

Προγραμματίζοντας
για

Virtual Reality

Γιάννης Τσιομπίκας

< nuclear@member.fsf.org >

(μια πρόχειρη εισαγωγή)

Virtual Reality

- Τι είναι;
 - Immersion
 - Presence



← brain

- Τι απαιτεί;

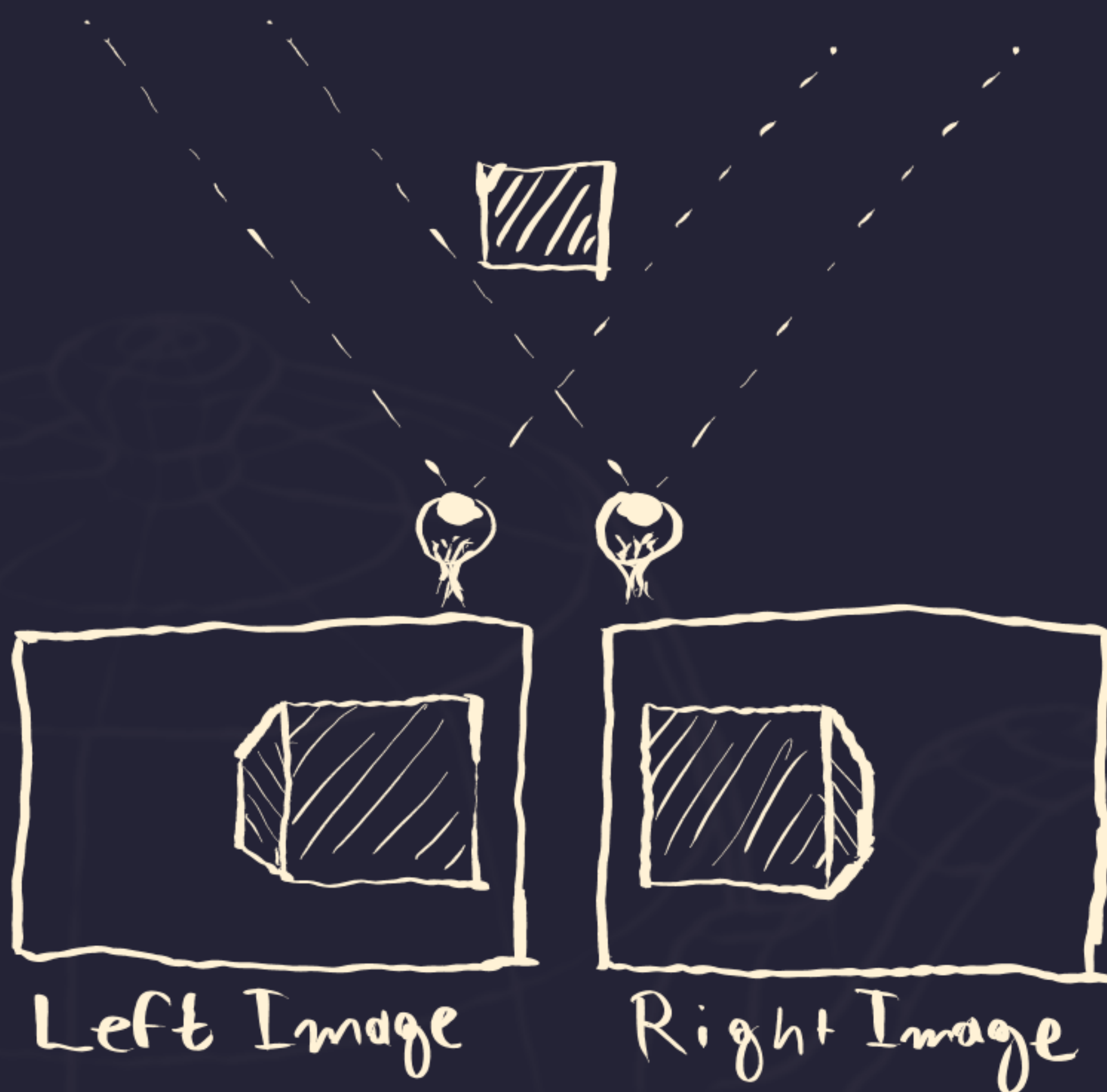
- Stereoscopic rendering
- Μεγάλο οπτικό πεδίο (FOV)
- Head tracking



6 degrees
of freedom

Stereoscopy

- Αντίληψη βάθους από τη διαφορά οπτικής ανάμεσα στα μάτια.

