CG PROJECTY ON COLOR CODE COMBINATION

//this program illustrates the concepts of color combination and usage of OpenGL library.

//to start program press "s"

//to emit red light press "r"

//to emit green light press "g"

//to emit blue light press "b"

```
#include<Windows.h>
#include<string.h>
#include<stdarg.h>
#include<stdio.h>
#include<glut.h>
static double x = 0.0;
static float red1 = 0.0;
static float green1 = 0.0;
static float blue1 = 0.0;
static float help1 = 1.0;
void stroke_output(GLfloat x, GLfloat y, const char* format, ...) {
	va_list args;
	char buffer[200], * p;
	va_start(args, format);
	vsprintf_s(buffer, format, args);
	va_end(args);
	glPushMatrix();
	glTranslatef(-2.5, y, 0);
```

```
	glScaled(0.003, 0.005, 0.005);
	for (p = buffer; *p; p++) {
		glutStrokeCharacter(GLUT_STROKE_ROMAN, *p);
	}
	glPopMatrix();
}
void doInit() {
	glClearColor(0.5, 0.5, 0.5, 0.0);
	glColor3f(0.0, 1.0, 1.0);
	glViewport(0, 0, 640, 480);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	gluPerspective(30.0f, (GLfloat)640 / (GLfloat)480, 0.1f, 200.0f);
	glMatrixMode(GL_MODELVIEW);
	glLoadIdentity();
	glClearDepth(2.0f);
	glEnable(GL_DEPTH_TEST);
	glEnable(GL_COLOR_MATERIAL);
	glDepthFunc(GL_LEQUAL);
}
void torch() {
	glPushMatrix();
	glutSolidSphere(0.8, 50, 50);
	glPopMatrix();

	glPushMatrix();
	glRotatef(90, 0, 1, 0);
```

```
	glScaled(0.5, 0.5, 3);
	glColor3f(0, 1, 1);
	glutSolidTorus(0.4, 1.5, 50, 50);
	glPopMatrix();

	glPushMatrix();
	glTranslatef(-1.5, 0, 0);
	glRotatef(90, 0, 1, 0);
	glScaled(0.7, 0.7, 1.5);
	glutSolidTorus(0.4, 1.5, 50, 50);
	glPopMatrix();
}
void help() {
	glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
	glPushMatrix();
	glScaled(0.7, 0.7, 0.7);
	stroke_output(-2, 3, "H -----> Toggle Help");
	stroke_output(-2, 2, "R -----> Toggle Red Light");
	stroke_output(-2, 1, "G -----> Toggle Green Light");
	stroke_output(-2, 0, "B -----> Toggle Blue Light");
	glPopMatrix();
	glFlush();
	glutSwapBuffers();
}
void draw() {
	glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
	glLoadIdentity();
```

```
glTranslatef(0.0f, 0.0f, -13.0f);
glPushMatrix();
glTranslatef(0, 0, -15);
glScaled(1, 1, 0.1);
glColor3f(0.3, 0.3, 0);
glutSolidCube(9);
glPopMatrix();

glPushMatrix();
glColor3f(1, 0, 1);
glutSolidCone(4.5, 15, 40, 40);
glPopMatrix();

glPushMatrix();
glTranslatef(0, 0, -14);
glScaled(1, 1, 0.1);
glColor3f(red1, green1, blue1);
glutSolidSphere(2, 30, 30);
glPopMatrix();

glPushMatrix();
glTranslatef(-1.5, -2, 2);
glRotatef(-20, 0, 0, 1);
glPushMatrix();
glRotatef(90, 0, 1, 0);
glScaled(0.3, 0.3, 0.3);
glColor3f(1, 0, 0);
```

```
torch();
glPopMatrix();
glPopMatrix();
if (red1) {
        glPushMatrix();
        glRotatef(50, 1, 0, 1);
        glRotatef(-55, 0, 1, 1);
        glColor3f(1, 0, 0);
        glutWireCone(1.0, 3, 10, 10);
        glPopMatrix();
