Write a shader program that displays a color hexagon. The color changes gradually from red to blue and from blue to red repetitively.

For this, I modified the movingv.cpp program. I changed the triangle to a polygon and added more points to create a hexagon in the display function:

```c
void display(void)
{
    GLfloat vec[4];

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glClearColor( 1.0, 1.0, 1.0, 0.0 ); //get white background color
    glColor3f ( 1, 0, 0 ); //red, this will have no effect if shader is loaded

    glBegin ( GL_POLYGON );
    glVertexAttrib1f ( timeParam, 0.1 );
    glVertex3f ( -1, 0, 0 );
    glVertex3f ( -0.5, -1, 0 );
    glVertex3f ( 0.5, -1, 0 );
    glVertex3f ( 1, 0, 0 );
    glVertex3f ( 0.5, 1, 0 );
    glVertex3f ( -0.5, 1, 0 );
    glVertex3f ( -1, 0, 0 );
    glEnd();

    glutSwapBuffers();
    glFlush();
}
```
Then I changed the Vertex Shader:

```cpp
uniform float time;   //value provided by application program
varying vec4 Color;

void main(void)
{
    float s = 0;
    float t = 0;

    s = s + 2.0 * sin ( 0.002 * time );
    t = 0.5 - s;
    Color = vec4 ( s, 0, t, 1.0 );

    gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
}
```

And finally, the Fragment Shader:

```cpp
varying vec4 Color;

void main(void)
{
    gl_FragColor = Color;
}
```

Summary:
I completed all parts of the lab and am giving myself full credit.