gl.vertexAttribPointer(positionLocation, 3, gl.FLOAT, false, 0, 0);

This line of code is used in WebGL to associate a specific vertex attribute, in this case, the vertex position attribute, with a corresponding buffer that contains the actual vertex data.

Let's break down the parameters of this function call:

- `positionLocation`: This parameter is the location of the vertex position attribute in the shader program.
- `3`: This parameter specifies the size of the attribute. In this case, the position attribute has three components (x, y, and z), so the size is 3.
- `gl.FLOAT`: This parameter specifies the data type of the attribute. In this case, the attribute is of type float.
- `false`: This parameter specifies whether the data should be normalized before being used. In this case, it is set to false.
- `0`: This parameter specifies the stride between each set of attribute data in the buffer. In this case, since the attribute data is tightly packed, the stride is 0.
- `0`: This parameter specifies the offset in bytes from the start of the buffer to the first element of the attribute. In this case, the offset is 0.

In summary, the `gl.vertexAttribPointer()` function is used to specify how WebGL should read the vertex attribute data from a buffer, so that it can be properly rendered on the screen.

gl.drawArrays(gl.LINES, 0, 3);

This line of code is used in WebGL to draw a set of primitives, specifically lines, using the current state of the WebGL context.

Let's break down the parameters of this function call:

- `gl.drawArrays()`: This function is used to draw a set of primitives based on the vertex data that has been specified previously.
- `gl.LINES`: This parameter specifies the type of primitive to be drawn. In this case, the primitive type is lines.
- `0`: This parameter specifies the starting index of the first vertex to be drawn.
- `3`: This parameter specifies the number of vertices to be used in the drawing operation. In this case, three vertices will be used to draw two lines.

So, this line of code is instructing WebGL to draw two lines using the first three vertices that have been specified in the vertex buffer. The specific appearance of the lines will depend on the values of the vertices and the WebGL context settings that have been specified.