}
glPushMatrix();
glTranslatef(0, -2, 2);
glRotatef(-20, 0, 0, 1);
glPushMatrix();
glRotatef(90, 0, 1, 0);
glScaled(0.3, 0.3, 0.3);
glColor3f(0, 1, 0);
torch();
glPopMatrix();
glPopMatrix();
if (green1) {
        glPushMatrix();
        glTranslatef(0, -0.5, -3);
        glRotatef(10, 1, 0, 1);
        glColor3f(0, 1, 0);
        glutWireCone(1.0, 7, 10, 10);
```

```
                glPopMatrix();
        }
        glPushMatrix();
        glTranslatef(1.5, -2, 2);
        glRotatef(-20, 0, 0, 1);
        glPushMatrix();
        glRotatef(90, 0, 1, 0);
        glScaled(0.3, 0.3, 0.3);
        glColor3f(0, 0, 1);
        torch();
        glPopMatrix();
        glPopMatrix();
        if (blue1) {
                glPushMatrix();
                glRotatef(90, 1, 0, 1);
                glColor3f(0, 0, 1);
                glutWireCone(1.0, 3, 10, 10);
                glPopMatrix();
        }
        glFlush();
        glutSwapBuffers();
}
void doDisplay()
{
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
        glLoadIdentity();
        glTranslatef(0.0f, 0.0f, -13.0f);
```

```
	if (help1)
		help();
	else
		draw();
	GLfloat mat_ambient[] = { 0.0f,1.0f,2.0f,1.0f };
	GLfloat mat_diffuse[] = { 0.0f,1.5f,.5f,1.0f };
	GLfloat mat_specular[] = { 5.0f,1.0f,1.0f,1.0f };
	GLfloat mat_shininess[] = { 100.0f };
	glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);
	glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);
	glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
	glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
	/*GLfloat lightIntensity[]={3.7f,0.7f,0.7f,1.0f}; Orange
GLfloat light_position[]={2.0f,5.0f,3.0f,1.0f};*/
/*light source properties*/
	GLfloat lightIntensity[] = { 1.7f,1.7f,1.7f,1.0f };
	GLfloat light_position[] = { 2.0f,0.0f,0.0f,0.0f };
	glLightfv(GL_LIGHT0, GL_POSITION, light_position);
	GLfloat light_position2[] = { 0.0f,0.0f,8.0f,0.0f };
	glLightfv(GL_LIGHT0, GL_POSITION, light_position2);
	GLfloat light_position3[] = { 6.0f,0.0f,5.0f,0.0f };
	glLightfv(GL_LIGHT0, GL_POSITION, light_position3);
	glLightfv(GL_LIGHT0, GL_DIFFUSE, lightIntensity);
	glFlush();
	glutSwapBuffers();
}
void mykey(unsigned char key, int x, int y)
```

```
{
        if (key == 'q' || key == 'Q')
        {
                exit(0);
        }
        if (key == 'h' || key == 'H')
        {
                help1 = !help1;
                glutPostRedisplay();
        }
        if (key == 's' || key == 'S') {
                glutIdleFunc(draw);
        }
        if (key == 'r' || key == 'R')
        {
                red1 = !red1;
                glutPostRedisplay();
        }
        if (key == 'g' || key == 'G')
        {
                green1 = !green1;
                glutPostRedisplay();
        }
        if (key == 'b' || key == 'B')
        {
                blue1 = !blue1;
