Graphics on Smartphones

Bohdan Romashchenko
Goal

- Learn, describe and demonstrate a couple of ways to bring custom 3D graphics to mobile apps
Motivation

- CG course 2020/2021
- Mobile app enthusiast
Mobile apps

- Android
- iOS
- Flutter
- Xamarin
- React Native
3D graphics

- Unity
- OpenGL
- Unreal Engine
- three.js
Android OpenGL

- Up to OpenGL ES 3.1
- Both SDK and NDK
- Good control over graphics
Android SDK OpenGL

- **GlSurfaceView**
- **GlSurfaceView.Renderer**
  - onSurfaceCreated()
  - onDrawFrame()
  - onSurfaceChanged()
DEMO
Three.js

- JS based cross-platform mobile apps (React Native, Ionic, etc.)
- Relatively easy to use
- OOP
React Native + Three.js

- Through Expo!
- Packages: three, expo-three, expo-gl
React Native + Three.js

```javascript
return (
  <GLView
    style={{ flex: 1 }}
    onContextCreate={async (gl) => {
      // THREE CODE GOES HERE
      // Render function
      const render = () => {
        requestAnimationFrame(render);
        renderer.render(scene, camera);
        gl.endFrameEXP();
      };
      render();
    }}
  >
)
```
React Native + Three.js + React Three Fiber

- Three.js classes as React components
- No overhead
React Native + Three.js + React Three Fiber

```javascript
return (
  <mesh
    {...props}
    ref={mesh}
    scale={active ? [1.5, 1.5, 1.5] : [1, 1, 1]}
    onClick={e => setActive(!active)}
    onPointerOver={(e) => setHover(true)}
    onPointerOut={(e) => setHover(false)}
  >
    <boxBufferGeometry attach="geometry" args={[1, 1, 1]} />
    <meshStandardMaterial
      attach="material"
      color={hovered ? "hotpink" : "orange"}
    />
  </mesh>
);
```
DEMO
Sources

1. OpenGL ES | Android Developers
2. OpenGL ES - OpenGL Wiki (khronos.org)
3. Three.js – JavaScript 3D Library (threejs.org)
4. expo/expo-three: Utilities for using THREE.js on Expo (github.com)
5. react-three-fiber