



Game Design and Development for iPhone OS

Part Two

Graeme Devine
iPhone Game Technologies

Introduction

- Okay, I hope you all came to Part One!
- This session is all about how we made Quest

What You'll Learn

- Optimizing for the platforms
- How we lit the dungeon
- Getting to ~ 30 fps or higher
- The artwork of Quest
- The dozen things you need to know





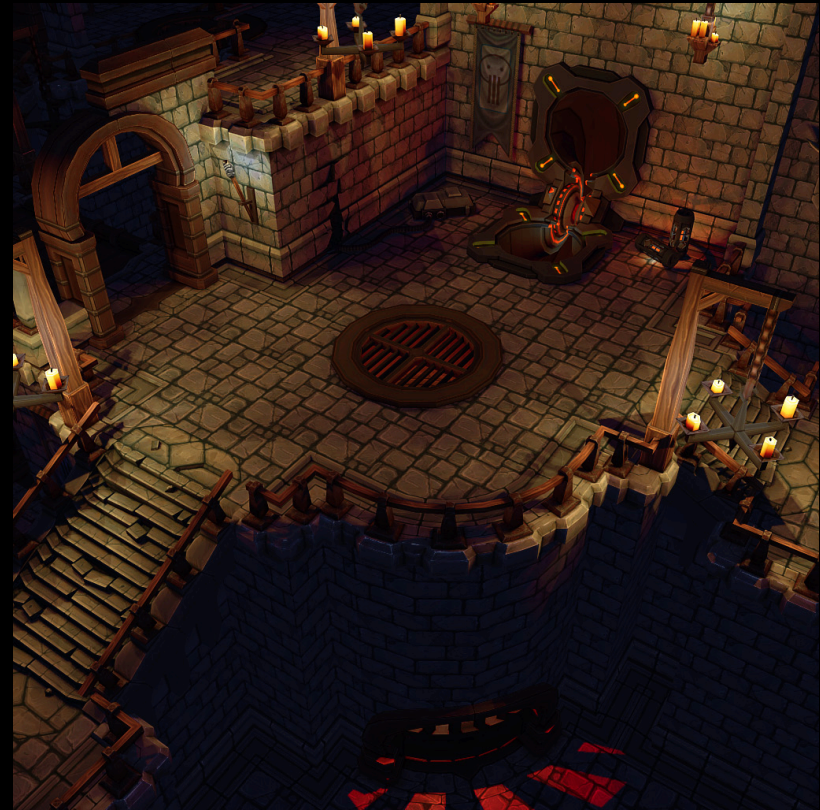
- Two months of work
- Three programmers
- One artist
- iPod Touch
- iPad
- iPhone

Demo

Jeff Ruediger
iPhone Game Technologies

Dungeon Lighting Design

- Diffuse map
- Lightmap, no dynamic lights
- Hardlight in shader ~ 12 fps
- Diffuse with algorithm change
- Precompute! ~ 45 fps





Hardlight in Fragment Shader

Textbook version. SLOW!

```
uniform sampler2D u_diffuseMap;  
uniform sampler2D u_lightMap;  
varying vec2 v_diffuseTexCoord;  
varying vec2 v_lightMapTexCoord;
```

```
void main(void)
```

```
{
```

```
    vec4 dColor = 2.0f * texture2D(u_diffuseMap, v_diffuseTexCoord);
```

```
    vec4 x = texture2D(u_lightMap, v_lightMapTexCoord);
```

```
    vec4 y = dColor - 1.0;
```

```
    vec4 a = x * dColor;
```

```
    vec4 b = x + y - (x * y);
```

```
    vec4 stepValue = step(0.5, x);
```

```
    dColor = mix(a, b, stepValue);
```

```
    gl_FragColor = dColor;
```

```
}
```

