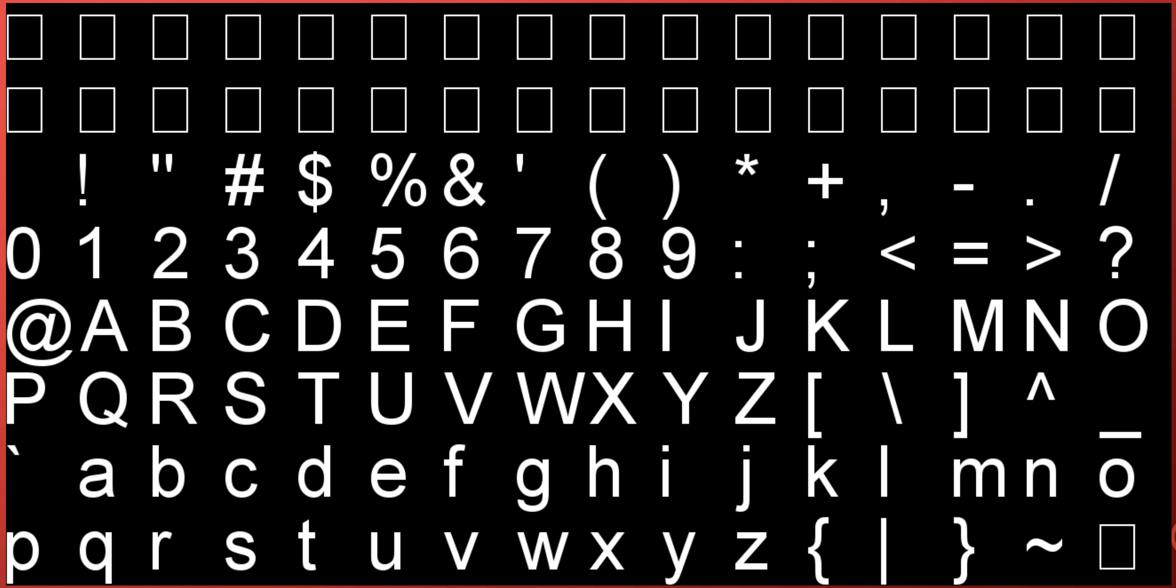
OPENGL TEXT AND SOUND

MARK DANIEL G. DACER

BITMAP FONTS

- A bitmap font is basically a 2D font. Although we'll place it in a 3D world, these fonts will have no thickness and can't be rotated or scaled, only translated. Furthermore, the font will always face the viewer, like a billboard. Although this can be seen as a potential disadvantage, on the other hand we won't have to worry about orienting the font to face the viewer



SYNTAX

- void glutBitmapCharacter(void *font, int character)
- Parameters:
  - font – the name of the font to use (see bellow for a list of what's available
  - character – what to render, a letter, symbol, number, etc…

- glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,'3');

FONTS AVAILABLE

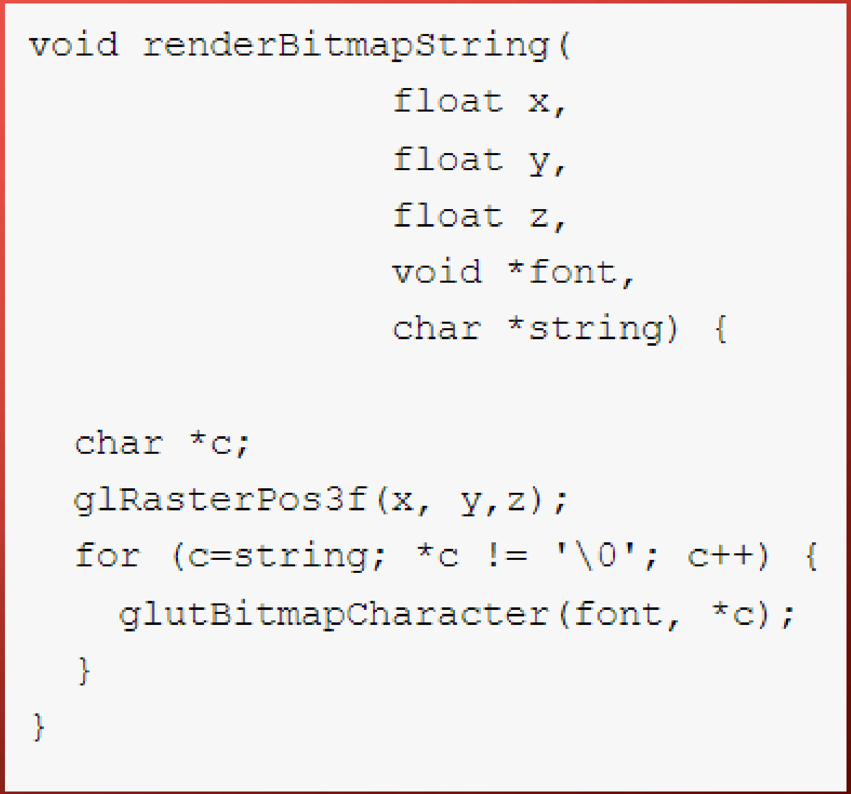
- GLUT_BITMAP_8_BY_13
- GLUT_BITMAP_9_BY_15
- GLUT_BITMAP_TIMES_ROMAN_10
- GLUT_BITMAP_TIMES_ROMAN_24
- GLUT_BITMAP_HELVETICA_10
- GLUT_BITMAP_HELVETICA_12
- GLUT_BITMAP_HELVETICA_18

