 'draw')
    {
        drawing = true;

        startDrawing(e);
    }
    else if (selectedTool === 'erase')
    {
        erasing = true;

        startErasing(e);
    }
}

function handleMouseUp()
{
    if (drawing)
```

```
    {
        stopDrawing();
    }
    else if (erasing)
    {
        erasing = false;

        stopErasing();
    }
}

function handleMouseMove(e)
{
    if (drawing)
    {
        draw(e);
    }
    else if (erasing)
```

```
    {
        erasePath(e);
    }
}

function startDrawing(e)
{
    lastPos = getMousePos(e);

    ctx.beginPath();

    ctx.moveTo(lastPos.x, lastPos.y);
}

function stopDrawing()
{
    drawing = false;
```

```
    paths.push(ctx.getImageData(0, 0,
canvas.width, canvas.height));
}

function draw(e)
{
    if (!drawing)
    {
        return;
    }

    let currentPos = getMousePos(e);

    ctx.lineTo(currentPos.x, currentPos.y);

    ctx.strokeStyle = lineColor;

    ctx.lineWidth = lineWidth;
```

```
    ctx.stroke();
}

function startErasing(e)
{
    if (!erasing)
    {
        return;
    }

    lastPos = getMousePos(e);

    ctx.beginPath();

    ctx.moveTo(lastPos.x, lastPos.y);
}
```

```
function stopErasing()
{
    if (!erasing)
    {
        return;
    }

    ctx.closePath();
}

function erasePath(e)
{
    if (!erasing)
    {
        return;
    }

    let currentPos = getMousePos(e);
```

```
    ctx.clearRect(currentPos.x - lineWidth / 2,
currentPos.y - lineWidth / 2, lineWidth,
lineWidth);
}

function getMousePos(e)
{
    let rect = canvas.getBoundingClientRect();

    return {
        x: e.clientX - rect.left,
        y: e.clientY - rect.top
    };
}

function selectTool(whichTool)
{
```

```
    selectedTool = whichTool;

    drawing = false;
}

function saveDrawing()
{
    let dataUrl =
canvas.toDataURL('image/png');

    let link = ce('a');

    link.href = dataUrl;

    link.download = 'drawing.png';

    link.click();
}
```

```
</script>

</body>

</html>
```

```
/* Dedicated to God the Father */

/* All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024 */

/* https://github.com/ChristopherTopalian */

/*
https://github.com/ChristopherAndrewTopalia
n */

/* style001.css */

body
{
    background-color: rgb(80, 80, 80);
    color: rgb(255, 255, 255);
    text-shadow: 2px 2px 4px rgb(0, 0, 0);
```

```
}

canvas
{
    border-style: solid;
    border-width: 1px;
    border-color: rgb(0, 0, 0);
}

a
{
    text-decoration: none;
    color: rgb(0, 255, 255);
}

#drawingCanvas
{
    position: absolute;
```

```
    left: 110px;
    top: 5px;
}

#toolbox
{
    position: fixed;
    left: 0px;
    top: 0px;
    width:70px;
    height: 400px;
    padding: 10px;
    border-style: solid;
    border-width: 1px;
    border-color: rgb(255, 255, 255);
    border-radius: 8px;
    overflow-y: scroll;
    scrollbar-width: thin;
```

```
}

.toolboxButton
{
    padding-left: 5px;
    padding-right: 5px;
    padding-top: 3px;
    padding-bottom: 3px;
    margin-bottom: 2px;
    border-style: solid;
    border-width: 1px;
    border-radius: 8px;
    border-color: rgb(255, 255, 255);
    background-color: rgb(20, 20, 20);
    font-size: 15px;
    color: rgb(255, 255, 255);
    line-height: 17px;
}
```