Fragment Shader for Dungeon

Diffuse and lightmap adjusted at build time

```
uniform lowp sampler2D u_diffuseMap;
uniform lowp sampler2D u_lightMap;
varying vec2 v_diffuseTexCoord;
varying vec2 v_lightMapTexCoord;

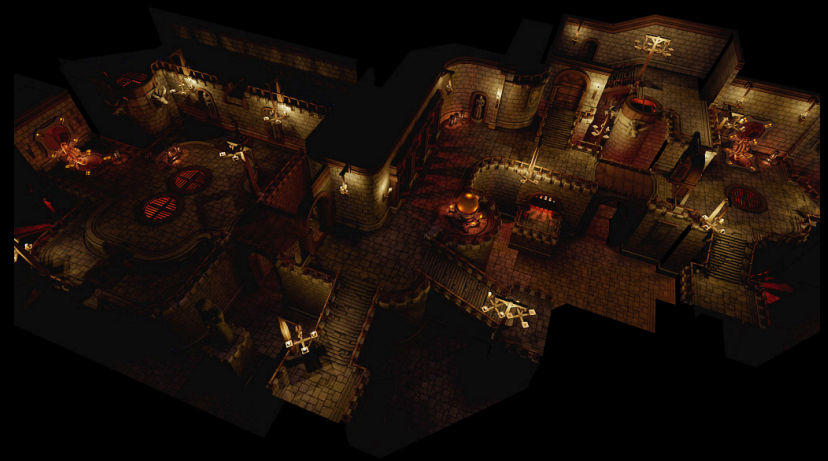
void main(void)
{
    // Sample the diffuse texture
    lowp vec4 dColor = texture2D(u_diffuseMap, v_diffuseTexCoord);

    // Multiply in the lightmap texture
    dColor *= texture2D(u_lightMap, v_lightMapTexCoord);

    gl_FragColor = dColor;
}
```

Dungeon Level Design

- Split into sections
- Stream in as you walk through
- Break up every 65535 index count
- Can drop old sections
- Mesh used to make lightmap



Character Lighting Design

- Use minimap as environment map
- Whoops!
- Fake it!
- One Dynamic rim light



Rim Light Vertex Shader

```
varying vec4 v_rimLight;
```

```
...
```

```
void main(void)
```

```
{
```

```
    // Skinning goes here, see OpenGL skinning presentation.
```

```
    ...
```

```
    lowp vec3 lightN = normalize(camWorldPos - vertexWorldPos);
```

```
    lowp float rimDot = dot(lightN, vertexNormal);
```

```
    lowp float rimValue = smoothstep(0.5, 1.0, rimDot);
```

```
    // Can also pull light color from an attrib.
```

```
    // rimValue * a_rimLightColor.r ...
```

```
    v_rimLight = vec4( rimValue, rimValue, rimValue, 1.0);
```

```
    // Generate Minimap coords based on vertexWorldPos.
```

```
    ...
```

```
}
```

Fragment Shader for Character

```
uniform lowp sampler2D u_diffuseMap;  
uniform lowp sampler2D u_lightMap;  
varying vec2 v_diffuseTexCoord;  
varying vec2 v_lightMapTexCoord;  
varying vec4 v_rimLight;
```

```
void main(void)  
{
```

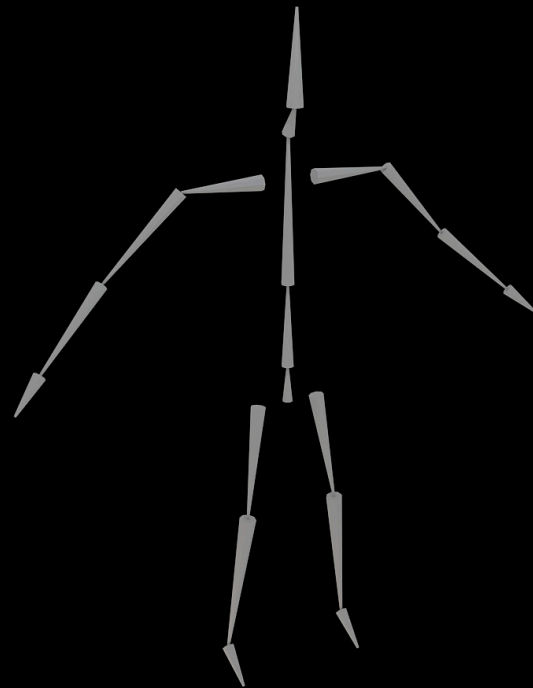
```
    // Sample the diffuse texture and multiply in the minimap  
    lowp vec4 dColor = texture2D(u_diffuseMap, v_diffuseTexCoord);  
    dColor *= texture2D(u_lightMap, v_lightMapTexCoord);
```

```
    // Add Rim Light  
    dColor += v_rimLight;  
    gl_FragColor = dColor;
```

```
}
```