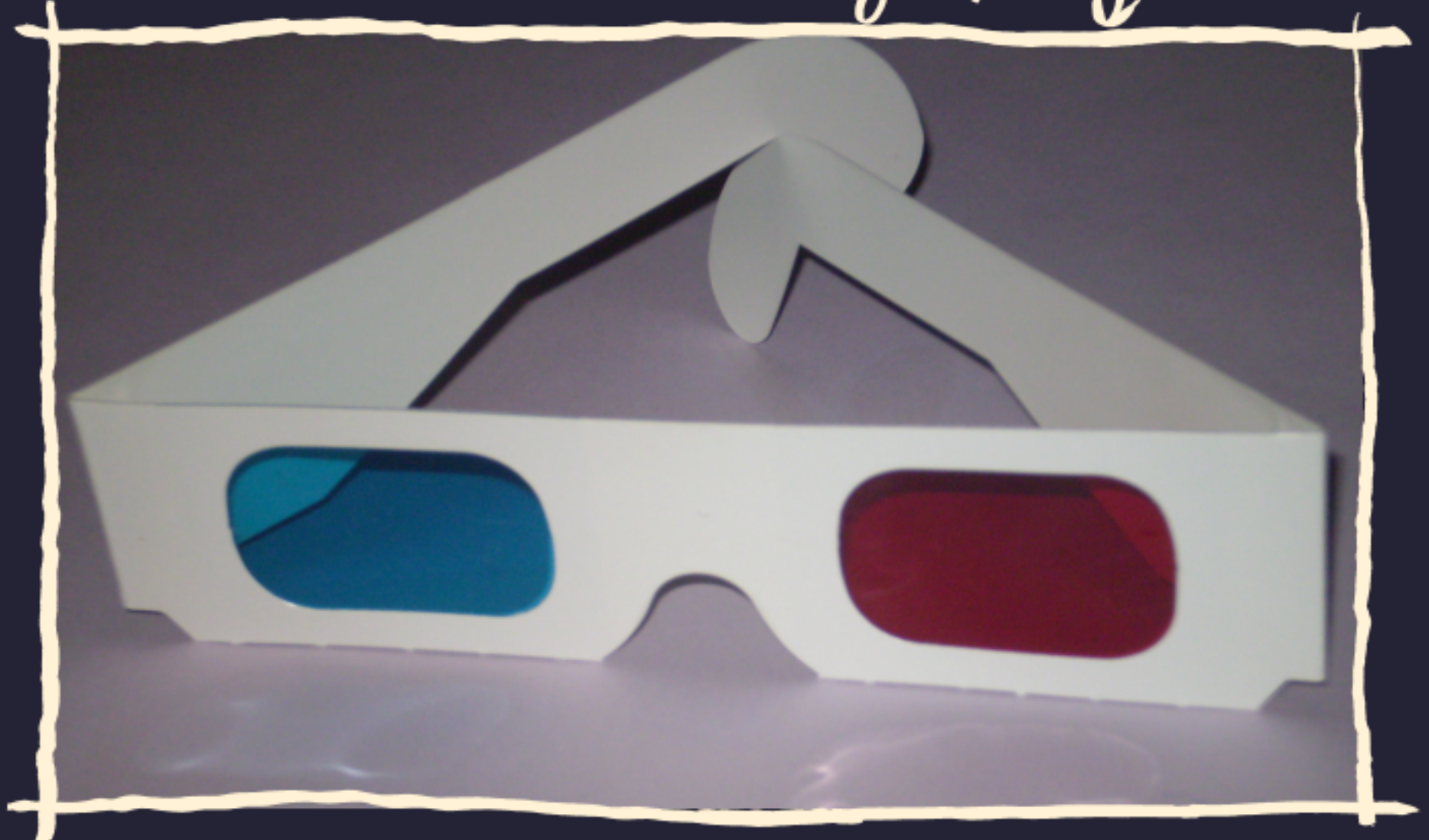


Stereoscopy

Anaglyph ↴



Stereoscope ↴



Shutter glasses ↴



tracking

Inertial

- Accelerometer
- Gyroscope
- Mag. Compass
- + Ευκωδία
- + Fast updates/
Low latency
- Drift

Optical

- Visible Light
- Infrared
- + Ακρίβεια
- Υπολογιστικά ακριβό
- αργά updates/
μεγάλο latency
- απαιτεί setup
στον χώρο

Oculus Rift

DK2



DK1

Oculus Rift - Tracking

IMU



← IR Camera

IR LEDs



Oculus Rift - Rendering

- Get tracking data (θέση/περιστροφή)
- Render 2 views σε ένα texture
- διορθωση παραμόρφωσης φακού } με Shader
- διορθωση chromatic aberration }

