# Department of Computing

# CS361: Computer Graphics

# Class: BSCS-5

# Lab03: Mouse and Keyboard events

# Date: 18th September, 2018

# Time: 9:00- 12:00

# Lab03: Mouse and Keyboard events

# Introduction

You can also detect a mouse click on your canvas. Again, this is done with addEventListener. There are quite a few mouse events you can detect: mousedown, mouseup, mousemove, mouseout and mouseover. As an example, here's some code that detects the mousedown event

**Objectives**

After performing this lab students should be able to:

Know the basics of HTML 5 Canvas, Under Stand Mouse and Keyboard events, Mirror Transformations

**Tools/Software Requirement**

For testing HTML,JS, Sublime. WebGL Lib.

**Description**

**Buttons in JS**

// 1. Create the button

var button = document.createElement("button");

button.innerHTML = "Do Something";

// 2. Append somewhere

var body = document.getElementsByTagName("body")[0];

body.appendChild(button);

// 3. Add event handler

button.addEventListener ("click", function() {

alert("did something");

});

/\* Read

**Example**

var Trangle;

‘’’’’’’’’’’’’’

document.getElementById("Triangle").onclick= function() {Triangle()};

var canvas = document.getElementById('webgl');

gl = getWebGLContext(canvas);

if (!gl){

console.log('Failed to find context');

}

canvas.onmousedown = function(ev) {click(ev, gl, canvas, a\_Position, a\_Size, u\_FragColor, tapCoordinates, tapColors, sizes );};

var program = initShaders( gl, "vertex-shader", "fragment-shader" );

gl.useProgram (program);

gl.program = program;

var a\_Position = gl.getAttribLocation(program, 'a\_Position');

if (a\_Position < 0) {

console.log ("Failed to Get Position");

return;

}

var a\_Size = gl.getAttribLocation(program, 'a\_Size');

if (a\_Size < 0) {

console.log ("Failed to Get Size");

return;

}

var u\_FragColor = gl.getUniformLocation(program, 'u\_FragColor');

if (u\_FragColor < 0) {

console.log ("Failed to Get Color");

return;

}

gl.clearColor(0.0, 0.0, 0.0, 1.0);

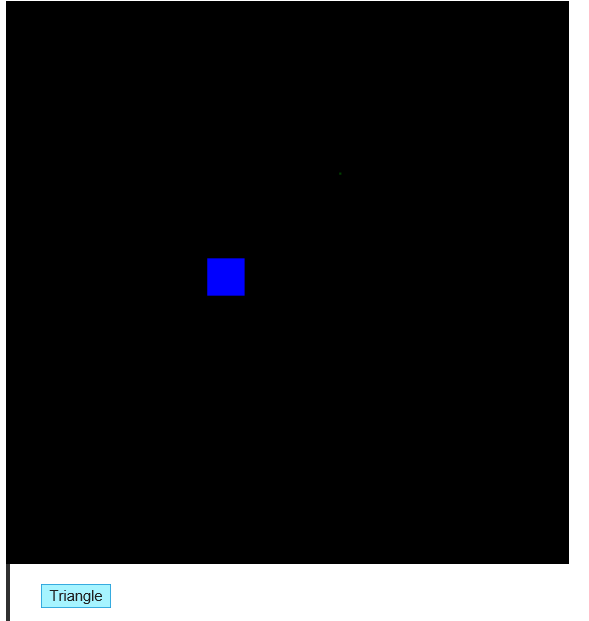
gl.clear(gl.COLOR\_BUFFER\_BIT);

}

function Triangle(){

,,,,,,,,,,,,,,,,,,,,,,

}



**Lab Task**

**​**  
Create three buttons in javascript with names:

1. Triangle

2. Quad

3. Colour

If user clicks the button titled "Triangle" then webGL draws a triangle on the canvas on mouse click using a predefined colour. (3.5)

If user presses the button titled "quad" webGL draws quads on mouse click using a predefined colour. (3.5)

Clicking the Colour button results in generation of a random color which is set as a "predefined colour" for drawing Triangle and Quads. (3)

**Deliverable**

Upload your full code with all the lib files.