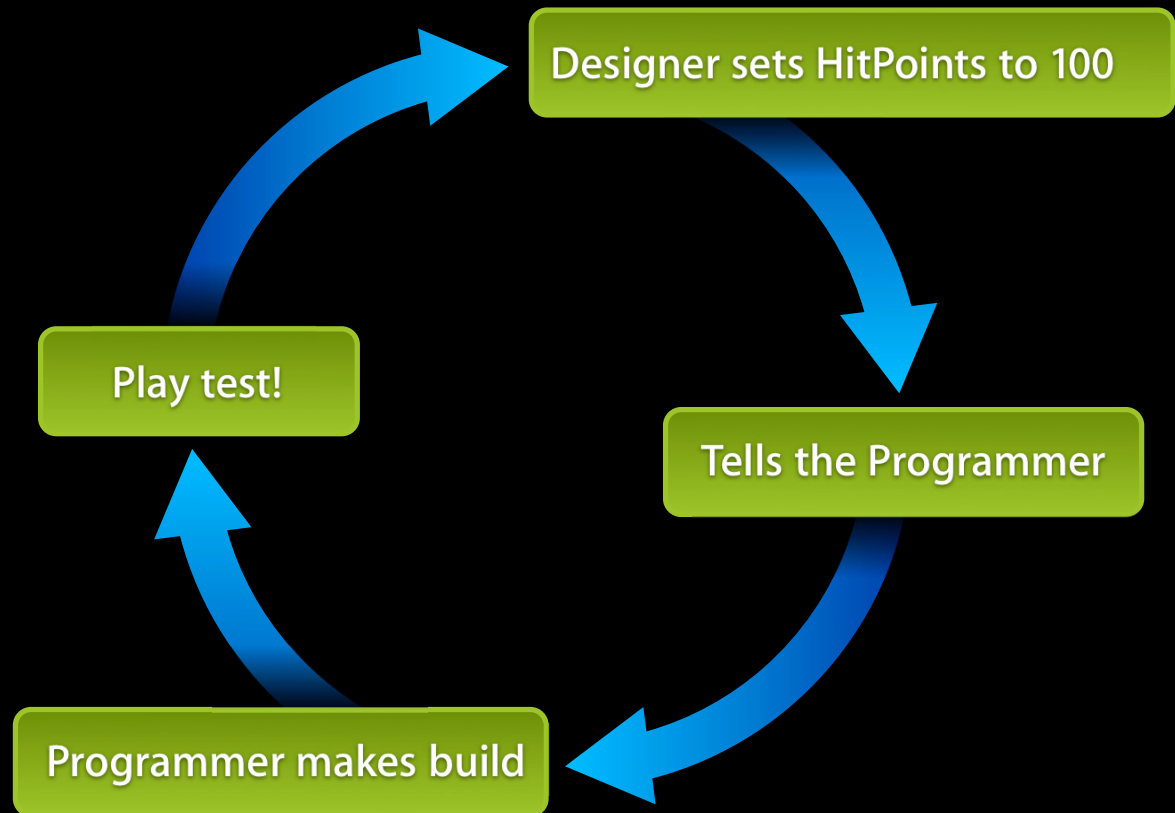

Character Animation

- Skinning vs. mesh animation
- CPU vs. GPU
- How many weights per vertex?
- One solution does not fit all