Lens
Distortion



Pre-distortion
to
compensate

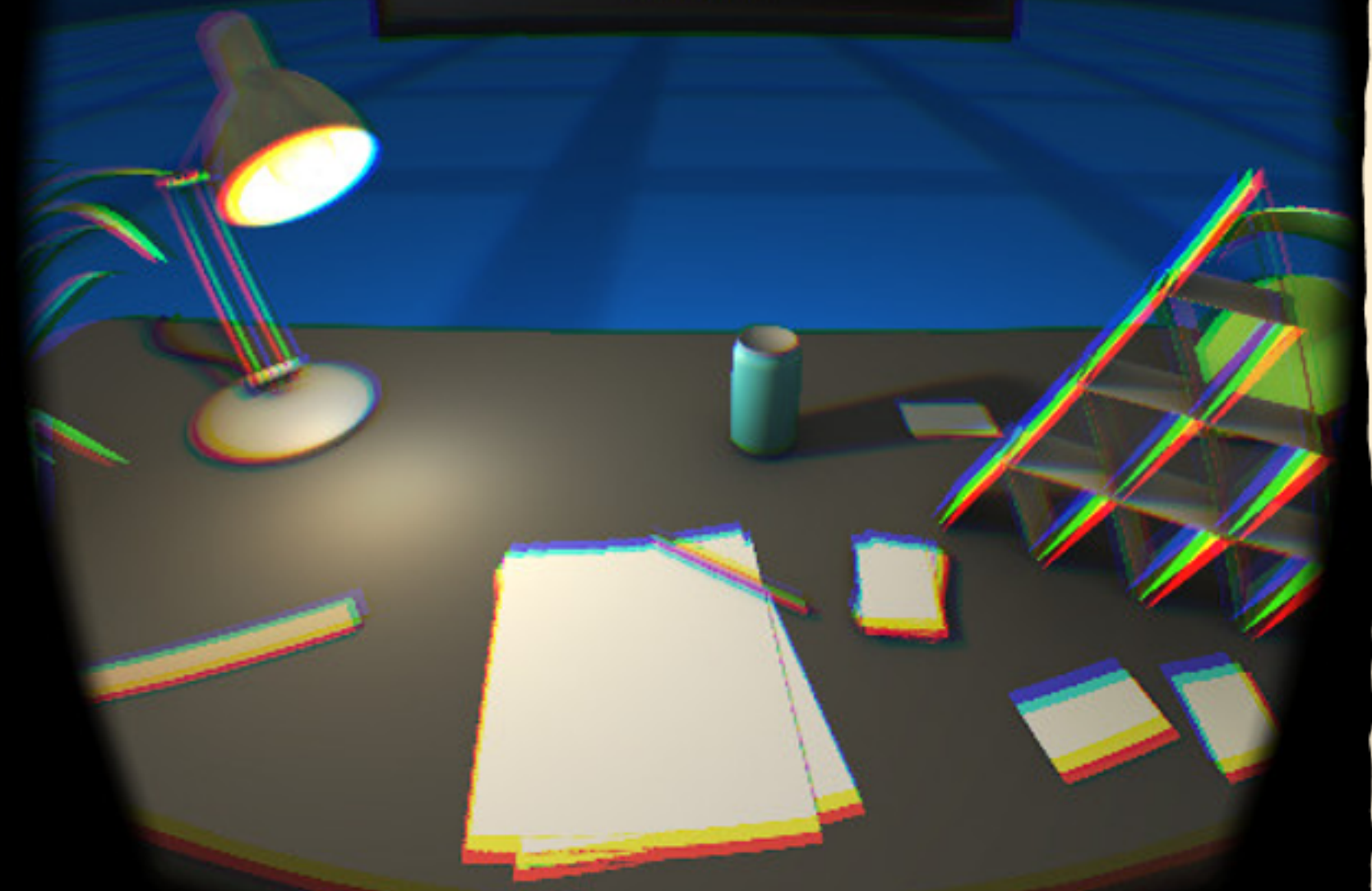