                glutPostRedisplay();
```

```
		}
}
int main(int argc, char* argv[])
{
		glutInit(&argc, argv);
		glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
		glutInitWindowSize(640, 480);
		glutInitWindowPosition(0, 0);
		glutCreateWindow("Basic Structures Orientation");
		glutDisplayFunc(doDisplay);
		glEnable(GL_LIGHTING);
		glEnable(GL_LIGHT0);
		glShadeModel(GL_SMOOTH);
		glEnable(GL_DEPTH_TEST);
		glEnable(GL_NORMALIZE);
		glutKeyboardFunc(mykey);
		doInit();
		glutMainLoop();
		return 0;
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<Project DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <ItemGroup Label="ProjectConfigurations">
    <ProjectConfiguration Include="Debug|Win32">
      <Configuration>Debug</Configuration>
      <Platform>Win32</Platform>
    </ProjectConfiguration>
    <ProjectConfiguration Include="Release|Win32">
      <Configuration>Release</Configuration>
      <Platform>Win32</Platform>
    </ProjectConfiguration>
    <ProjectConfiguration Include="Debug|x64">
      <Configuration>Debug</Configuration>
      <Platform>x64</Platform>
    </ProjectConfiguration>
    <ProjectConfiguration Include="Release|x64">
      <Configuration>Release</Configuration>
      <Platform>x64</Platform>
    </ProjectConfiguration>
  </ItemGroup>
  <PropertyGroup Label="Globals">
    <VCProjectVersion>16.0</VCProjectVersion>
    <Keyword>Win32Proj</Keyword>
    <ProjectGuid>{62d200c9-8202-405d-bed6-5fce67f44087}</ProjectGuid>
    <RootNamespace>colorcombination</RootNamespace>
    <WindowsTargetPlatformVersion>10.0</WindowsTargetPlatformVersion>
```

```
  </PropertyGroup>
  <Import Project="$(VCTargetsPath)\Microsoft.Cpp.Default.props" />
  <PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Debug|Win32'"
Label="Configuration">
    <ConfigurationType>Application</ConfigurationType>
    <UseDebugLibraries>true</UseDebugLibraries>
    <PlatformToolset>v142</PlatformToolset>
    <CharacterSet>Unicode</CharacterSet>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Release|Win32'"
Label="Configuration">
    <ConfigurationType>Application</ConfigurationType>
    <UseDebugLibraries>false</UseDebugLibraries>
    <PlatformToolset>v142</PlatformToolset>
    <WholeProgramOptimization>true</WholeProgramOptimization>
    <CharacterSet>Unicode</CharacterSet>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Debug|x64'" Label="Configuration">
    <ConfigurationType>Application</ConfigurationType>
    <UseDebugLibraries>true</UseDebugLibraries>
    <PlatformToolset>v142</PlatformToolset>
    <CharacterSet>Unicode</CharacterSet>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Release|x64'" Label="Configuration">
    <ConfigurationType>Application</ConfigurationType>
    <UseDebugLibraries>false</UseDebugLibraries>
    <PlatformToolset>v142</PlatformToolset>
