VIRTUAL REALITY

Erik Pronk
&
Alexander Chatzizacharias

@alex90_ch

@erikjpronk

PRESENTATION

AGENDA

INTRODUCTION

VIRTUAL REALITY

VIRTUAL REALITY & JAVA

ALTERNATIVES

PRESENTER

ALEXANDER CHATZIZACHARIAS

MASTER IN GAME STUDIES

SOFTWARE ENGINEER @ CGI

JAVAONE ROCKSTAR 2016

ABOUT ALEXANDER

CGI

jdriven

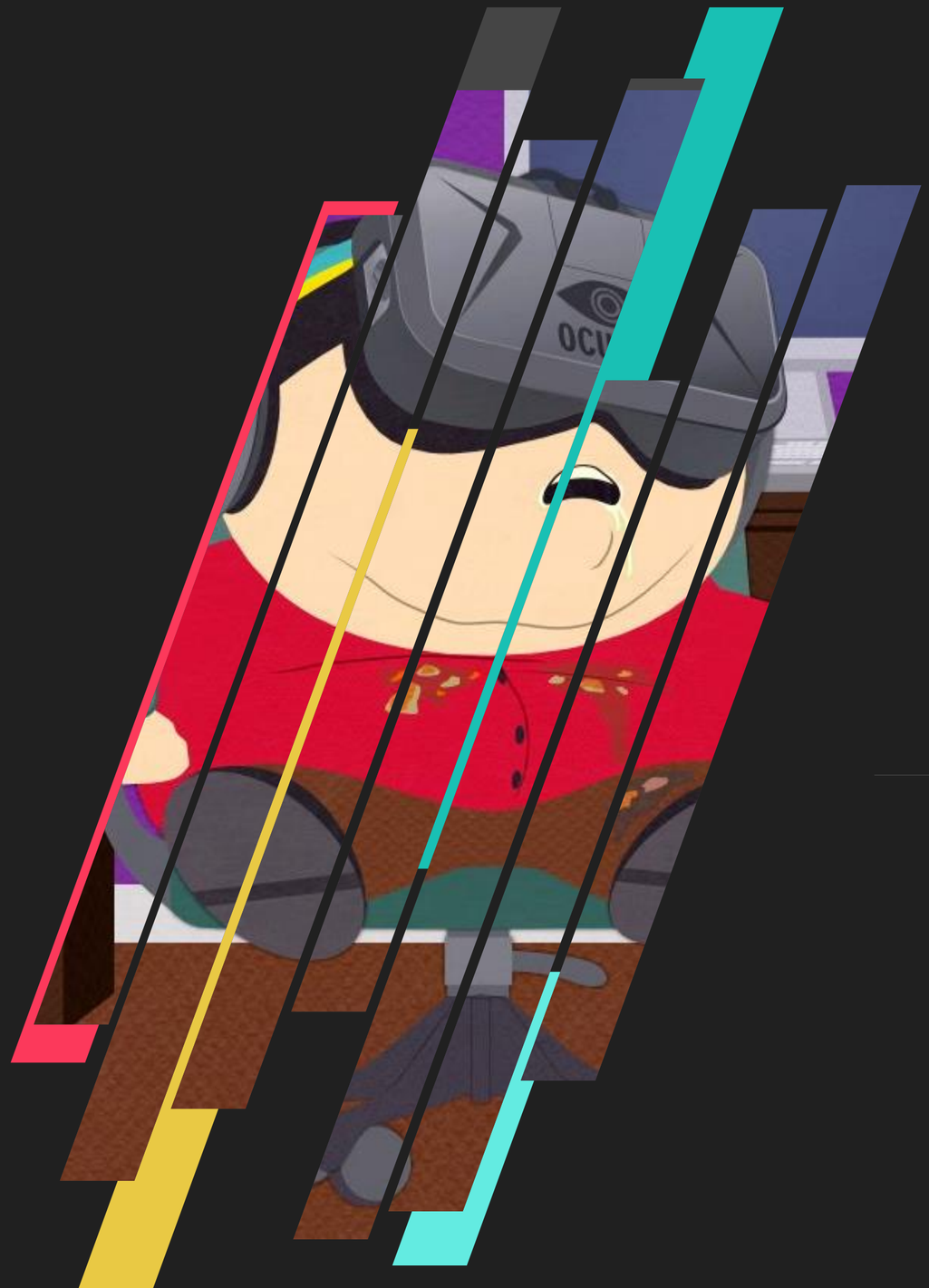
PRESENTER

ERIK PRONK

- BACHELOR IN FOOD SCIENCE
- 2 TIMES MARATHON FINISHER
- SOFTWARE ENGINEER @ JDRIVEN

ABOUT ERIK

jdriven



Virtual Reality

WHAT IS VIRTUAL REALITY?

”The computer-generated simulation of a 3D image or environment that can be interacted with in a seemingly real or physical way by a person using special electronic equipment, such as a helmet with a screen inside or gloves fitted with sensors.”