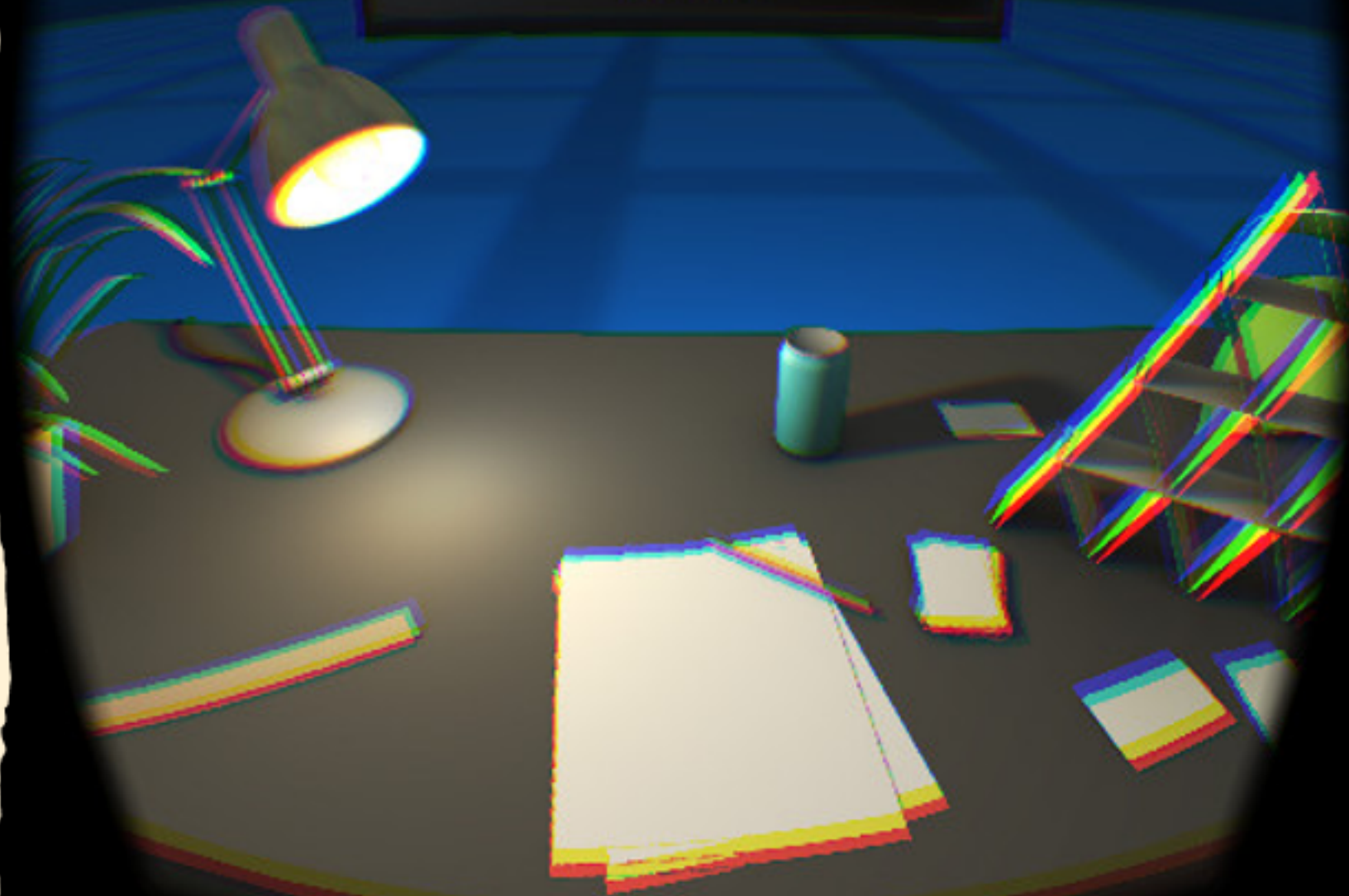
Oculus Rift - Rendering