    <WholeProgramOptimization>true</WholeProgramOptimization>
```

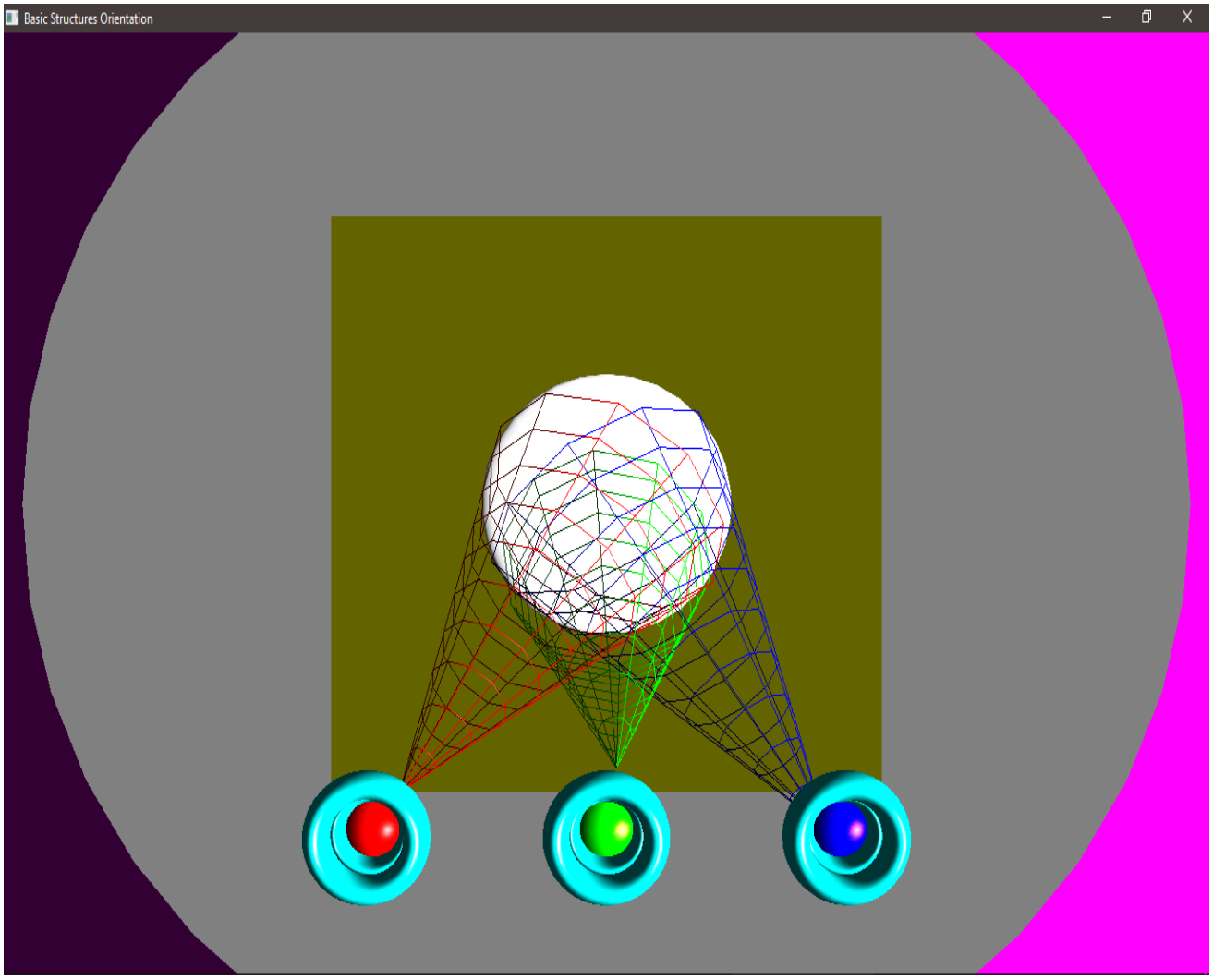
```
    <CharacterSet>Unicode</CharacterSet>
  </PropertyGroup>
  <Import Project="$(VCTargetsPath)\Microsoft.Cpp.props" />
  <ImportGroup Label="ExtensionSettings">
  </ImportGroup>
  <ImportGroup Label="Shared">
  </ImportGroup>
  <ImportGroup Label="PropertySheets"
Condition="'$(Configuration)|$(Platform)'=='Debug|Win32'">
    <Import Project="$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props"
Condition="exists('$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props')"
Label="LocalAppDataPlatform" />
  </ImportGroup>
  <ImportGroup Label="PropertySheets"
Condition="'$(Configuration)|$(Platform)'=='Release|Win32'">
    <Import Project="$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props"
Condition="exists('$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props')"
Label="LocalAppDataPlatform" />
  </ImportGroup>
  <ImportGroup Label="PropertySheets" Condition="'$(Configuration)|$(Platform)'=='Debug|x64'">
    <Import Project="$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props"
Condition="exists('$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props')"
Label="LocalAppDataPlatform" />
  </ImportGroup>
  <ImportGroup Label="PropertySheets" Condition="'$(Configuration)|$(Platform)'=='Release|x64'">
    <Import Project="$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props"
Condition="exists('$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props')"
Label="LocalAppDataPlatform" />
  </ImportGroup>
  <PropertyGroup Label="UserMacros" />
  <PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Debug|Win32'">
```

```
    <LinkIncremental>true</LinkIncremental>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Release|Win32'">
    <LinkIncremental>false</LinkIncremental>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Debug|x64'">
    <LinkIncremental>true</LinkIncremental>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Release|x64'">
    <LinkIncremental>false</LinkIncremental>
  </PropertyGroup>
  <ItemDefinitionGroup Condition="'$(Configuration)|$(Platform)'=='Debug|Win32'">
    <ClCompile>
      <WarningLevel>Level3</WarningLevel>
      <SDLCheck>true</SDLCheck>

<PreprocessorDefinitions>WIN32;_DEBUG;_CONSOLE;%(PreprocessorDefinitions)</PreprocessorDef
initions>
      <ConformanceMode>true</ConformanceMode>
    </ClCompile>
    <Link>
      <SubSystem>Console</SubSystem>
      <GenerateDebugInformation>true</GenerateDebugInformation>
    </Link>
  </ItemDefinitionGroup>
  <ItemDefinitionGroup Condition="'$(Configuration)|$(Platform)'=='Release|Win32'">
    <ClCompile>
      <WarningLevel>Level3</WarningLevel>
```

```
      <FunctionLevelLinking>true</FunctionLevelLinking>
      <IntrinsicFunctions>true</IntrinsicFunctions>
