

RAČUNALNA GRAFIKA

Mario Bat

Tomislav Prodanović

OPENGL ES
(Android)



Sadržaj

- Uvod
- OpenGL ES
- Postavljanje okruženja
- GLSurfaceView
- Renderer
- Poligoni
- Primjeri
- Zaključak



Uvod

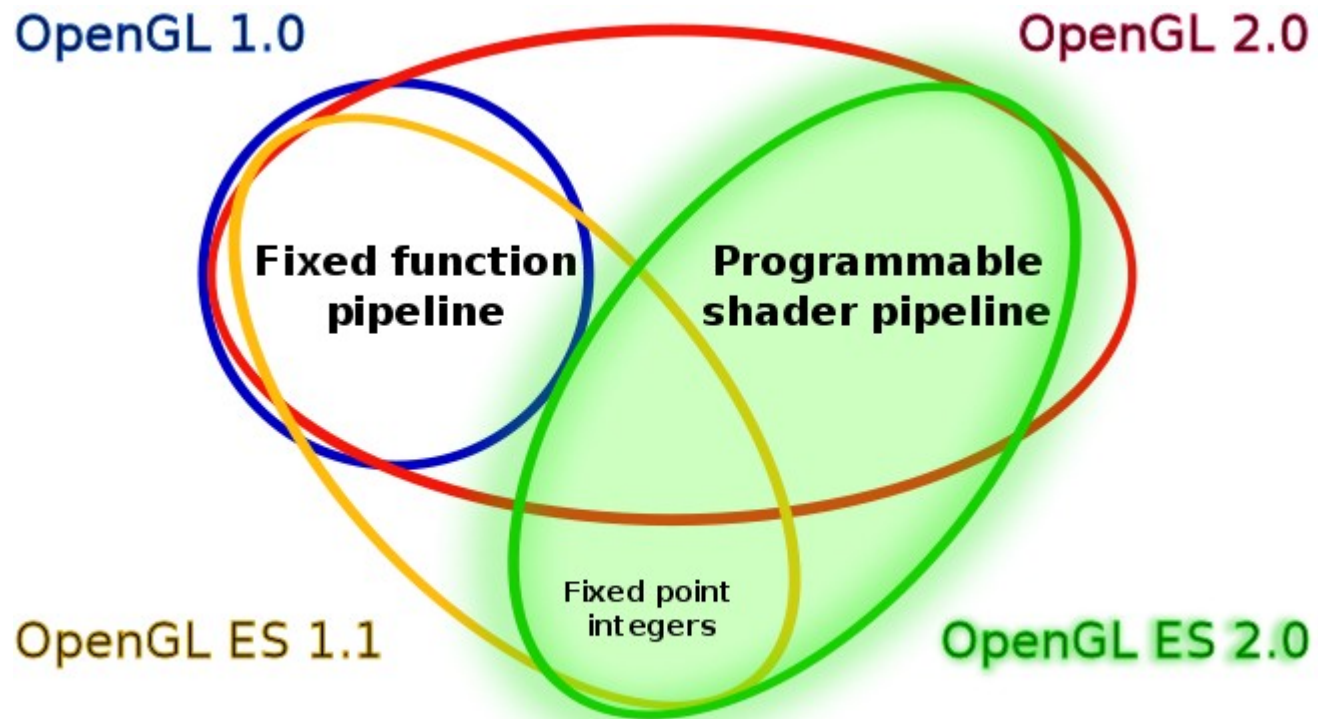
- Android je operacijski sustav za mobilne uređaje (smartphone i tableti) temeljen na Linux platformi
- Android, Inc. osnovan 2003. godine u Palo Alto, California
- 17.8.2005 Google preuzima Android Inc.
- OpenGL ES 1.0, 1.1, 2.0