Position: Sit Stand
Camera Bounds: On Off
Recenter



Position: Sit Stand
Camera Bounds: On Off
Recenter



Oculus Rift in action



Oculus SDK

(libovr / ovr)

- Αναλαμβάνει tracking & διορθώσεις φακού
- proprietary
- Κύρια πλατφόρμα τα windows
Η GNU/Linux έκδοση μένει συχνά πίσω
(0.5.0.1 vs 0.8.0.0)
- Δουλεύει... πλέον

Oculus SDK - initialization

```
ovr_Initialize(0);
ovrHmd hmd = ovrHmd_Create(0);

ovrSizei texsz[2];
texsz[0] = ovrHmd_GetFovTextureSize(hmd, ovrEye_Left, hmd->DefaultEyeFov[0], 1.0);
texsz[1] = ovrHmd_GetFovTextureSize(hmd, ovrEye_Right, hmd->DefaultEyeFov[1], 1.0);

rtarg = create_rtarg(texsz[0].w + texsz[1].w, texsz[0].h);

ovrGLTexture ovr_tex[2];
for(i=0; i<2; i++) {
    ovr_tex[i].OGL.Header.API = ovrRenderAPI_OpenGL;
    ovr_tex[i].OGL.Header.TextureSize.w = rtarg.width;
    ovr_tex[i].OGL.Header.TextureSize.h = rtarg.height;
    ovr_tex[i].OGL.Header.RenderViewport.Pos.x = i ? rtarg.width / 2.0 : 0;
    ovr_tex[i].OGL.Header.RenderViewport.Pos.y = 0;
    ovr_tex[i].OGL.Header.RenderViewport.Size.w = rtarg.width / 2.0;
    ovr_tex[i].OGL.Header.RenderViewport.Size.h = rtarg.height;
    ovr_tex[i].OGL.TexId = rtarg.texid; /* GL texture id */
}

ovrGLConfig ovr_cfg;
ovr_cfg.OGL.Header.API = ovrRenderAPI_OpenGL;
ovr_cfg.OGL.Header.BackBufferSize = hmd->Resolution;
ovr_cfg.OGL.Disp = glXGetCurrentDisplay();
```