HISTORY OF VR

STEREOSCOPIC PHOTOS & VIEWERS

1838

01

02

1929

LINK TRAINER, THE FIRST FLIGHT SIMULATOR

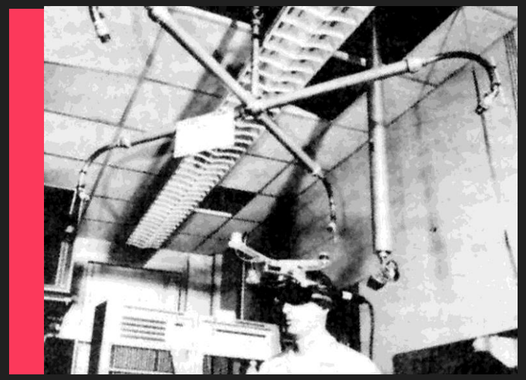
MORTON HEILIG’S SENSORAMA

1950s

03

04

1968



SWORD OF DAMOCLES

VIRTUAL REALITY, THE NAME WAS BORN

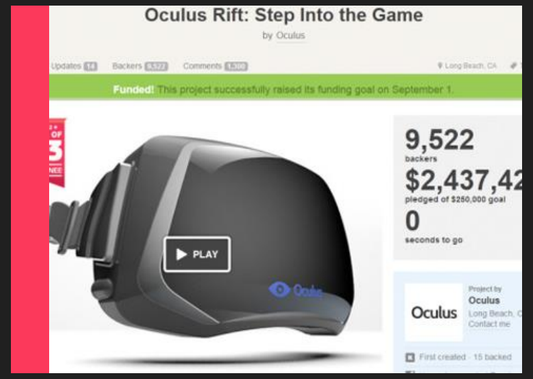
1987

05

06

1995

NINTENDO VIRTUAL BOY



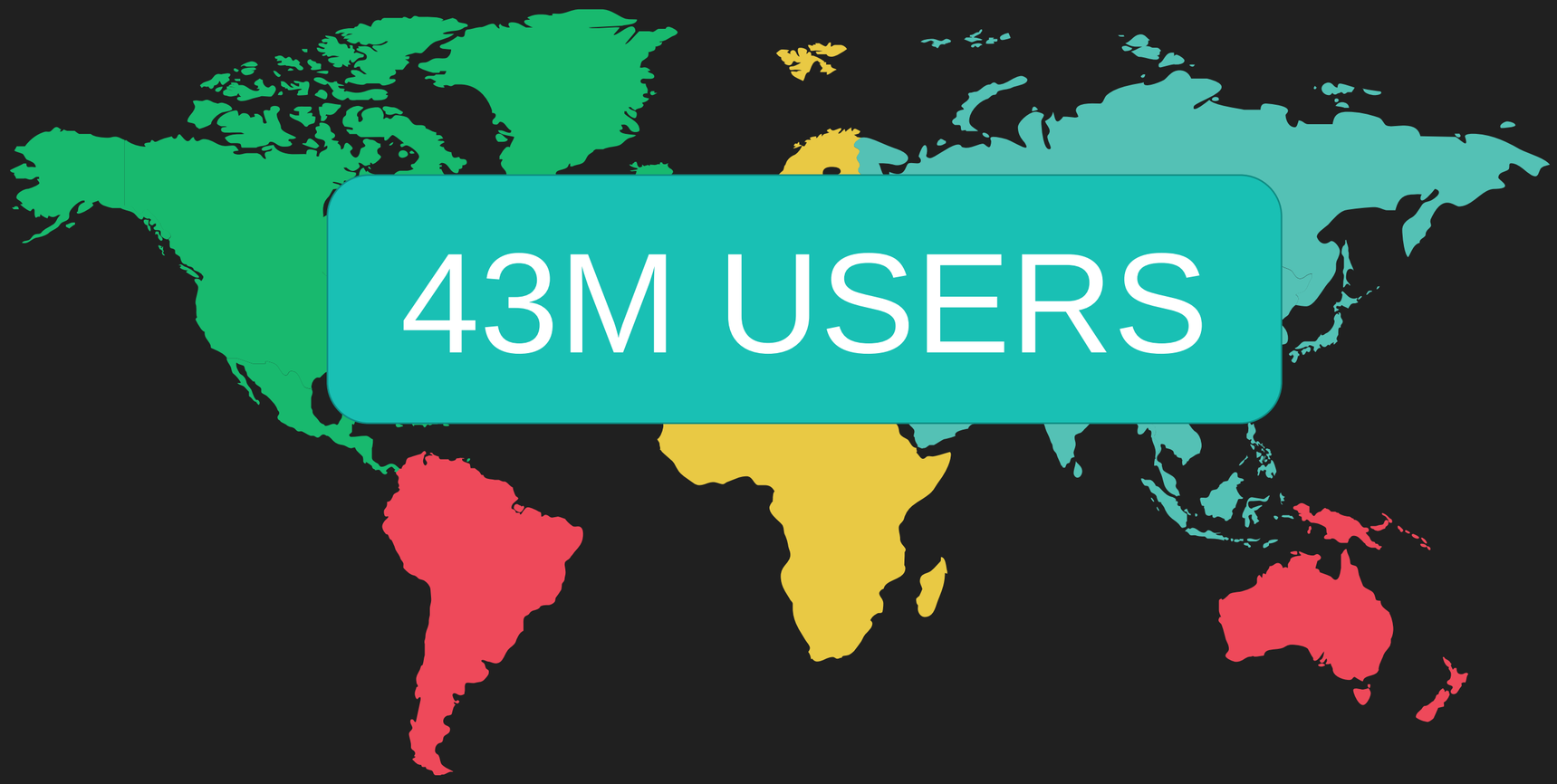
OCULUS RIFT KICK STARTER

2012

07

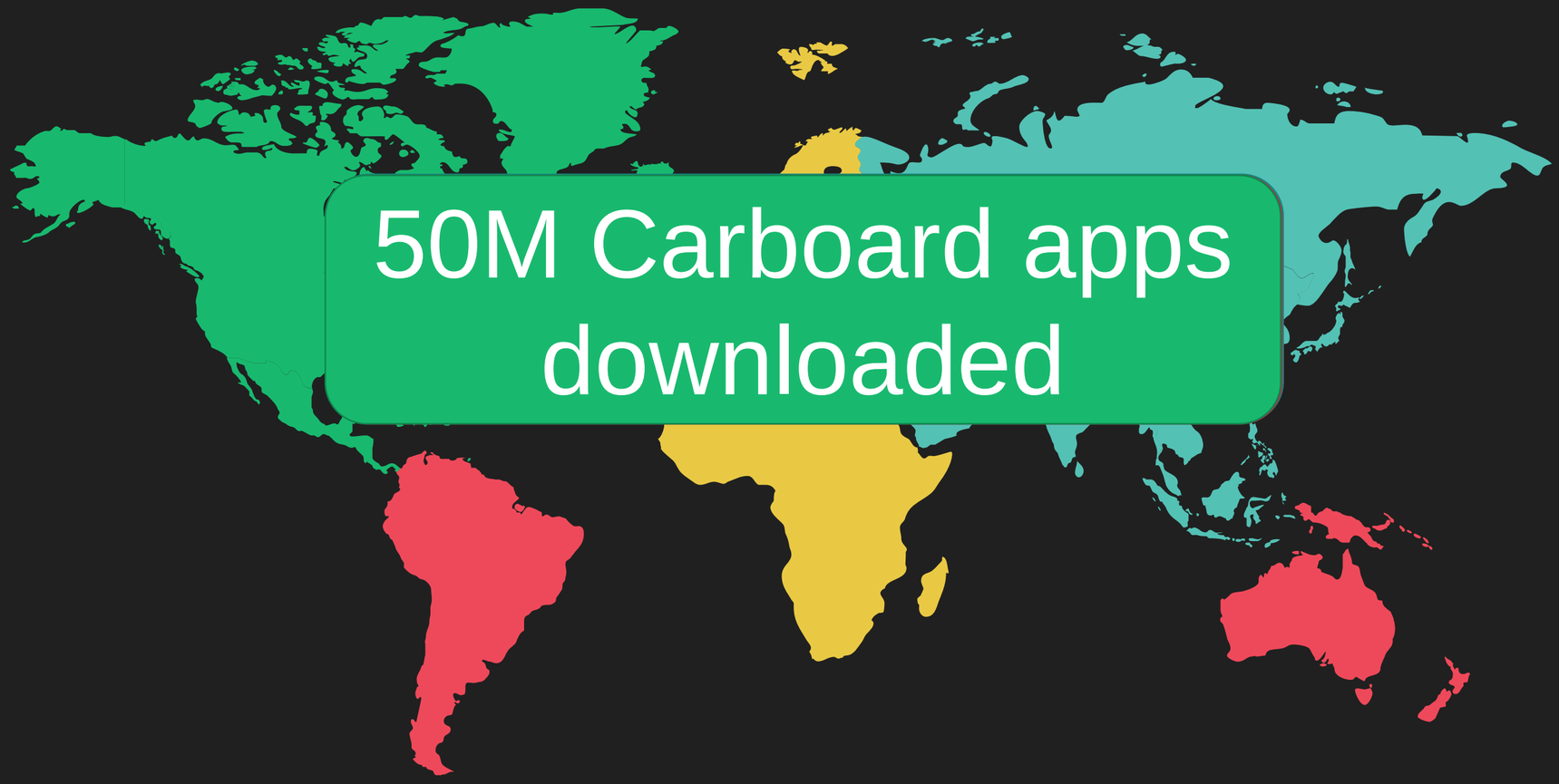
2016

VIRTUAL REALITY IS HOT



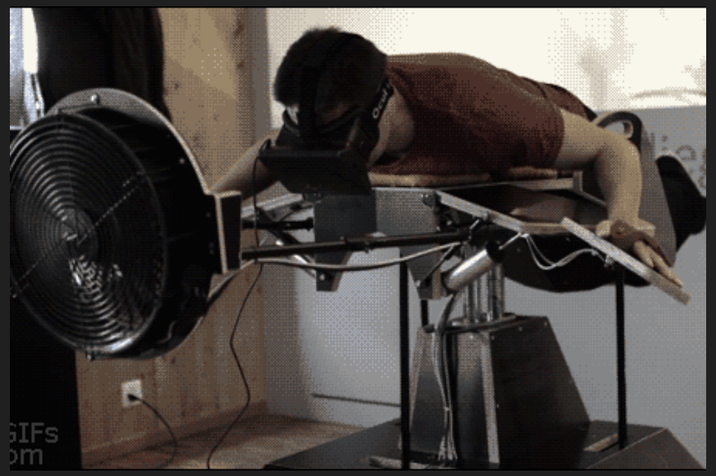
VIRTUAL REALITY IS HOT

VIRTUAL REALITY IS HOT



BUT WHY?