FONTS (CONT.)

- One important thing to know is what is the actual raster position. The raster position can be set with the family of functions *glRasterPos* from the OpenGL library. The syntax of two functions from this family is presented below.
- void glRasterPos2f(float x, float y);
  void glRasterPos3f(float x, float y, float z);
  - Parameters:
  - x, y, z – local coordinates for the text to appear

FONTS (CONT.)

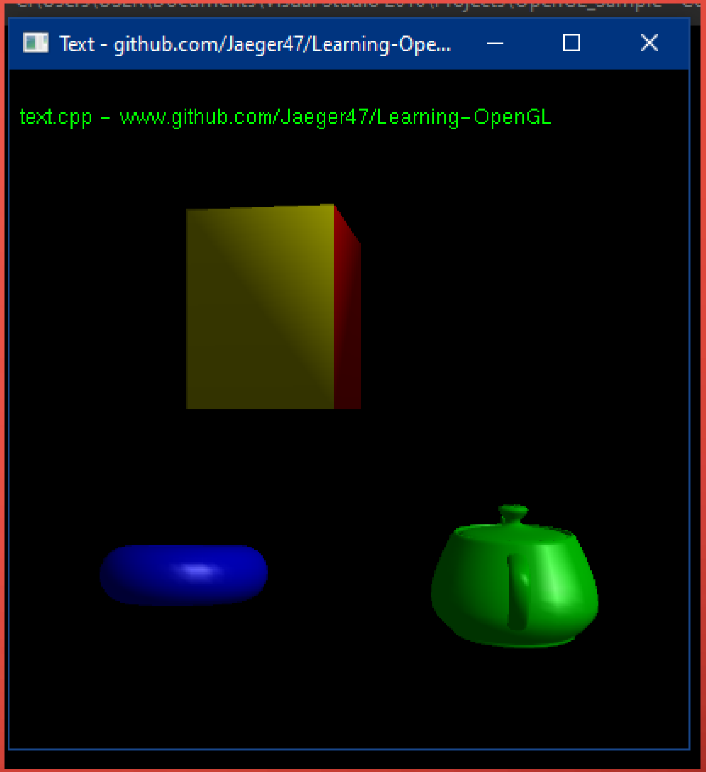
- The function glutBitmapCharacter renders the character at the required position and advances the current raster position by the width of the character. Therefore, to render a string, successive calls to glutBitmapCharacter will suffice to achieve the desired output. The following function renders a string starting at the specified raster position:



```
void renderBitmapString(
                float x,
                float y,
                float z,
                void *font,
                char *string) {

  char *c;
  glRasterPos3f(x, y,z);
  for (c=string; *c != '\0'; c++) {
    glutBitmapCharacter(font, *c);
  }
}
```

PREVIEW

```
void renderBitmapString(
        float x,
        float y,
        float z,
        void *font,
        char *string) {
  char *c;
  glRasterPos3f(x, y,z);
  for (c=string; *c != '\0'; c++) {
    glutBitmapCharacter(font, *c);
  }
}
```



```
text.cpp – www.github.com/Jaeger47/Learning-OpenGL
```

```
glDisable(GL_LIGHTING);
glColor4f(0.0f,1.0f,0.0f,1.0f);
renderBitmapString(-4.0f, 3.5f, 5.0f, GLUT_BITMAP_HELVETICA_12,"text.cpp – www.github.com/Jaeger47/Learning-OpenGL");
glEnable(GL_LIGHTING);
```

OPENGL SOUND

- OpenGL doesn't offer us any support for audio capabilities (like many other aspects of game development). We have to manually load audio files into a collection of bytes, process and convert them to an audio stream, and manage multiple audio streams appropriately for use in our game. This can get complicated pretty quick and requires some low-level knowledge of audio engineering.