Oculus SDK - Rendering

```
ovrHmd_BeginFrame(hmd, 0);  
glBindFramebuffer(GL_FRAMEBUFFER, rtarg.fbo);
```

```
ovrPosef pose[2];  
for(i=0; i<2; i++) {  
    glViewport(i == 0 ? 0 : rtarg.width / 2, 0, rtarg.width / 2, rtarg.height);
```

HMD
optics

```
    ovrMatrix4f proj = ovrMatrix4f_Projection(hmd->DefaultEyeFov[i], 0.5, 500.0, 1);  
    glMatrixMode(GL_PROJECTION);  
    glLoadTransposeMatrixf(proj.M[0]);
```

```
    pose[i] = ovrHmd_GetHmdPosePerEye(hmd, i);  
    glMatrixMode(GL_MODELVIEW);  
    glLoadIdentity();  
    glTranslatef(eye_offset[i].x, eye_offset[i].y, eye_offset[i].z);
```

← get tracking

← θέση ματιού
στο κεφάλι

```
    float rot_mat[16];  
    quat_to_matrix(&pose[i].orientation.x, rot_mat);  
    glMultMatrix(rot_mat);
```

← περιστροφή κεφαλιού
← θέση

```
    glTranslatef(-pose[i].Position.x, -pose[i].Position.y, -pose[i].Position.z);  
    glTranslatef(0, -ovrHmd_GetFloat(hmd, OVR_KEY_EYE_HEIGHT, 1.65), 0);
```

↑ ύψος ματιών χρήστη

```
    draw_scene();  
}
```

```
glBindFramebuffer(GL_FRAMEBUFFER, 0);  
ovrHmd_EndFrame(hmd, pose, &ovr_tex[0].Texture);
```


Alternatives

- Open HMD

(<http://openhmd.net>)

- libgoatvr

(<http://github.com/jtsiomb/libgoatvr>)

Goat VR



- multiple VR backends
 - Oculus SDK 0.8 (Windows)
 - Oculus SDK 0.5 (GNU/Linux)
 - OpenHMD (incomplete)
 - Null (fallback)
- Free software (LGPL v3)

Goat VR - Null backend



Brand-new problems

- Simulation sickness (ναυτία)
 - latency
 - αυτόματη κίνηση (physics, cinematics)
 - μετακίνηση στον χώρο (FPS controls)

Brand-new problems

- Simulation sickness (ναυτία)
 - latency
 - αυτόματη κίνηση (physics, cinematics)
 - μετακίνηση στον χώρο (FPS controls)

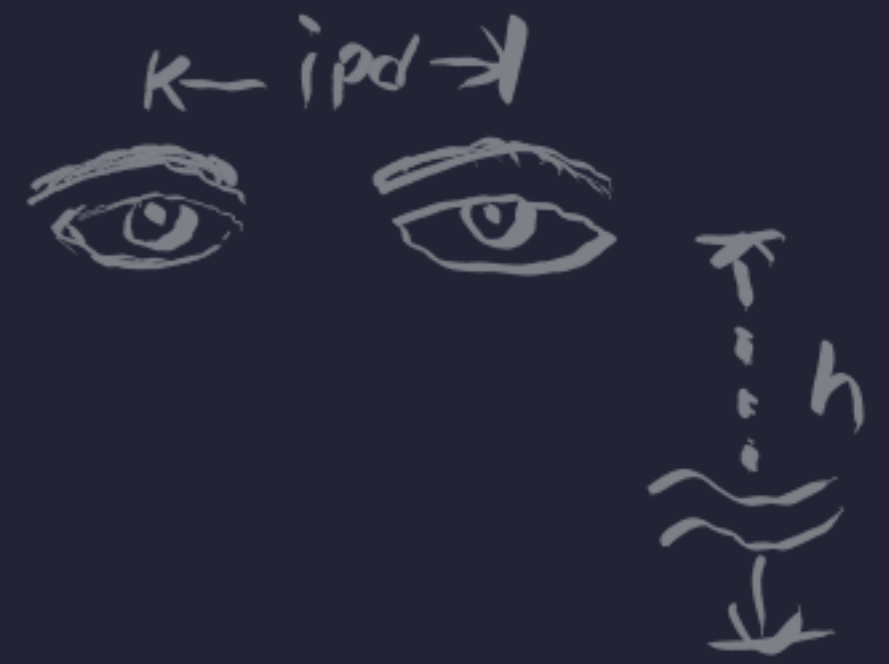
- Αντίληψη κλίμακας
 - inter pupillary distance
 - ύψος ματιών