IT IS NEW



IT WORKS

FUN & IMMERSIVE

HOW TO ENTER VR

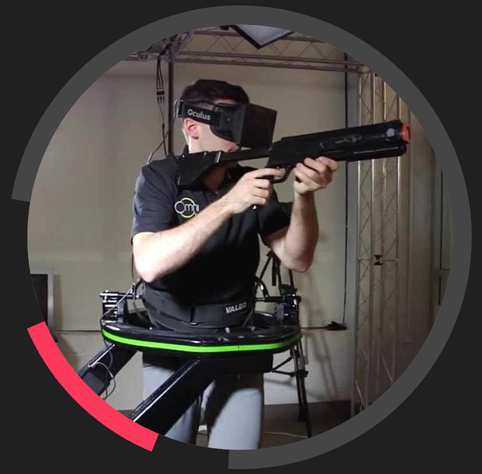
HTC VIVE

OCULUS RIFT

PLAYSTATION VR

GEAR VR, CARDBOARD

COMMON USES



GAMES

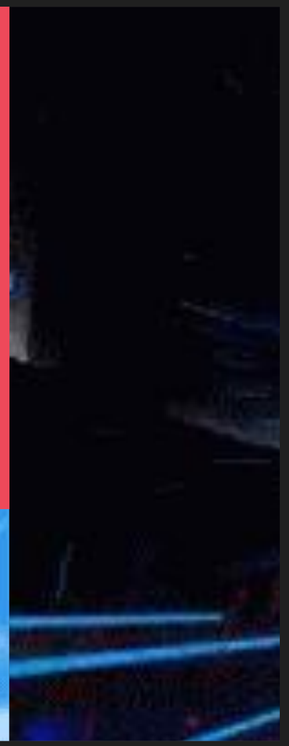
SIMULATION

HEALTH CARE

360°

GAMES







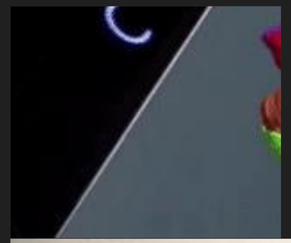


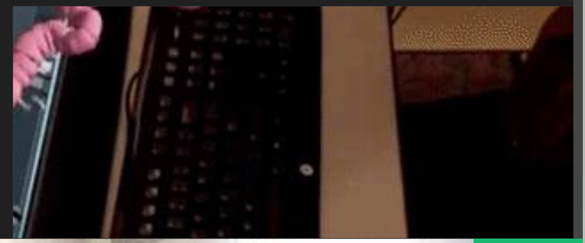


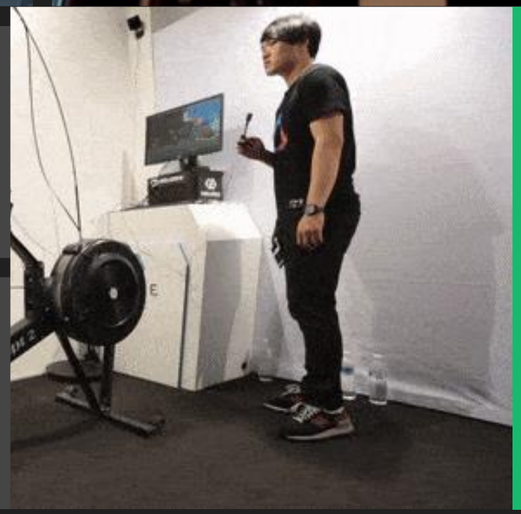
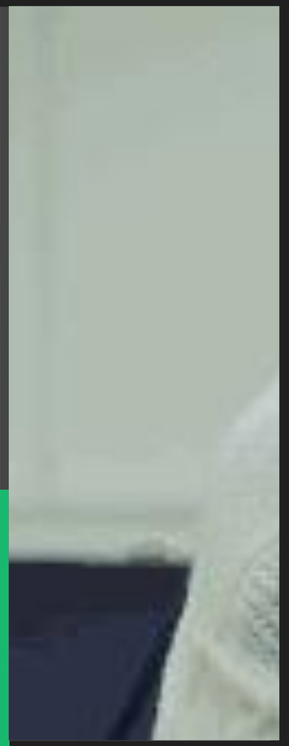
SIMULATION