IRRKLANG

- IrrKlang is a high level 2D and 3D cross platform (Windows, Mac OS X, Linux) sound engine and audio library that plays WAV, MP3, OGG, and FLAC files to name a few. It also features several audio effects like reverb, delay, and distortion that can be extensively tweaked.

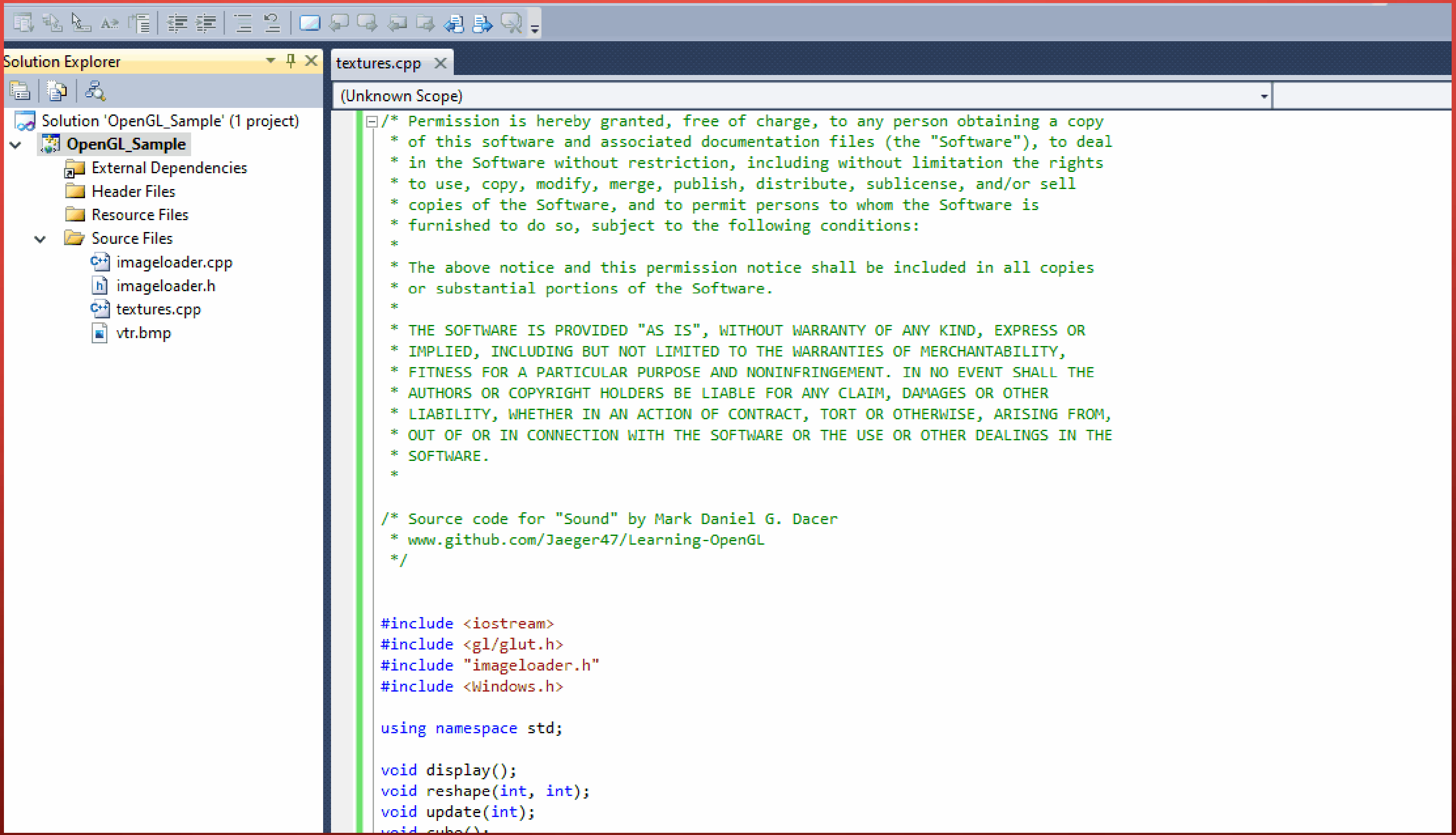
SETTING UP IIRKLANG

- Download iirKlang on this website: https://www.ambiera.com/irrklang/downloads.html
- Use 32 bit version
- Extract the zip, you'll see a bunch of files and folders
- 1st go to the bin\win32-visualStudio folder and copy all the .dll files to your system32 and sysWOW64

SETTING UP IIRKLANG (CONT.)