```
.toolboxButton:hover
{
    border-color: aqua;
}

.toolboxButton:active
{
    color: aqua;
}

.labels
{
    font-size: 13px;
}

.inputStyle001
{
```

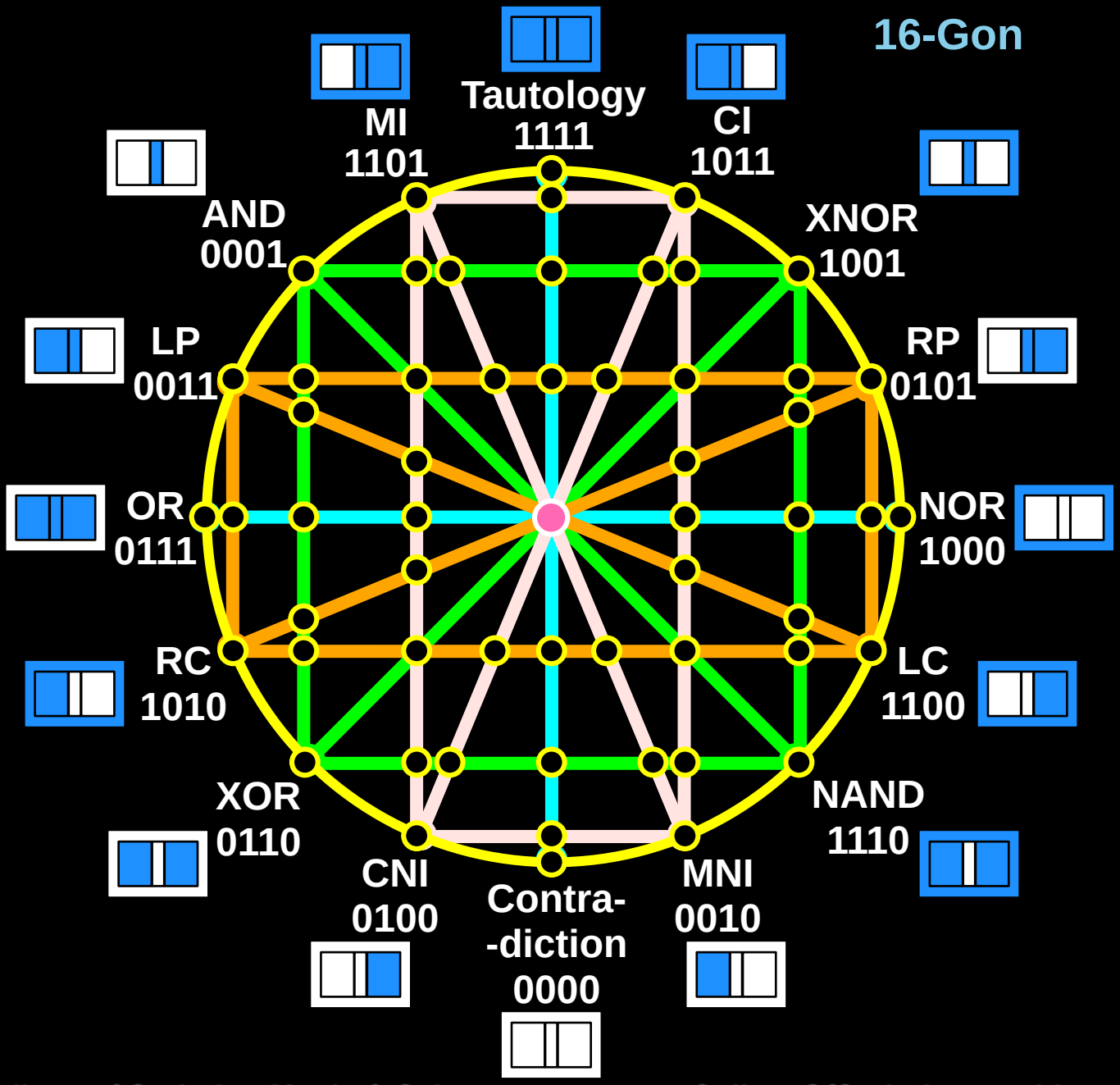
```
    width: 70px;
    padding-left: 10px;
    padding-right: 10px;
    border-radius: 8px;
    background-color: rgb(0, 0, 0);
    font-size: 15px;
    color: rgb(255, 255, 255);
}

.imageInput
{
    position: fixed;
    right: 0px;
    top: 0px;
    width: 100px;
    padding-left: 1px;
    padding-right: 1px;
    padding-top: 1px;
```

```
    padding-bottom: 1px;
    margin-bottom: 1px;
    border-style: solid;
    border-width: 1px;
    border-radius: 8px;
    border-color: rgb(255, 255, 255);
    background-color: rgb(20, 20, 20);
    font-size: 15px;
    color: rgb(255, 255, 255);
    line-height: 15px;
    z-index: 100;
}
```

Christopher Topalian

True Artificial Intelligence System

16-Gon



For More Tutorials:

CollegeOfScripting.weebly.com

CollegeOfScripting.wordpress.com

Youtube.com/ScriptingCollege

Twitter.com/CollegeOfScript

GitHub.com/ChristopherTopalian

GitHub.com/ChristopherAndrewTopalian

Sites.google.com/view/CollegeOfScripting

Dedicated to God the Father

This book is created by the
College of Scripting Music & Science.
Always remember, that each time you write a script with a pencil and paper, it becomes imprinted so deeply in memory that the material and methods are learned extremely well.

When you Type the scripts, the same is true. The more you type and write out the scripts by keyboard or pencil and paper, the more you will learn programming!

Write and Type every example that you find.
Keep all of your scripts organized.
Every script that you create increases your programming abilities.
SEEING CODE, is one thing,
but WRITING CODE is another.
Write it, Type it, Speak It, See It, Dream It.

CollegeOfScripting.weebly.com