HEALTH CARE







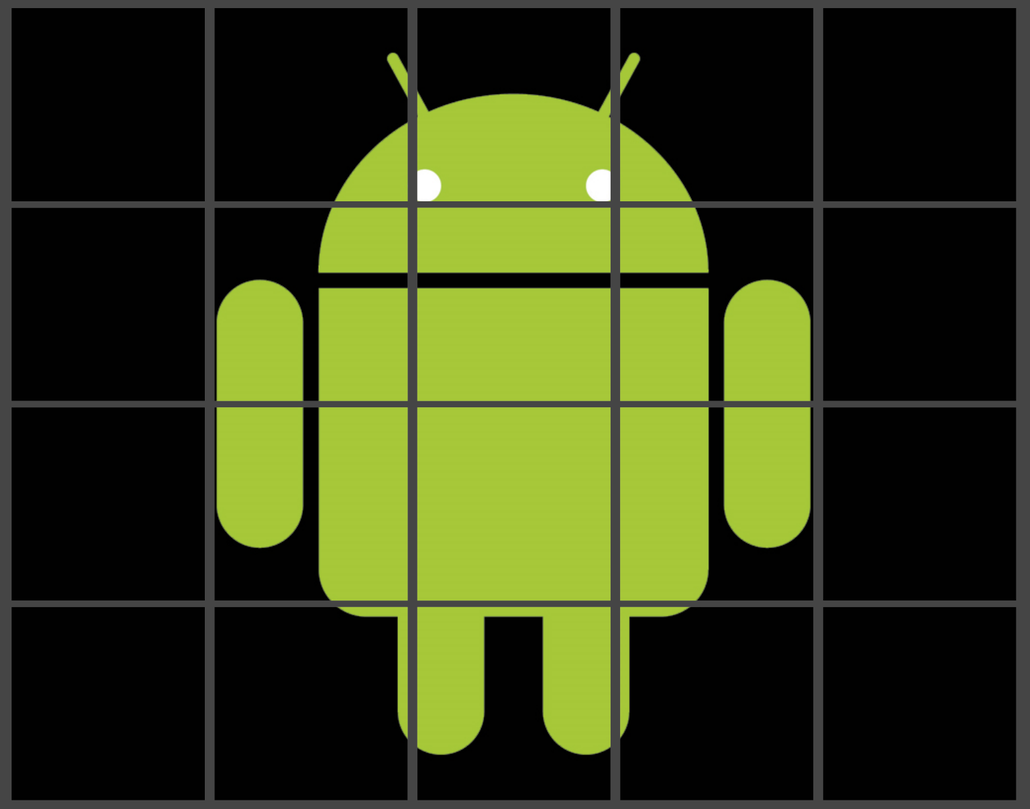





360°

VR & Java

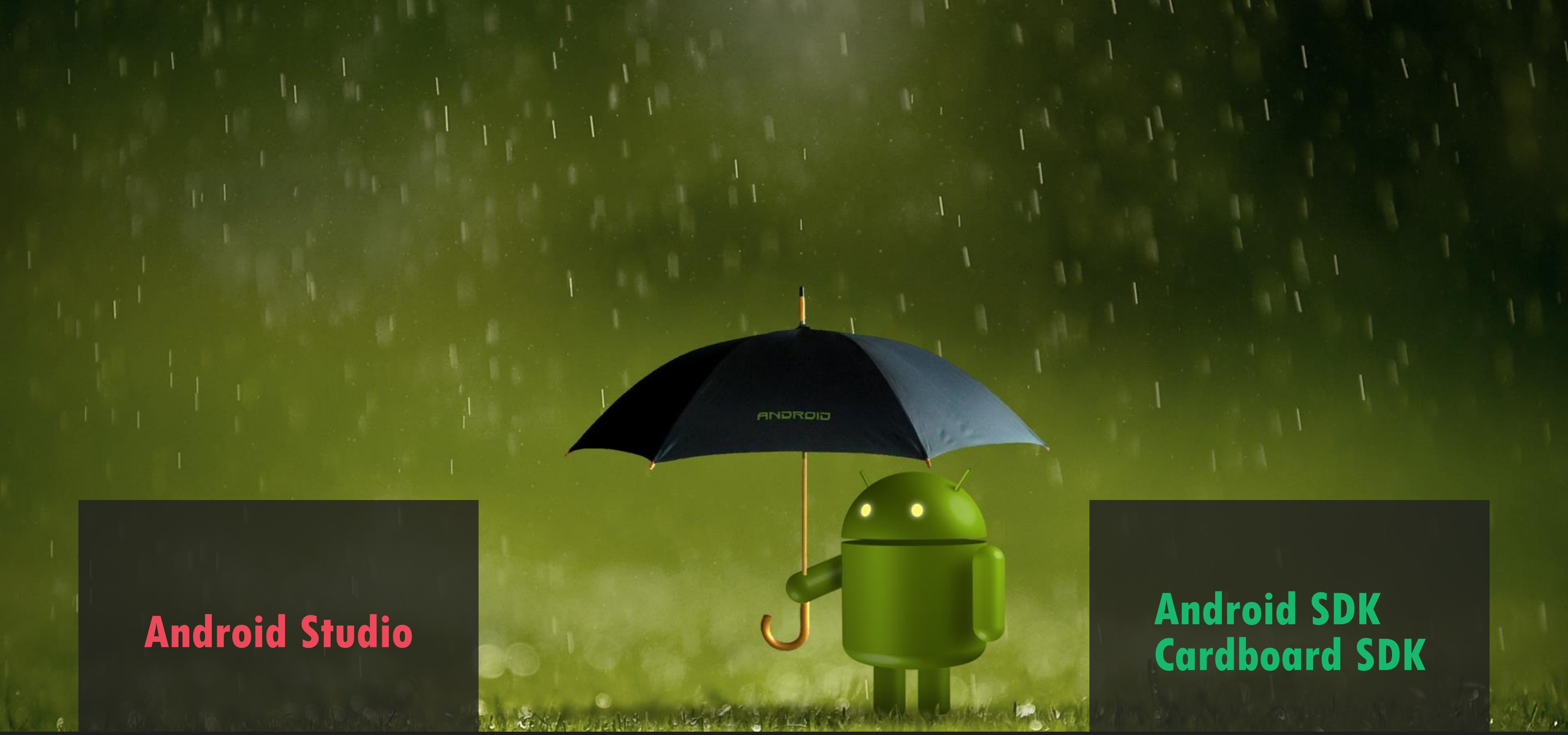
Coding VR in Java

HEAD MOUNTED DISPLAYS



ANDROID SMARTPHONES

ANDROID OPENGL NATIVE



Android Studio

Android SDK
Cardboard SDK

ANDROID OPENGL NATIVE

Android PRO'S

01 CODE IN JAVA

02 PERFORMANCE

03 COMMUNITY

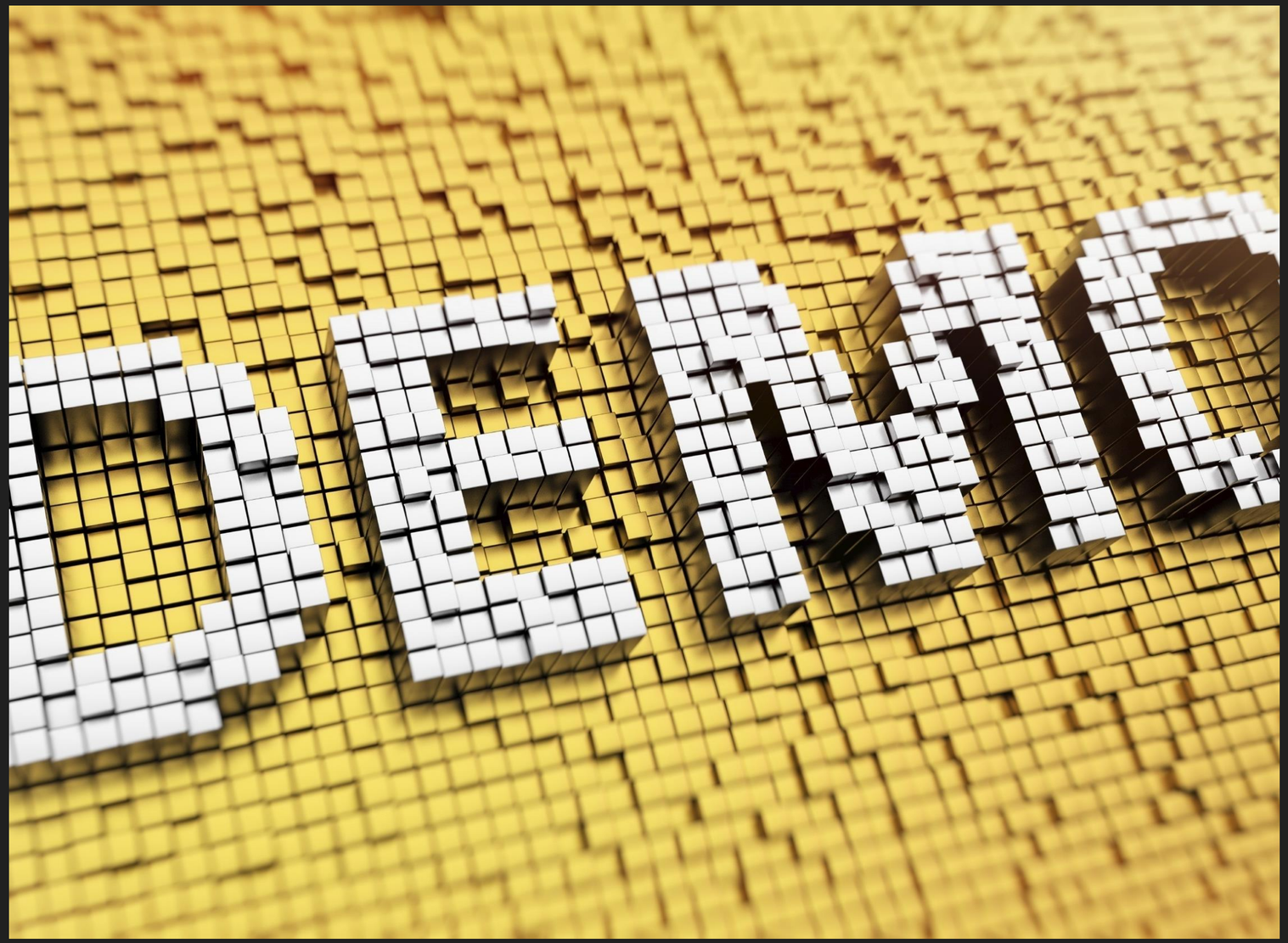
Android CON'S

01 ONLY SMARTPHONES

02 LIMITED BY PLATFORM

03 OPENGL FOR 3D

Demo



360 IMAGES

Xml view definition

Layout XML

```
<com.google.vr.sdk.widgets.pano.VrPanoramaView
    android:id="@+id/pano_view"
    android:layout_margin="5dip"
    android:layout_width="match_parent"
    android:layout_height="250dip"/>
```

360 IMAGES

com.google.vr.sdk.widgets.pano.VrPanoramaView

VrPanoramaView

```
panoWidgetView = (VrPanoramaView) findViewById(R.id.pano_view);

    Options panoOptions = new Options();
    panoOptions.inputType = Options.TYPE_STEREO_OVER_UNDER;
    //panoOptions.inputType = Options.TYPE_MONO;
    panoWidgetView.loadImageFromBitmap(bitmap,panoOptions);
```