      <SDLCheck>true</SDLCheck>
<PreprocessorDefinitions>WIN32;NDEBUG;_CONSOLE;%(PreprocessorDefinitions)</PreprocessorDefinitions>
      <ConformanceMode>true</ConformanceMode>
    </ClCompile>
    <Link>
      <SubSystem>Console</SubSystem>
      <EnableCOMDATFolding>true</EnableCOMDATFolding>
      <OptimizeReferences>true</OptimizeReferences>
      <GenerateDebugInformation>true</GenerateDebugInformation>
    </Link>
  </ItemDefinitionGroup>
  <ItemDefinitionGroup Condition="'$(Configuration)|$(Platform)'=='Debug|x64'">
    <ClCompile>
      <WarningLevel>Level3</WarningLevel>
      <SDLCheck>true</SDLCheck>
<PreprocessorDefinitions>_DEBUG;_CONSOLE;%(PreprocessorDefinitions)</PreprocessorDefinitions>
      <ConformanceMode>true</ConformanceMode>
    </ClCompile>
    <Link>
      <SubSystem>Console</SubSystem>
      <GenerateDebugInformation>true</GenerateDebugInformation>
    </Link>
  </ItemDefinitionGroup>
```

```
  <ItemDefinitionGroup Condition="'$(Configuration)|$(Platform)'=='Release|x64'">
    <ClCompile>
      <WarningLevel>Level3</WarningLevel>
      <FunctionLevelLinking>true</FunctionLevelLinking>
      <IntrinsicFunctions>true</IntrinsicFunctions>
      <SDLCheck>true</SDLCheck>

<PreprocessorDefinitions>NDEBUG;_CONSOLE;%(PreprocessorDefinitions)</PreprocessorDefinitions
>
      <ConformanceMode>true</ConformanceMode>
    </ClCompile>
    <Link>
      <SubSystem>Console</SubSystem>
      <EnableCOMDATFolding>true</EnableCOMDATFolding>
      <OptimizeReferences>true</OptimizeReferences>
      <GenerateDebugInformation>true</GenerateDebugInformation>
    </Link>
  </ItemDefinitionGroup>
  <ItemGroup>
    <ClCompile Include="color_combination.cpp" />
  </ItemGroup>
  <Import Project="$(VCTargetsPath)\Microsoft.Cpp.targets" />
  <ImportGroup Label="ExtensionTargets">
  </ImportGroup>
</Project>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="4.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <ItemGroup>
    <Filter Include="Source Files">
      <UniqueIdentifier>{4FC737F1-C7A5-4376-A066-2A32D752A2FF}</UniqueIdentifier>
      <Extensions>cpp;c;cc;cxx;c++;cppm;ixx;def;odl;idl;hpj;bat;asm;asmx</Extensions>
    </Filter>
    <Filter Include="Header Files">
      <UniqueIdentifier>{93995380-89BD-4b04-88EB-625FBE52EBFB}</UniqueIdentifier>
      <Extensions>h;hh;hpp;hxx;h++;hm;inl;inc;ipp;xsd</Extensions>
    </Filter>
    <Filter Include="Resource Files">
      <UniqueIdentifier>{67DA6AB6-F800-4c08-8B7A-83BB121AAD01}</UniqueIdentifier>
      <Extensions>rc;ico;cur;bmp;dlg;rc2;rct;bin;rgs;gif;jpg;jpeg;jpe;resx;tiff;tif;png;wav;mfcribbon-
ms</Extensions>
    </Filter>
  </ItemGroup>
  <ItemGroup>
    <ClCompile Include="color_combination.cpp">
      <Filter>Source Files</Filter>
    </ClCompile>
  </ItemGroup>
</Project>
```


Basic Structures Orientation