OPENGL ES

- OpenGL for Embedded Systems
- Podskup OpenGL 3D grafičkih API-a za crtanje 3D grafike na mobilnim uređajima
- OpenGL ES-om upravlja neprofitni tehnološki konzorcij, Khronos Group Inc.
 - <http://www.khronos.org/opengles/>

OPENGL ES



OPENGL ES

- Fixed Function Pipeline (FFP) – skup funkcija zadužene za proces renderiranja
- Fixed Point Integers – konačan broj decimalne preciznosti
- Programmable Shader Pipeline – programabilni shaderi
- Zaključak: za početnike se preporuča Android 1.0 i 1.1

Android SDK instalacija

- Instalirati JDK (Java Development Kit)
 - <http://www.oracle.com/technetwork/java/javase/downloads/jdk-7u2-download-1377129.html>
(JDK 7)
- Instalirati Eclipse IDE (Eclipse Integrated Development Kit)
 - <http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/indigosr1>

Android SDK instalacija

- Instalirati Android SDK Starter Package (ovaj SDK nije potpun te sadrži samo osnovne SDK alate, koji olakšavaju instalaciju ostalih SDK komponenti)
 - <http://developer.android.com/sdk/index.html>

Android SDK instalacija

- Instalirati ADT Plugin
 - Pokrenuti Eclipse > Help > Install New Software...
 - Kliknuti na Add (gore desno)
 - U prozoru „Add Repository” upisati Name npr. „ADT Plugin”, u Location upisati „<https://dl-ssl.google.com/android/eclipse/>”
 - AKO IMATE PROBLEMA PRILIKOM DOHVAĆANJA PLUGINA UPIŠITE HTTP UMJESTO HTTPS – HTTPS se koristi zbog sigurnosnih razloga



Android SDK instalacija

- U sljedećem prozoru označiti checkbox „Developer Tools” > Next
- U slijedećem prozoru možemo vidjeti popis alata koji za skinuti > Next
- Pročitati i prihvatiti EULA > Finish
- Restartirati Eclipse