Xml view definition

Layout XML

```
<com.google.vr.sdk.widgets.video.VrVideoV
iew
    android:id="@+id/video_view"
    android:layout_width="match_parent"
    android:layout_height="250dip"/>
```

360 VIDEOS

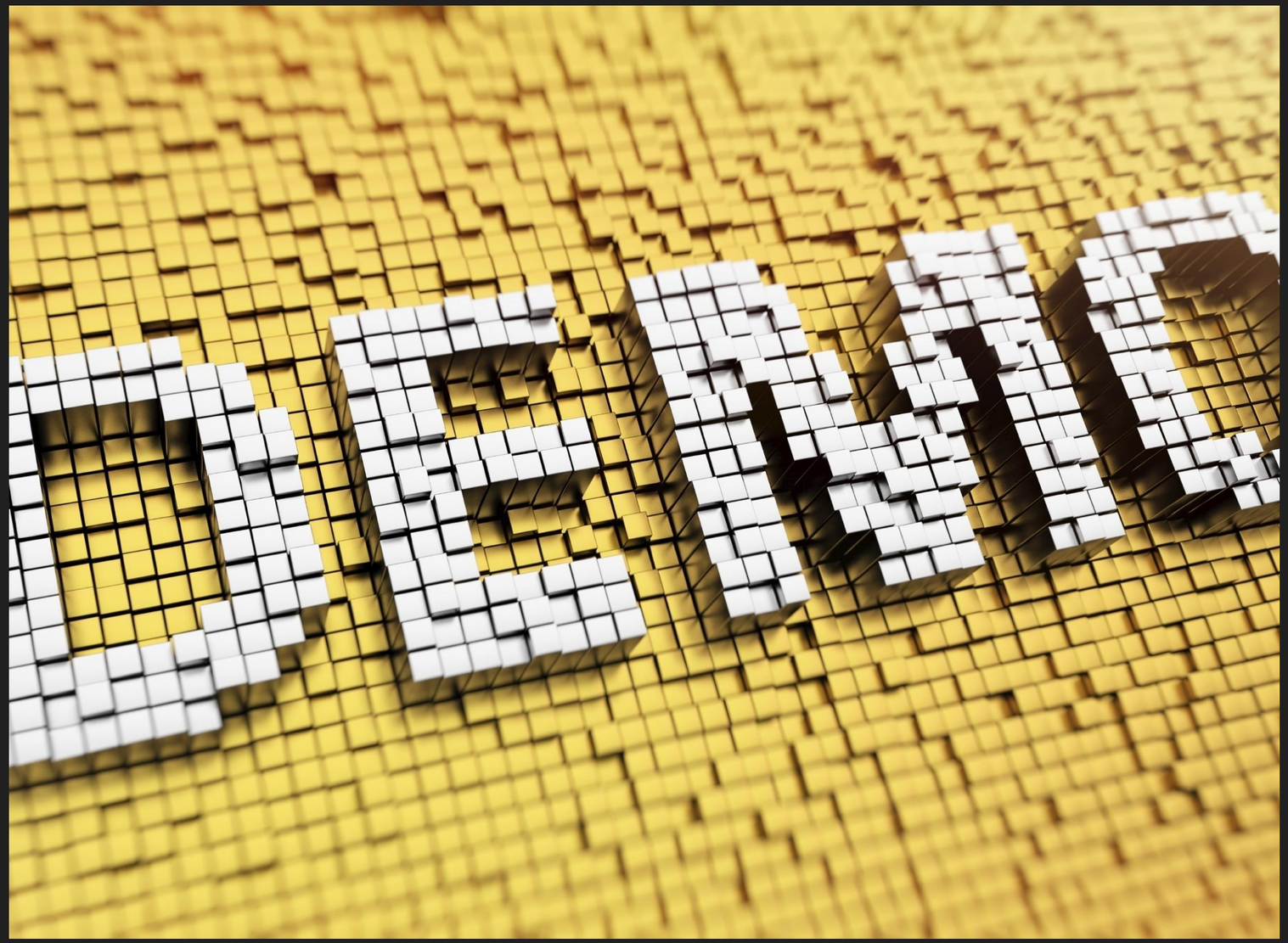
com.google.vr.sdk.widgets.video.VrVideoView

VrVideoView

```
videoWidgetView = (VrVideoView) findViewById(R.id.video_view);

 VrVideoView.Options options = new VrVideoView.Options();
 options.inputType = Options.TYPE_MONO;
 try {
     videoWidgetView.loadVideoFromAsset(”video.mp4", options);
     videoWidgetView.playVideo();
 }catch(Exception e){
     Log.e(TAG,”This is an easter egg”);
 }
```

Demo



Xml view definition

Layout XML

```
<com.google.vr.sdk.base.GvrView
    android:id="@+id/gvr_view"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true" />
```

com.google.vr.sdk.base

GvrActivity & StereoRenderer

public class DemoMainActivity extends GvrActivity implements GvrView.StereoRenderer

ENABLING ANDROID VR

com.google.vr.sdk.base

GvrActivity & StereoRenderer

```
public void onCreate(Bundle savedInstanceState)
{
    setContentView(R.layout.main);
    setGvrView(gvrView);
}
```

com.google.vr.sdk.base

GvrActivity & StereoRenderer

```
public void onSurfaceCreated(EGLConfig config) {
    GLES20.glClearColor(0.4f, 0.8f, 1.0f, 0.5f);
}
```

com.google.vr.sdk.base

GvrActivity & StereoRenderer

```
public void onNewFrame(HeadTransform headTransform) {
   // Build the camera matrix and apply it to the ModelView.
        Matrix.setLookAtM(camera, 0, 0.0f, 0.0f, CAMERA_Z, 0.0f, 0.0f, 0.0f,
0.0f,    1.0f, 0.0f);
        headTransform.getHeadView(headView, 0);
        headTransform.getEulerAngles(headRotation, 0);
        gvrAudioEngine.setHeadRotation(
         headRotation[0], headRotation[1], headRotation[2], headRotation[3]);
        fps.logFrame();
}
```

com.google.vr.sdk.base

GvrActivity & StereoRenderer

```
public void onDrawEye(Eye eye) {
        // Apply the eye transformation to the camera.
        Matrix.multiplyMM(view, 0, eye.getEyeView(), 0, camera, 0);

        // Set the position of the light
        Matrix.multiplyMV(lightPosInEyeSpace, 0, view, 0,
        LIGHT_POS_IN_WORLD_SPACE, 0);
}
```

ENABLING ANDROID VR

com.google.vr.sdk.base

GvrActivity & StereoRenderer

```
public void onCardboardTrigger() {
        shootLaser();

        // Bzz bzz bzz bzz
        vibrator.vibrate(50);
}
```

DRAWING 3D

android.opengl.GLES20

GLES20

VERTEX DATA (position in a 3D space)
COLOR DATA (RGBA data)
NORMAL DATA (for lighting calculations)

DRAWING 3D

```
/**
 * Contains vertex, normal and color data. This stuff makes me really sad :(
 */
public final class WorldLayoutData {

    public static final float[] CUBE_COORDS = new float[] {
            // Front face
            -1.0f, 1.0f, 1.0f,
            -1.0f, -1.0f, 1.0f,
            1.0f, 1.0f, 1.0f,
            -1.0f, -1.0f, 1.0f,
            1.0f, -1.0f, 1.0f,
            1.0f, 1.0f, 1.0f,

            // Right face
            1.0f, 1.0f, 1.0f,
            1.0f, -1.0f, 1.0f,
            1.0f, 1.0f, -1.0f,
            1.0f, -1.0f, 1.0f,
            1.0f, -1.0f, -1.0f,
            1.0f, 1.0f, -1.0f,

            // Back face
            1.0f, 1.0f, -1.0f,
            1.0f, -1.0f, -1.0f,
            -1.0f, 1.0f, -1.0f,
            1.0f, -1.0f, -1.0f,
            -1.0f, -1.0f, -1.0f,
            -1.0f, 1.0f, -1.0f,

            // Left face
            -1.0f, 1.0f, -1.0f,
            -1.0f, -1.0f, -1.0f,
            -1.0f, 1.0f, 1.0f,
            -1.0f, -1.0f, -1.0f,
            -1.0f, -1.0f, 1.0f,
            -1.0f, 1.0f, 1.0f,
```

