```
// global variables
var canvas=null;
var gl=null; // webgl context
var bFullscreen=false;
var canvas_original_width;
var canvas_original_height;

// To start animation : To have requestAnimationFrame() to be called "cross-browser" compatible
var requestAnimationFrame =
window.requestAnimationFrame ||
window.webkitRequestAnimationFrame ||
window.mozRequestAnimationFrame ||
window.oRequestAnimationFrame ||
window.msRequestAnimationFrame;

// To stop animation : To have cancelAnimationFrame() to be called "cross-browser" compatible
var cancelAnimationFrame =
window.cancelAnimationFrame ||
window.webkitCancelRequestAnimationFrame || window.webkitCancelAnimationFrame ||
window.mozCancelRequestAnimationFrame || window.mozCancelAnimationFrame ||
window.oCancelRequestAnimationFrame || window.oCancelAnimationFrame ||
window.msCancelRequestAnimationFrame || window.msCancelAnimationFrame;

// onload function
function main()
{
    // get <canvas> element
    canvas = document.getElementById("AMC");
    if(!canvas)
        console.log("Obtaining Canvas Failed\n");
    else
        console.log("Obtaining Canvas Succeeded\n");
    canvas_original_width=canvas.width;
    canvas_original_height=canvas.height;

    // register keyboard's keydown event handler
    window.addEventListener("keydown", keyDown, false);
    window.addEventListener("click", mouseDown, false);
    window.addEventListener("resize", resize, false);

    // initialize WebGL
    init();

    // start drawing here as warming-up
    resize();
    draw();
}

function toggleFullScreen()
{
```

```
51      // code
52      var fullscreen_element =
53      document.fullscreenElement ||
54      document.webkitFullscreenElement ||
55      document.mozFullScreenElement ||
56      document.msFullscreenElement ||
57      null;
58
59      // if not fullscreen
60      if(fullscreen_element==null)
61      {
62          if(canvas.requestFullscreen)
63              canvas.requestFullscreen();
64          else if(canvas.mozRequestFullScreen)
65              canvas.mozRequestFullScreen();
66          else if(canvas.webkitRequestFullscreen)
67              canvas.webkitRequestFullscreen();
68          else if(canvas.msRequestFullscreen)
69              canvas.msRequestFullscreen();
70          bFullscreen=true;
71      }
72      else // if already fullscreen
73      {
74          if(document.exitFullscreen)
75              document.exitFullscreen();
76          else if(document.mozCancelFullScreen)
77              document.mozCancelFullScreen();
78          else if(document.webkitExitFullscreen)
79              document.webkitExitFullscreen();
80          else if(document.msExitFullscreen)
81              document.msExitFullscreen();
82          bFullscreen=false;
83      }
84  }
85
86  function init()
87  {
88      // code
89      // get WebGL 2.0 context
90      gl = canvas.getContext("webgl2");
91      if(gl==null) // failed to get context
92      {
93          console.log("Failed to get the rendering context for WebGL");
94          return;
95      }
96      gl.viewportWidth = canvas.width;
97      gl.viewportHeight = canvas.height;
98
99      // set clear color
100     gl.clearColor(0.0, 0.0, 1.0, 1.0); // blue
101 }
102
```

```
103 function resize()
104 {
105     // code
106     if(bFullscreen==true)
107     {
108         canvas.width=window.innerWidth;
109         canvas.height=window.innerHeight;
110     }
111     else
112     {
113         canvas.width=canvas_original_width;
114         canvas.height=canvas_original_height;
115     }
116
117     // set the viewport to match
118     gl.viewport(0, 0, canvas.width, canvas.height);
119 }
120
121 function draw()
122 {
123     // code
124     gl.clear(gl.COLOR_BUFFER_BIT);
125
126     // animation loop
127     requestAnimationFrame(draw, canvas);
128 }
129
130 function keyDown(event)
131 {
132     // code
133     switch(event.keyCode)
134     {
135         case 70: // for 'F' or 'f'
136             toggleFullScreen();
137             break;
138     }
139 }
140
141 function mouseDown()
142 {
143     // code
144 }
145
```