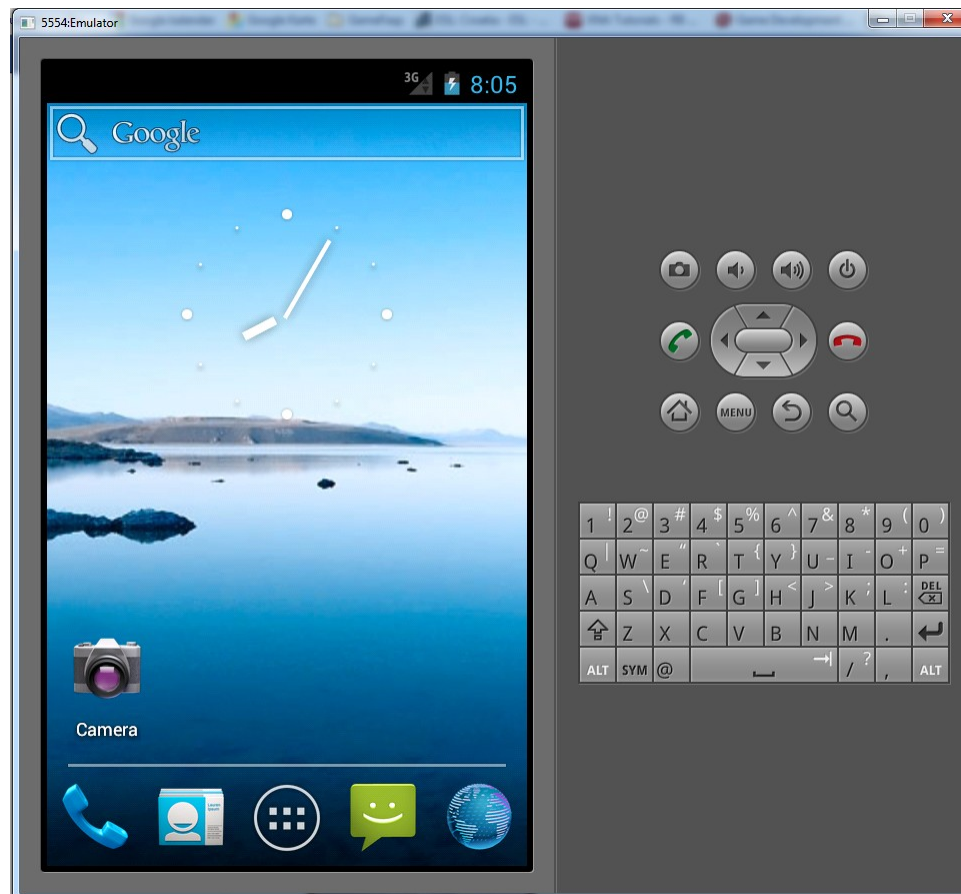
Konfiguracija Android SDK

- Otvoriti Eclipse
- Windows > Preferences
- Lijevo na popisu odabrati Android > u SDK Location upisati putanju do android sdk npr. „C:\Program Files(x86)\Android\android-sdk”
> Apply > OK
 - Ispod bi se trebao prikazati popis Android platformi

Kreiranje emulatora

- Kako bi se mogle pokretati Android aplikacije potrebno je postaviti emulator AVD (Android Virtual Device)
- Pokrenemo Eclipse > Window > AVD Manager
- U sljedećem prozoru kliknemo na New > u Name upišemo naziv npr. „Emulator” > Target postavimo verziju Androida koju želimo pokretati > Create AVD
- Sve ostale opcije ostavimo defaultne

Kreiranje emulatora



Kreiranje projekta

- Pokrenuti Eclipse
- File > New > Android Project > upisati Project Name > Next
- Treba odabrati SDK target tj. za koju Android platformu ćemo razvijati aplikaciju > Next
- Upisati Package Name npr. „org.foi.rg”> Finish

Kreiranje projekta

- Ovisno o verziji OpenGL-a u kojoj se radi aplikacija, bilo bi dobro dodati neke postavke u AndroidManifest.xml
 - `<uses-feature
 android:glEsVersion="0x00010000"
 android:required="true" />`
- Ovom deklaracijom omogućavamo Android Marketu restrikciju aplikacije na mobilne uređaje koji moraju podržavati OpenGL ES 1.0.

GLSurfaceView

- setContent – funkcija koja postavlja „platno” (view) koje korisnik vidi
- defaultno je postavljen `R.layout.main` – to je resurs koji se nalazi u direktoriju layout
- OpenGL ne podržava ovaj view, ne može „crtati” na njemu – potrebno je zamijeniti s nečim što OpenGL podržava



GLSurfaceView

- GLSurfaceView – view na kojemu e moguće crtati i manipulirati objektima preko OpenGL API
 - Potrebno mu je dodijeliti renderer
- Renderer – definira metode potrebne za crtanje u OpenGLSurfaceView
 - Potrebno je napraviti implementaciju kao posebnu klasu i zakačiti ju na GLSurfaceView



GLSurfaceView

```
GLSurfaceView view = new GLSurfaceView()  
view.setRenderer(new OpenGLRenderer());  
setContentView(view);
```

OpenGLRenderer je klasa koja nasljeđuje
Renderer klasu i nju tek treba kreirati – ona sadrži
metode za crtanje na view

Renderer - OpenGLRenderer

- Nasljeđuje 3 metode:
 - `public void onDrawFrame(GL10 gl)`
 - Poziva se kada svaki puta kada se crta novi frame
 - `Public void onSurfaceChanged(GL10 gl, int width, int height)`
 - Poziva se kada se mijenja površina (npr. prijelaz iz Portrait u Landscape)