MOVING 3D OBJECTS

android.opengl.Matrix

Matrix

```
float[] modelLaser = new float[16];
Matrix.setIdentityM(modelLaser,0);
Matrix.rotateM(modelLaser, 0, pitch, 0.0f, 0.0f, 1.0f);
Matrix.rotateM(modelLaser, 0, yaw, 0.0f, 1.0f, 0.0f);
//Every frame
Matrix.translateM(modelLaser, 0, 0.5f, 0, 0);
```

MOVING 3D OBJECTS

android.opengl.Matrix

Matrix

```
boolean checkZ = modelCube[14] - 0.5f < modelLaser[14] && modelCube[14] +
0.5f > modelLaser[14];
boolean checkY = modelCube[13] - 0.5f < modelLaser[13] && modelCube[13] +
0.5f > modelLaser[13];
if(checkZ && checkY){
        // COLLISION!
}
```

com.google.vr.sdk.audio.GvrAudioEngine

GvrAudioEngine

```
gvrAudioEngine = new GvrAudioEngine(this,
GvrAudioEngine.RenderingMode.BINAURAL_HIGH_QUALITY);

gvrAudioEngine.preloadSoundFile(EXPLOSION_SOUND_FILE);
sourceId = gvrAudioEngine.createSoundObject(EXPLOSION_SOUND_FILE);
gvrAudioEngine.setSoundObjectPosition(
sourceId, modelPosition[0], modelPosition[1], modelPosition[2]);

gvrAudioEngine.playSound(successSourceId, false);
```

https://github.com/Rajawali/Rajawali

Android Studio

Android SDK &
Rajawali

ANDROID WITH RAJAWALI

Xml view definition

Layout XML

```
<com.google.vr.sdk.base.GvrView
    android:id="@+id/gvr_view"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true" />
```

ENABLING RAJAWALI VR

org.rajawali3d.vr.VRActivity

VRActivity

```
public void onCreate(Bundle savedInstanceState) {
        mRenderer = new
RajawaliVRExampleRenderer(this);
        setRenderer(mRenderer);
}
```

ENABLING RAJAWALI VR

org.rajawali3d.vr.VRActivity

VRActivity

```
public void
onCardboardTrigger() {
        mRenderer.spawnBullet();
}
```

USING RAJAWALI VR

org.rajawali3d.vr.renderer.VRRenderer

VRRenderer

```
public void initScene() {
    DirectionalLight light = new DirectionalLight(0.2f, -1f, 0f);
    getCurrentScene().addLight(light);
    getCurrentCamera().setFarPlane(1000);
    getCurrentScene().setBackgroundColor(0xdddddd);
}
```

USING RAJAWALI VR

org.rajawali3d.vr.renderer.VRRenderer

VRRenderer

```
public void onRender(long elapsedTime, double deltaTime)
{
        super.onRender(elapsedTime, deltaTime);
        boolean isLookingAt = isLookingAtObject(cube);
}
```

SPAWNING AND USING 3D OBJECTS

org.rajawali3d.primitives

PRIMITIVES

```
bullet = new Sphere(2);
Material cubeMat = new Material();
bullet.setMaterial(cubeMat);
bullet.setColor(Color.YELLOW);
bullet.setPosition(getCurrentCamera().getPosition().clone());
bullet.setRotation(getCurrentCamera().getOrientation().clone
());
getCurrentScene().addChild(bullet);
```

SPAWNING AND USING 3D OBJECTS

org.rajawali3d.primitives

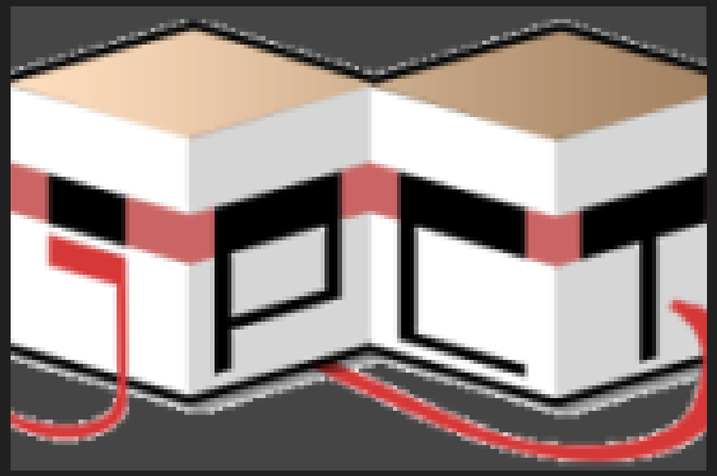
PRIMITIVES

```
//every frame
bullet.moveForward(-5);
```