- When using visual studio 2010 follow this
- 1st create a folder name irr on your *visual studio directory*\VC\include Example“: C:\Program Files\Microsoft Visual Studio 10.0\VC\include
- 2nd go to the irrklang folder and go to include then copy all the .h files it to the newly created irr folder.
- 3rd go the irrklang folder and go to the lib\Win32-visualStudio folder and copy all files to visual studio directory\VC\lib folder
- Example: C:\Program Files\Microsoft Visual Studio 10.0\VC\lib

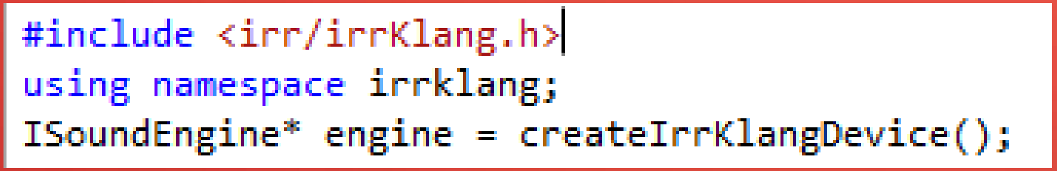
LAST STEP



```
/* Permission is hereby granted, free of charge, to any person obtaining a copy
 * of this software and associated documentation files (the "Software"), to deal
 * in the Software without restriction, including without limitation the rights
 * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
 * copies of the Software, and to permit persons to whom the Software is
 * furnished to do so, subject to the following conditions:
 *
 * The above notice and this permission notice shall be included in all copies
 * or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
 * SOFTWARE.
 *

/* Source code for "Sound" by Mark Daniel G. Dacer
 * www.github.com/Jaeger47/Learning-OpenGL
 */


#include <iostream>
#include <gl/glut.h>
#include "imageloader.h"
#include <Windows.h>

using namespace std;

void display();
void reshape(int, int);
void update(int);
```

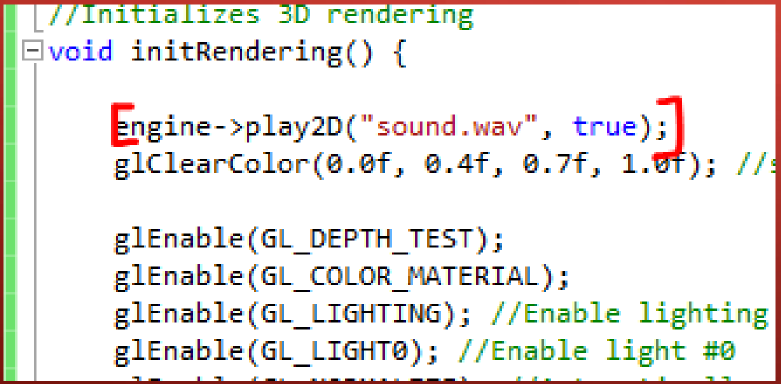
PREVIEW

- irrKlang setup



```
#include <irr/irrKlang.h>
using namespace irrklang;
ISoundEngine* engine = createIrrKlangDevice();
```

- Playing audio using play2D(*LOCATION, LOOPING*)



```
//Initializes 3D rendering
void initRendering() {

    engine->play2D("sound.wav", true);
    glClearColor(0.0f, 0.4f, 0.7f, 1.0f); //s

    glEnable(GL_DEPTH_TEST);
    glEnable(GL_COLOR_MATERIAL);
    glEnable(GL_LIGHTING); //Enable lighting
    glEnable(GL_LIGHT0); //Enable light #0
```

ADDITIONAL

- Please read the irrKlang for more information: https://www.ambiera.com/irrklang/tutorials.html
- If you cant setup or having errors setting up email, dm me or go to the faculty for clarifications.

EXTRA LINKS

- https://learn.microsoft.com/en-us/windows/win32/opengl/fonts-and-text
- https://www.ambiera.com/irrklang/index.html
- https://learnopengl.com/In-Practice/2D-Game/Audio