Brand-new problems

- Simulation sickness (ναυτία)
 - latency
 - αυτόματη κίνηση (physics, cinematics)
 - μετακίνηση στον χώρο (FPS controls)

- Αντίληψη κλίμακας
 - interpupillary distance
 - ύψος ματιών



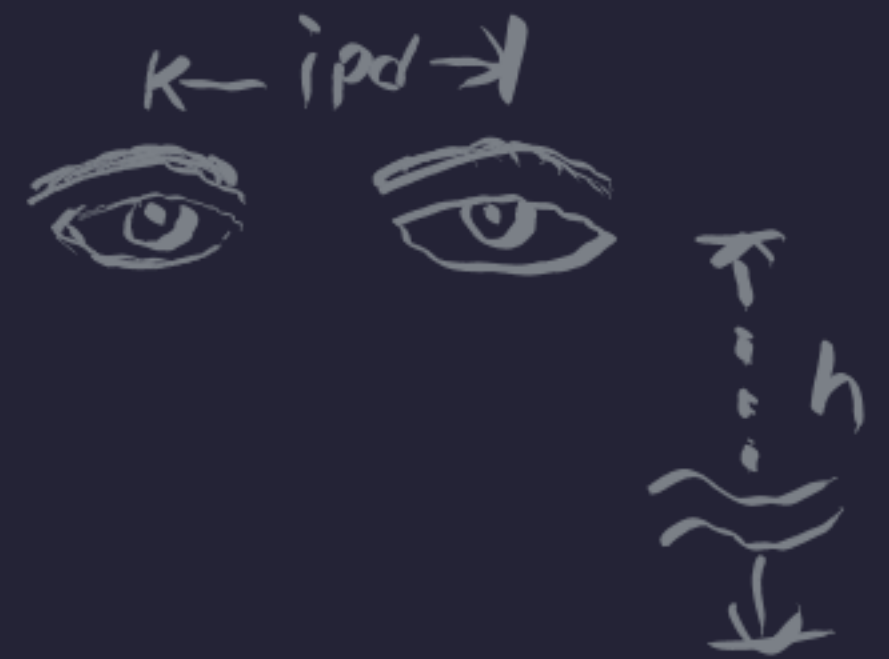
- UI


- crosshairs 
- 2D overlays (e.g. health)
- windows (inventory)

Brand-new problems

- Simulation sickness (ναυτία)
 - latency
 - αυτόματη κίνηση (physics, cinematics)
 - μετακίνηση στον χώρο (FPS controls)

- Αντίληψη κλίμακας
 - interpupillary distance
 - ύψος ματιών



- UI
 - crosshairs 
 - 2D overlays (e.g. health)
 - windows (inventory)

- Input

More stuff

- Michael Abrash's blog @ Valve:
blogs.valvesoftware.com/abrash
- All Abrash & John Carmack talks out there
- Oculus sites:
 - developer.oculus.com
 - share.oculus.com
- Other VR sites:
 - roadtovr.com
 - mtbs3d.com

More ^{of my} stuff

- Simple oculus SDK / OpenGL example
<http://wp.me/p9Evn-5T>
- Proper Oculus DK2 setup on GNU/Linux
<http://wp.me/p9Evn-6v>
- website: nuclear.mutantstargoat.com
- blog: codelab.wordpress.com
- youtube: youtube.com/user/nuclearthelab
- code:
 - github.com/jtsiomb
 - nuclear.mutantstargoat.com/hg