OTHER FRAMEWORKS

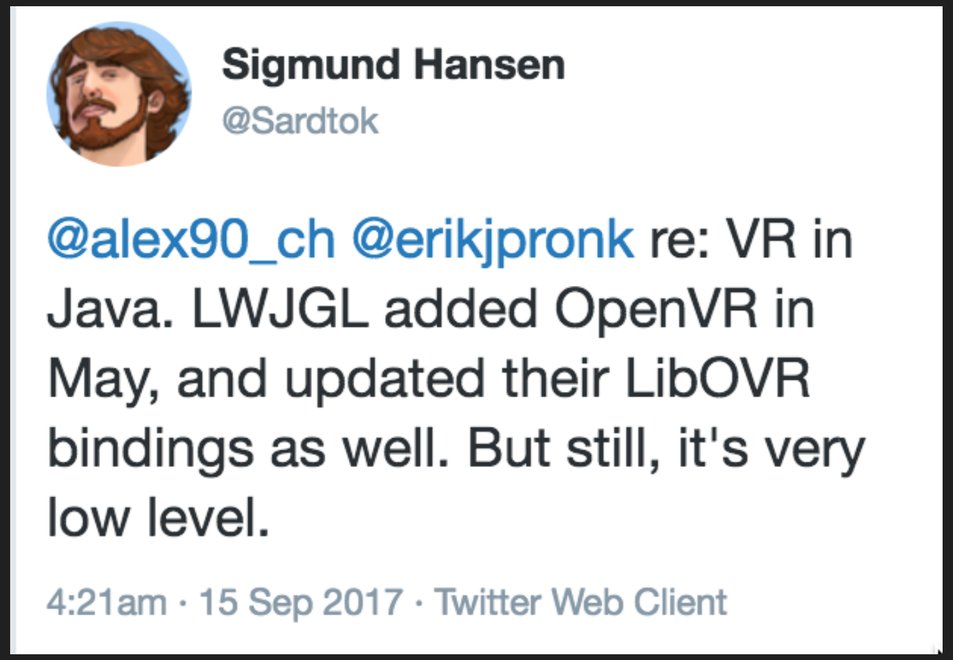
smartGL

libGDX



jPCT-AE

OTHER FRAMEWORKS

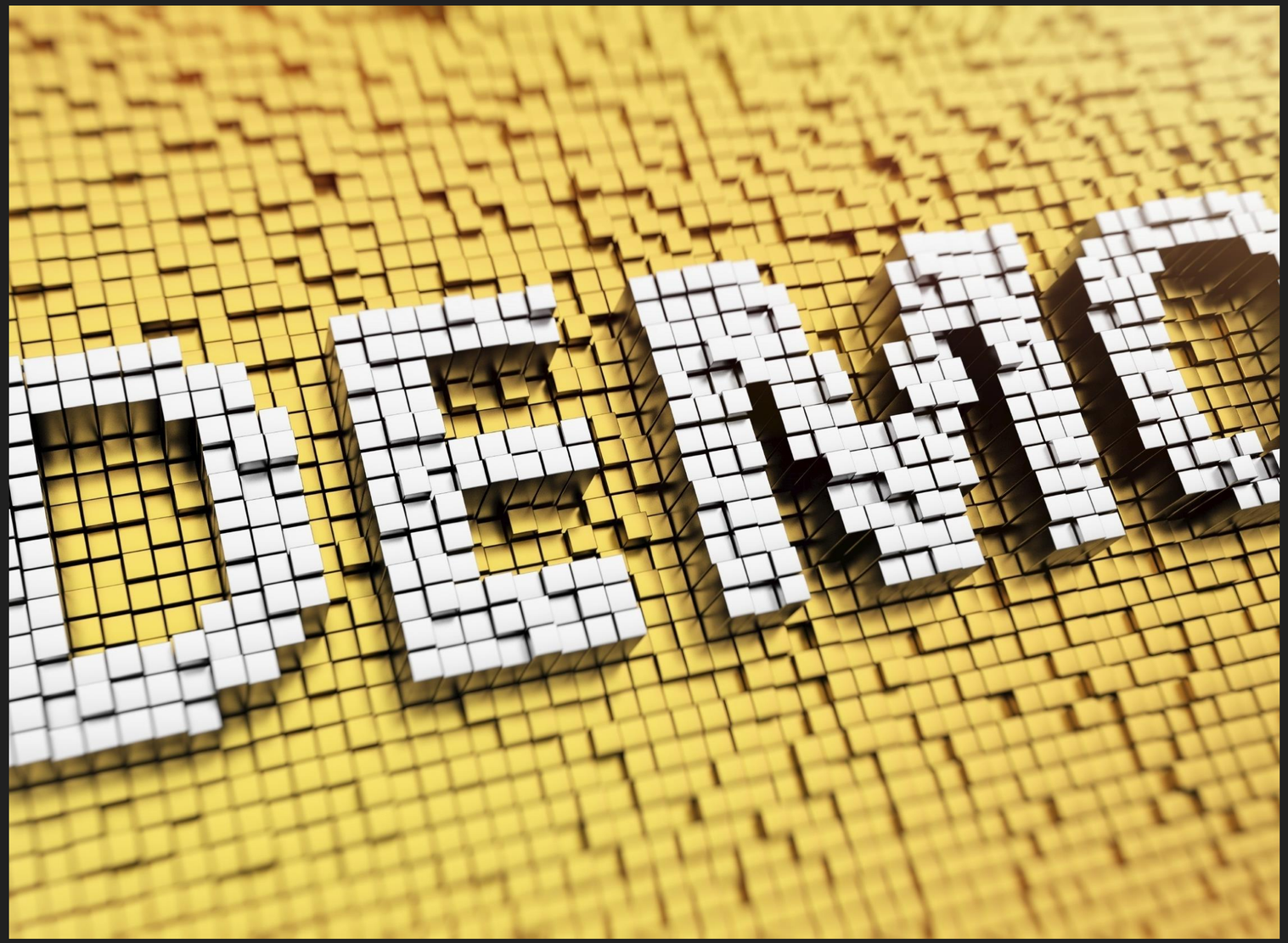


ALTERNATIVES

WebVR

Virtual Reality from the browser

Demo



SCRIPT IMPORT

AFRAME Framework

```
<script src="https://aframe.io/releases/0.2.0/aframe.min.js"></script>
```

External libraries

```
<script src="https://rawgit.com/ngokevin/aframe-layout-
component/master/dist/aframe-layout-component.min.js"></script>
<script src="https://rawgit.com/ngokevin/aframe-template-
component/master/dist/aframe-template-component.min.js"></script>
```

REGISTER COMPONENTS

AFRAME.

```
registerComponent('click-listener', {
        init: function() {
                var el = this.el;
                window.addEventListener('click', function() {
                        el.emit('click', null, false);
                        document.getElementById('gunsound').play();
                });
        }
});
```

AFRAME HTML

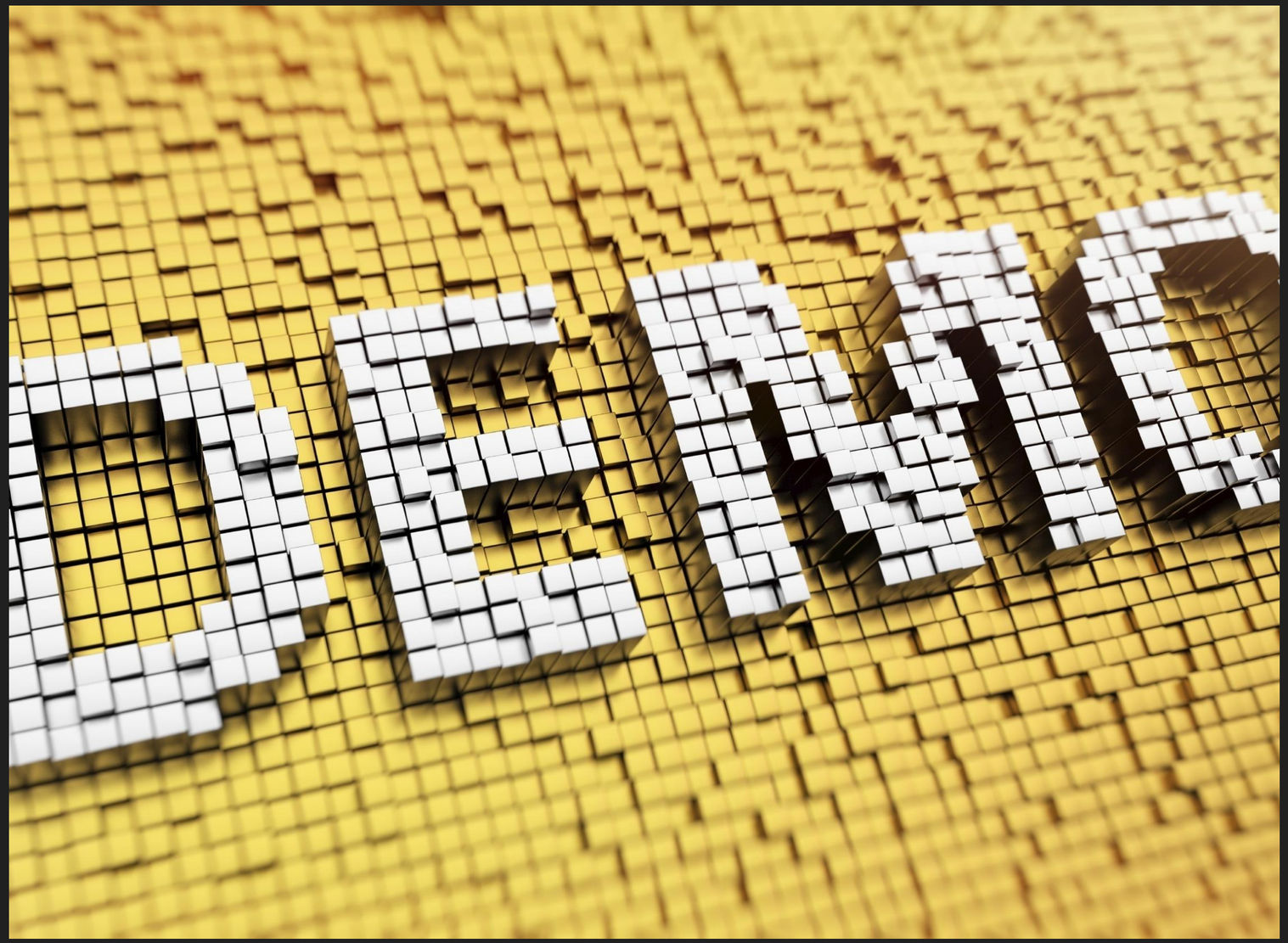
AFRAME

```
<a-scene>
        <a-assets> </a-assets>
        <a-entity id="enemy-sprite" class="enemy">
                <a-box src="#boxTexture_jdriven" position="-1 0.6 -15" rotation="0 45 0”>
                        <a-animation attribute="opacity" begin="collider-hit" dur="1000" ease="linear" >
                        </a-animation>
                 </a-box>
        </a-entity>
        <a-entity position="0 0 0" id="camera" camera look-controls="enabled: true" acceleration: 100">
        </a-entity>
        <a-sky src="#skyTexture"></a-sky>
</a-scene>
```

