SPRAWOZDANIE

Zajęcia: Grafika komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 7

Data 21.04.2022

Temat: Tekstury OpenGL

Natalia Pierzchała

Informatyka I stopień,

stacjonarne,

4 semestr,

Gr. 2a

1. Polecenie:

Celem jest teksturowanie piramidy z użyciem dwóch sposobów ładowania tekstur: użycie tekstury z buforu kolorów (rysowanie w Panel); ładowanie tekstury z pliku (trzy pliki przykładowe do pobrania).

Należy opracować metody textureFromPainting() oraz textureFromResource() klasy Lab7.

2. Wprowadzane dane:

```
private Texture textureFromResource(String resourceName) throws IOException {
    /*
        Reads image from file.
     */
    URL textureURL;
    textureURL = this.getClass().getClassLoader().getResource(resourceName);
    BufferedImage img = ImageIO.read(Objects.requireNonNull(textureURL));

    Texture texture;
    ImageUtil.flipImageVertically(img);

    /*
        Makes the context current in current thread
     */
    GLContext context = displayGL.getContext();
    boolean needsRelease = false;
    if (!context.isCurrent()) {
        context.makeCurrent();
        needsRelease = true;
    }

    GL2 gl2 = context.getGL().getGL2(); // Gets the gl2 on current context

    /*
        Creates texture from file read before.
     */
    texture = AWTTextureIO.newTexture(displayGL.getGLProfile(), img, true);
    texture.setTexParameteri(gl2, GL2.GL_TEXTURE_WRAP_S, GL2.GL_REPEAT);
    texture.setTexParameteri(gl2, GL2.GL_TEXTURE_WRAP_T, GL2.GL_REPEAT);

    if (needsRelease) {
        context.release();
    }

    return texture;
}
```

```
private Texture textureFromPainting() {
    Texture texture;
    BufferedImage img = paintPanel.copyOSC(); // Gets the image from paintPanel

    /*
        Makes the context current in current thread
     */
    GLContext context = displayGL.getContext();
    boolean needsRelease = false;
    if (!context.isCurrent()) {
        context.makeCurrent();
        needsRelease = true;
    }

    GL2 gl2 = context.getGL().getGL2(); // Gets the gl2 on current context

    /*
        Creates texture from image gotten before from paintPanel.
     */
    texture = AWTTextureIO.newTexture(displayGL.getGLProfile(), img, true);
    texture.setTexParameteri(gl2, GL2.GL_TEXTURE_WRAP_S, GL2.GL_REPEAT);
    texture.setTexParameteri(gl2, GL2.GL_TEXTURE_WRAP_T, GL2.GL_REPEAT);

    if (needsRelease) {
        context.release();
    }

    return texture;
}

private void paintingFromOpenGL() {
    GLContext context = displayGL.getContext(); // OpenGL context for the display panel.
    boolean needsRelease = false;  // Will be set to true if context needs to be made current.
    if (!context.isCurrent()) {
        // Make the context current on the current thread.
        context.makeCurrent();
        needsRelease = true;
    }
    GL2 gl2 = context.getGL().getGL2();
    AWTGLReadBufferUtil readBuf = new AWTGLReadBufferUtil(displayGL.getGLProfile(), false);
    BufferedImage img = readBuf.readPixelsToBufferedImage(gl2, true); // Get display content as image.
    if (needsRelease) {
        context.release();
    }
    paintPanel.installImage(img); // copy the image into the PaintPanel.
}
```

```java
public void display(GLAutoDrawable drawable) {

    GL2 gl2 = drawable.getGL().getGL2(); // The object that contains all the OpenGL methods.

    gl2.glClear(GL2.GL_COLOR_BUFFER_BIT | GL2.GL_DEPTH_BUFFER_BIT);

    camera.apply(gl2); // Sets projection and view transformations.

    Texture tex = currentTexture; // Creates new Texture object that represents currentTexture

    /*
        Checks if currentTexture (tex) is null, if not it binds texture it
        holds to context and draws the shape with texture applied. If there is no
        currentTexture (tex is null) it disables 2D textures.
     */
    if (tex != null) {
        tex.enable(gl2);
        tex.bind(gl2);
        drawCurrentShape(gl2);
        tex.disable(gl2);
    } else
        drawCurrentShape(gl2);

} // end display()
```

3. Wykorzystane komendy:

https://github.com/99lucky8/Grafika-komputerowa.git

4. Wyniki działania