Kreiranje poligona

- Napravimo klasu Kvadrat
- U Androidu se točke definiraju u polju tipa float

```
private float vertices[] = {  
    -1.0f, 1.0f, 0.0f, // 0. Gore lijevo  
    -1.0f, -1.0f, 0.0f, // 1. Dolje lijevo  
    1.0f, -1.0f, 0.0f, // 2. Dolje desno  
    1.0f, 1.0f, 0.0f, // 3. Gore desno  
};
```

Kreiranje poligona

- Moramo odrediti način na koji će se prije definirani vektori spajati

- Određujemo ih u polju tipa integer

```
private short indices[] = { 0, 1, 2, 0,  
2, 3};
```

- Definirani vektori će se spajati na način

0-1-2 0-2-3

- REDOSLIJED JE VAŽAN!

Kreiranje poligona

- Android aplikacije se programiraju u Java programskom jeziku, ali temeljna implementacija je pisana u C-u
- Prije nego proslijedimo naše podatke OpenGL-u moramo ih pretvoriti u njemu razumljiv oblik

```
private FloatBuffer vertexBuffer;  
private ShortBuffer indexBuffer;
```

Kreiranje poligona

- Za to se koristi poseban set buffera koji sprema naše podatke u izvorni redoslijed byteova

```
ByteBuffer vbb = ByteBuffer.allocateDirect(vertices.length * 4);  
vbb.order(ByteOrder.nativeOrder());  
vertexBuffer = vbb.asFloatBuffer();  
vertexBuffer.put(vertices);  
vertexBuffer.position(0);
```

- Naši vektori su definirani u polju tipa float i stoga je potrebno množiti sa 4

Kreiranje poligona

- Potrebno je definirati metodu koja će iscrtavati željeni poligon

```
public void draw(GL10 gl) {  
    gl.glEnableClientState(GL10.GL_VERTEX_ARRAY);  
    gl.glVertexPointer(3, GL10.GL_FLOAT, 0, vertexBuffer);  
    gl.glDrawElements(GL10.GL_TRIANGLES, indices.length,  
        GL10.GL_UNSIGNED_SHORT, indexBuffer);  
    gl.glDisableClientState(GL10.GL_VERTEX_ARRAY);  
    gl.glDisable(GL10.GL_CULL_FACE);  
}
```

Kreiranje poligona

- Na kraju je u klasi OpenGLRenderer potrebno instancirati objekt Square
- U onDrawFrame je potrebno ubaciti

```
gl.glClear(GL10.GL_COLOR_BUFFER_BIT |  
GL10.GL_DEPTH_BUFFER_BIT);  
gl.glLoadIdentity();  
gl.glTranslatef(0, 1.5f, -7);  
square.draw(gl);
```

Ostale funkcionalnosti

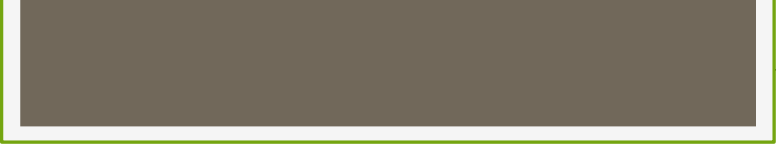
- OpenGL ES kao i običan OpenGL podržava FlatColor i SmoothColor
- Podržava texture
- Translacije, rotacije, skaliranje, PushMatrix, PopMatrix

Korisni tutoriali

- <http://blog.jayway.com/2009/12/03/opengl-es-tutorial-for-android-part-i/>
- <http://developer.android.com/guide/topics/graphics/opengl.html>
- <http://www.droidnova.com/android-3d-game-tutorial-part-i,312.html>
- <http://obviam.net/>
- Video:
 - <http://www.youtube.com/watch?v=U4Bk5rmIpic>

Zaključak

- Važno je poznavanje OpenGL-a i razumijevanje matrica
- Potrebno puno vremena za savladavanje
- Velika sličnost OpenGL-a za Android i iPhone



HVALA NA PAŽNJI!