リスト3.1　「glColorMap.cpp」の一部

```
#include <windows.h>
#include <stdio.h>
#include <GL/glut.h>
#include "../../myPrimitive2.h"
#include "../../myTexture2.h"
#include "../../imageLoadSave.h"
#define TEX_WIDTH 64
#define TEX_HEIGHT 64
GLubyte texImage[TEX_HEIGHT][TEX_WIDTH][4];
GLuint texName[2];
//他は省略
void init(void)
{
 //中略
  //テクスチャー
  glGenTextures(2, texName);//テクスチャ・オブジェクトの名前付け
  //objNo=0
  makeCheckImage(4);
  setTexture(0);
  //objNo=1
  Bitmap* bm1 = new Bitmap();
  loadBitmap(bm1, "../../bmp/face1.bmp");
  makeTexImage(bm1);
  setTexture(1);
}

//--------------------------------------------------------------------
void makeCheckImage(int numCheck)
{
  int i, j, ii, jj, col;
  float ww = TEX_WIDTH / (float)numCheck;
  float hh = TEX_HEIGHT / (float)numCheck;

  for(j = 0; j < TEX_HEIGHT; j++)
  {
    jj = (int)((float)j / hh);
    for(i = 0; i < TEX_WIDTH; i++)
    {
      ii = (int)((float)i / ww);
      col = ((ii + jj) & 1) * 255;

      texImage[j][i][0] = col; //red
      texImage[j][i][1] = 0;   //green
      texImage[j][i][2] = ~col;//blue
      texImage[j][i][3] = 255;
    }
  }
}

//画像ファイルからイメージ・データを作成
void makeTexImage(Bitmap *bm)
{
  int i, j, ii, jj;

  if(bm->bi.Width < TEX_WIDTH) printf("画像ファイルのｻｲｽﾞ>=TEX_WIDTHとすること¥n");
  for(j = 0; j < TEX_HEIGHT; j++)
  {
    jj = j * (int)((float)bm->bi.Height / (float)TEX_HEIGHT);
    for(i = 0; i < TEX_WIDTH; i++)
    {
      ii = i * (int)((float)bm->bi.Width / (float)TEX_WIDTH) ;
      if(bm->bi.BitCount <= 24)
      {
```

```
        texImage[j][i][0] = (bm->pixel[ii+jj*bm->bi.Width]).r;
        texImage[j][i][1] = (bm->pixel[ii+jj*bm->bi.Width]).g;
        texImage[j][i][2] = (bm->pixel[ii+jj*bm->bi.Width]).b;
        texImage[j][i][3] = 255;
      }
      else
      {
        texImage[j][i][0] = (bm->pixelA[ii+jj*bm->bi.Width]).r;
        texImage[j][i][1] = (bm->pixelA[ii+jj*bm->bi.Width]).g;
        texImage[j][i][2] = (bm->pixelA[ii+jj*bm->bi.Width]).b;
        texImage[j][i][3] = (bm->pixelA[ii+jj*bm->bi.Width]).a;
      }
    }
  }
  if(bm->bi.BitCount <= 24) free(bm->pixel);
  else free(bm->pixelA);
}

void setTexture(int n)
{
  glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
  //テクスチャ・オブジェクトの作成
  glBindTexture(GL_TEXTURE_2D, texName[n]);
  //テクスチャの指定

glTexImage2D(GL_TEXTURE_2D,0,GL_RGBA,TEX_WIDTH,TEX_HEIGHT,0,GL_RGBA,GL_UNSIGNE
D_BYTE,texImage);
  //テクスチャの繰り返し方法の指定
  glTexParameteri(GL_TEXTURE_2D,GL_TEXTURE_WRAP_S,GL_CLAMP);
  glTexParameteri(GL_TEXTURE_2D,GL_TEXTURE_WRAP_T,GL_CLAMP);
  //テクスチャを拡大・縮小する方法の指定
  glTexParameteri(GL_TEXTURE_2D,GL_TEXTURE_MAG_FILTER,GL_LINEAR);
  glTexParameteri(GL_TEXTURE_2D,GL_TEXTURE_MIN_FILTER,GL_LINEAR);
  //色の調整
  //glTexEnvi(GL_TEXTURE_ENV,GL_TEXTURE_ENV_MODE,GL_BLEND);
  //glTexEnvi(GL_TEXTURE_ENV,GL_TEXTURE_ENV_MODE,GL_DECAL);
  glTexEnvi(GL_TEXTURE_ENV,GL_TEXTURE_ENV_MODE,GL_MODULATE);
  glBindTexture(GL_TEXTURE_2D, 0);
}

void display(void)
{
//中略
  //描画
  glBindTexture(GL_TEXTURE_2D, texName[0]);
  draw0();
  glBindTexture(GL_TEXTURE_2D, 0);
  glBindTexture(GL_TEXTURE_2D, texName[1]);
  draw1();
  glBindTexture(GL_TEXTURE_2D, 0);

  drawFloor(10.0, 10.0, 10, 10);
  //影
  drawShadow();
//中略
  glutSwapBuffers();
}
```