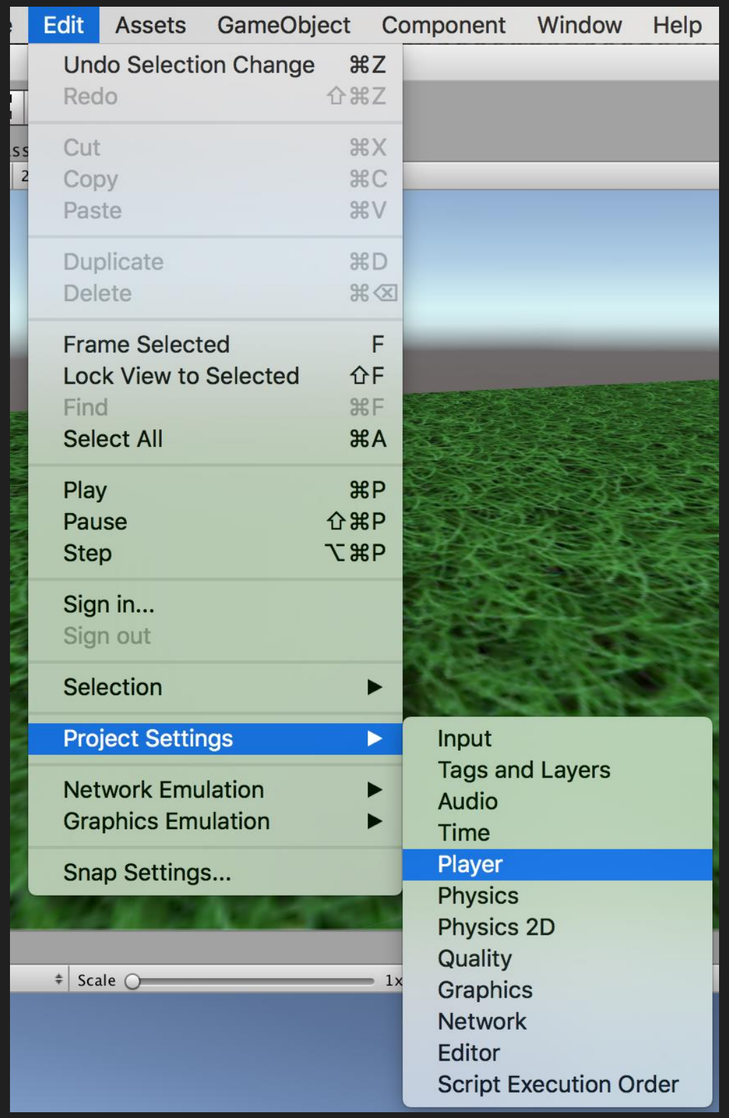
UNITY3D

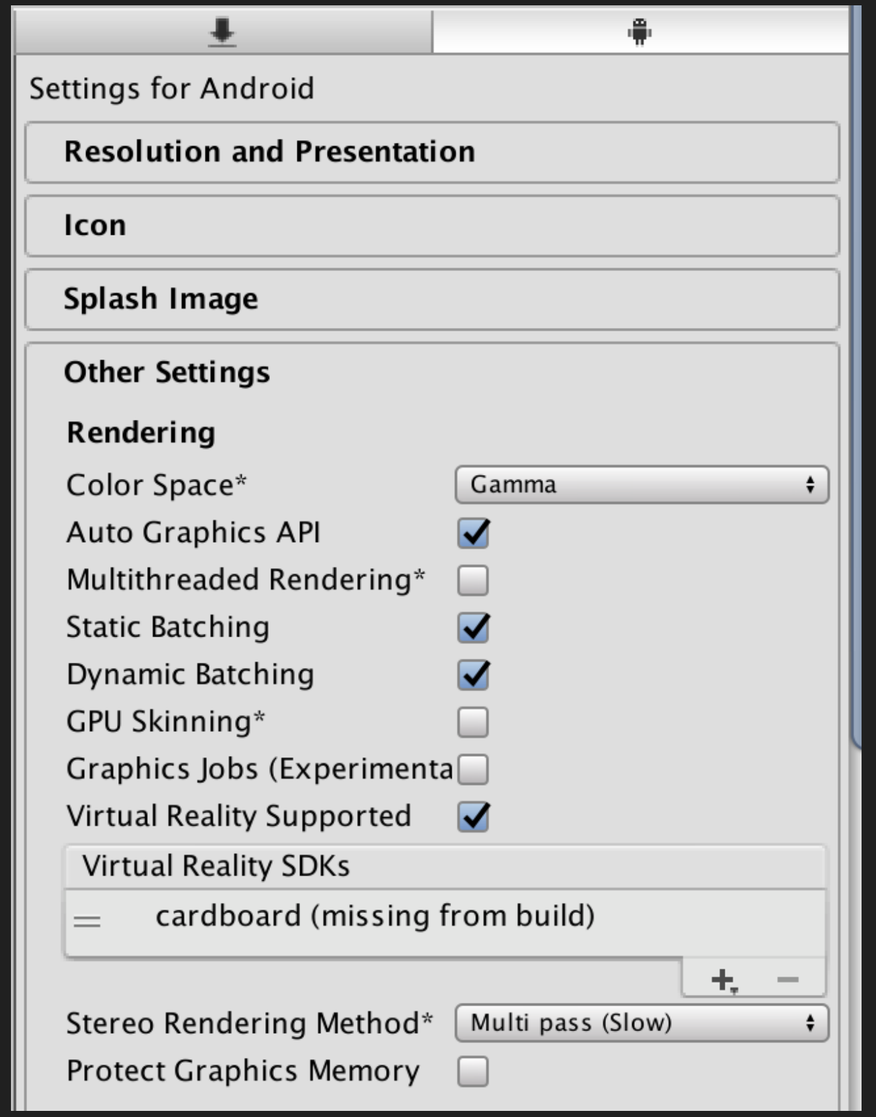
Most used tool for VR

Demo



ENABLING Unity VR





Spawning and moving objects

```
using UnityEngine;

UnityEngine

void Update () {

        if(Input.GetMouseButtonDown(0)) {
                Instantiate(bullet,
                transform.position + transform.right * 0.1f,
                transform.rotation);
        }
}
```

Spawning and moving objects

```
using UnityEngine;

UnityEngine

 transform.position += transform.forward * Time.deltaTime *
30;
```

Spawning and moving objects

using UnityEngine;

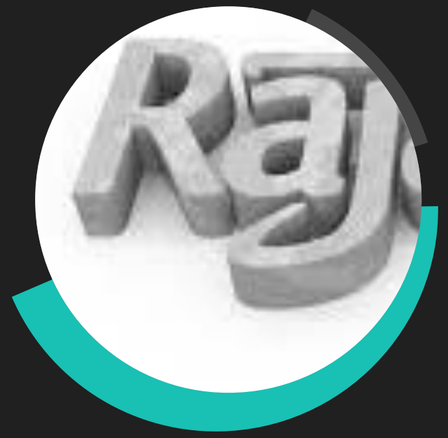
UnityEngine

GetComponent<AudioSource>().Play();

COMPARISSON

45.87MB
85
60
Very relaxed

5.5MB
175
60
Javascript



9.54MB
245
60
Too many…

12.21MB
1071
60
∞

THANKS FOR WATCHING

https://github.com/erikjpronk/javaone-demos

https://github.com/erikjpronk/javaone-demos