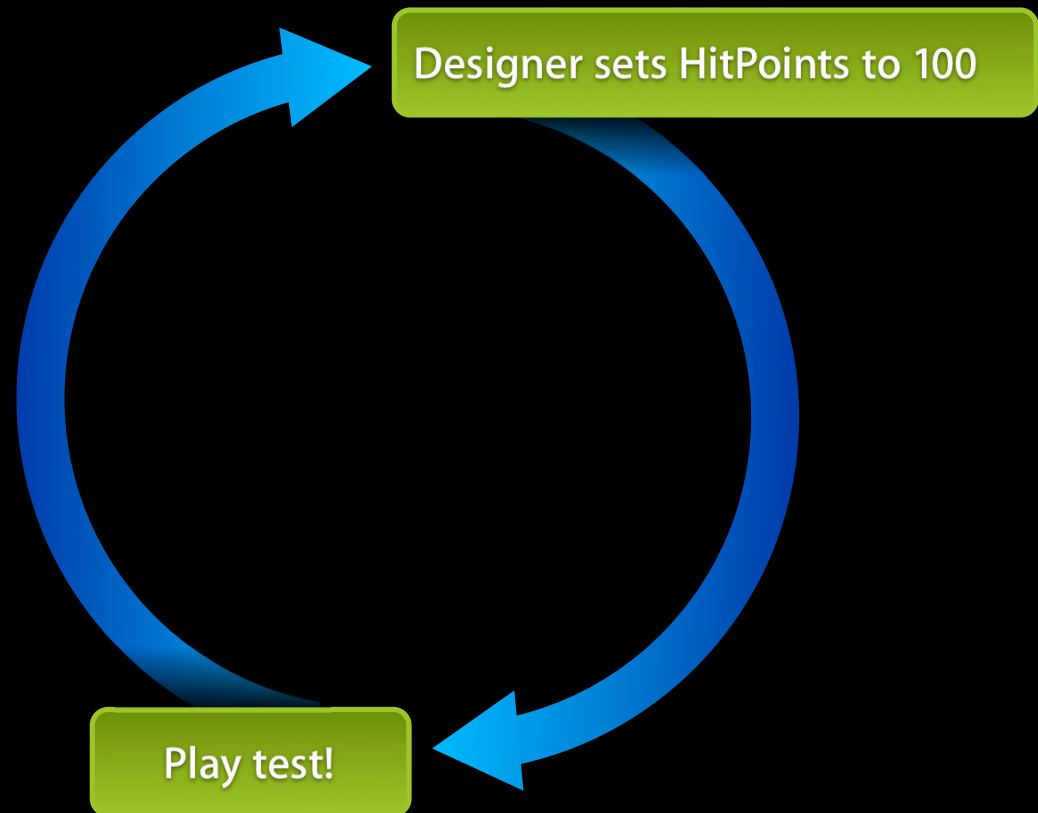
Game Design Iteration

- Typical design loop



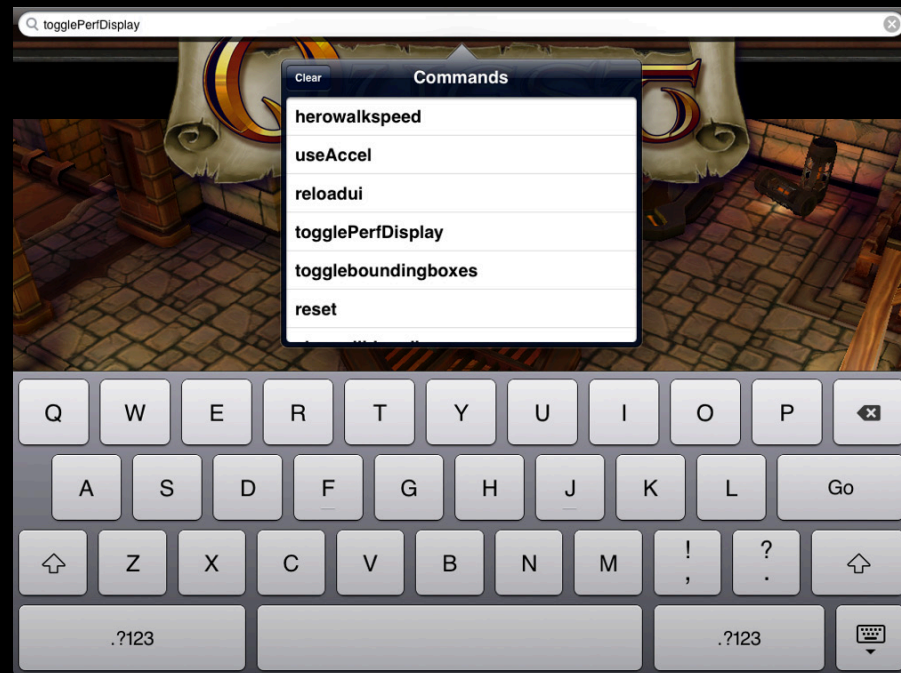
Game Design Iteration

- Typical design loop
- Fast iteration loop



Game Design Iteration

- Typical design loop
- Fast iteration loop
- Data drive your engine



GameTools Console

Adding a config variable

```
// At application start. Register once.
```

```
- (void)appInit {
```

```
    NSNumber* hp = [NSNumber numberWithFloat: 100.0f];  
    [self addConfigVariable: @"heroHitPoints"  
        withValue: hp  
        addCallback: @"heroHitPointsChanged:"];
```

```
}
```

```
// Callback
```

```
- (void)heroHitPointsChanged:(NSDictionary*)change {
```

```
    NSNumber* hp = [change objectForKey:@"new"];  
    NSLog(@"Hitpoints changed to %.2f", [hp floatValue]);
```

```
}
```

GameTools Console

Adding a console command

```
// At application start. Register once.  
- (void) appInit {  
    [self addConsoleCommand: @"toggleBBox"  
        withCallback: @"toggleBBox"];  
}  
  
- (void) toggleBBox {  
    // Turn on/off Bounding box rendering.  
}
```

GameTools Console

Turning features on/off at runtime

```
// At application start. Register once.
- (void) appInit {
    NSNumber* woot = [NSNumber numberWithBool: YES];
    [self addConfigVariable: @"JeffsCrazyFeature"
        withValue: woot];
}

- (void) gameUpdate {
    if( [self isConfigDefined:@"JeffsCrazyFeature"] )
    {
        // Best Feature ever goes here.
    }
}
```

Materials available at:

<http://developer.apple.com/wwdc/>

Game Artwork Generation

Pete Parisi

iPhone Game Technologies

Art for Quest

- Art style
- Texture atlas
- The grid
- Reusable library
- Art efficiency



Style

- Choosing a style
- Cohesive
- Achievable



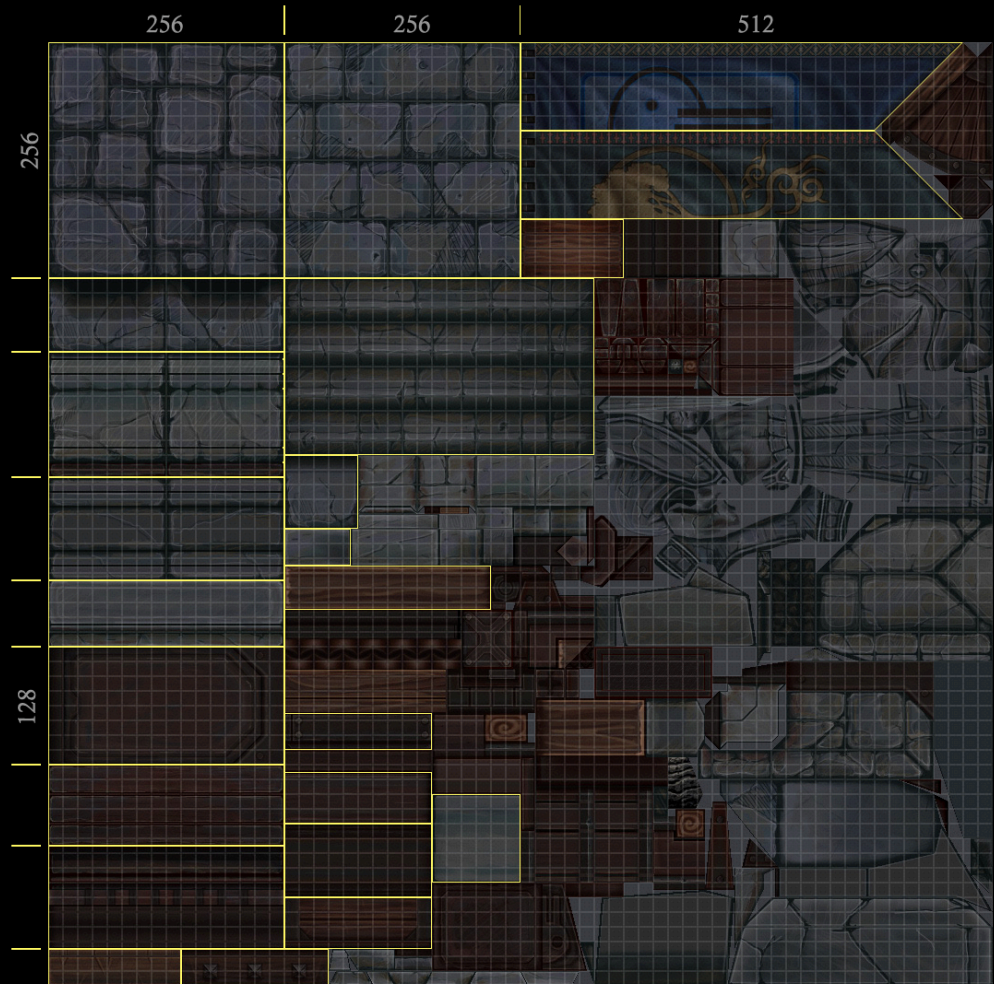
Texture Atlas

- Entire dungeon one 1024x1024
- Shared on many things
- Artistically unified



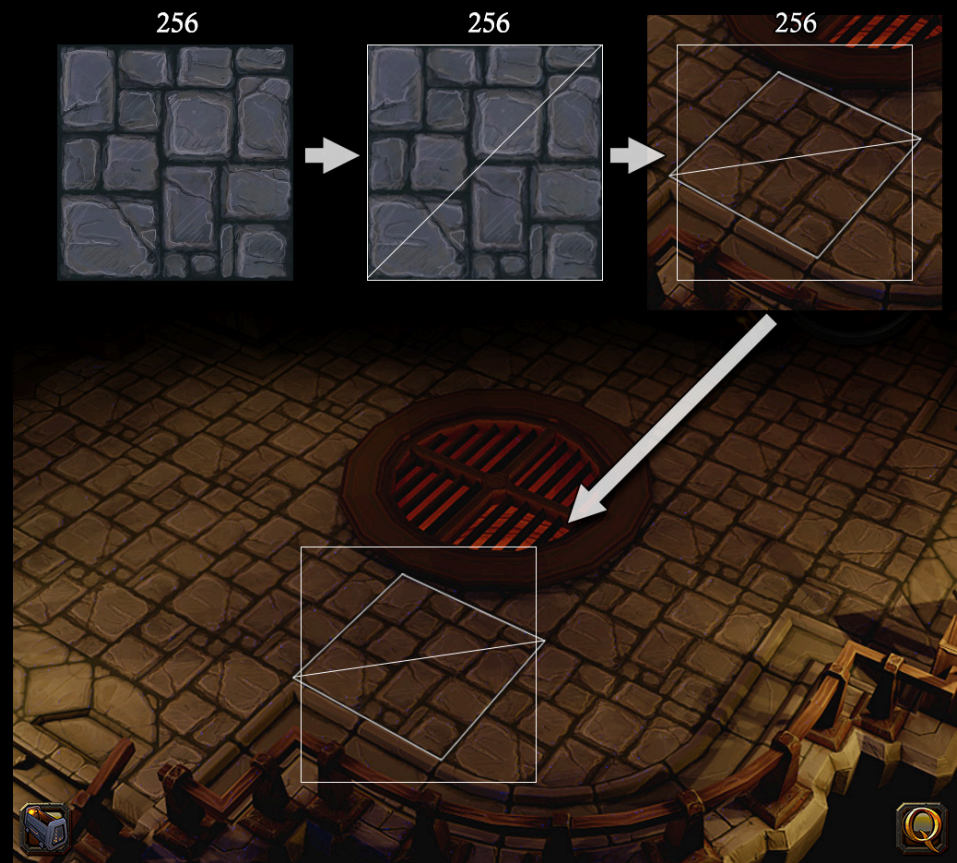
Texture Atlas

On the grid



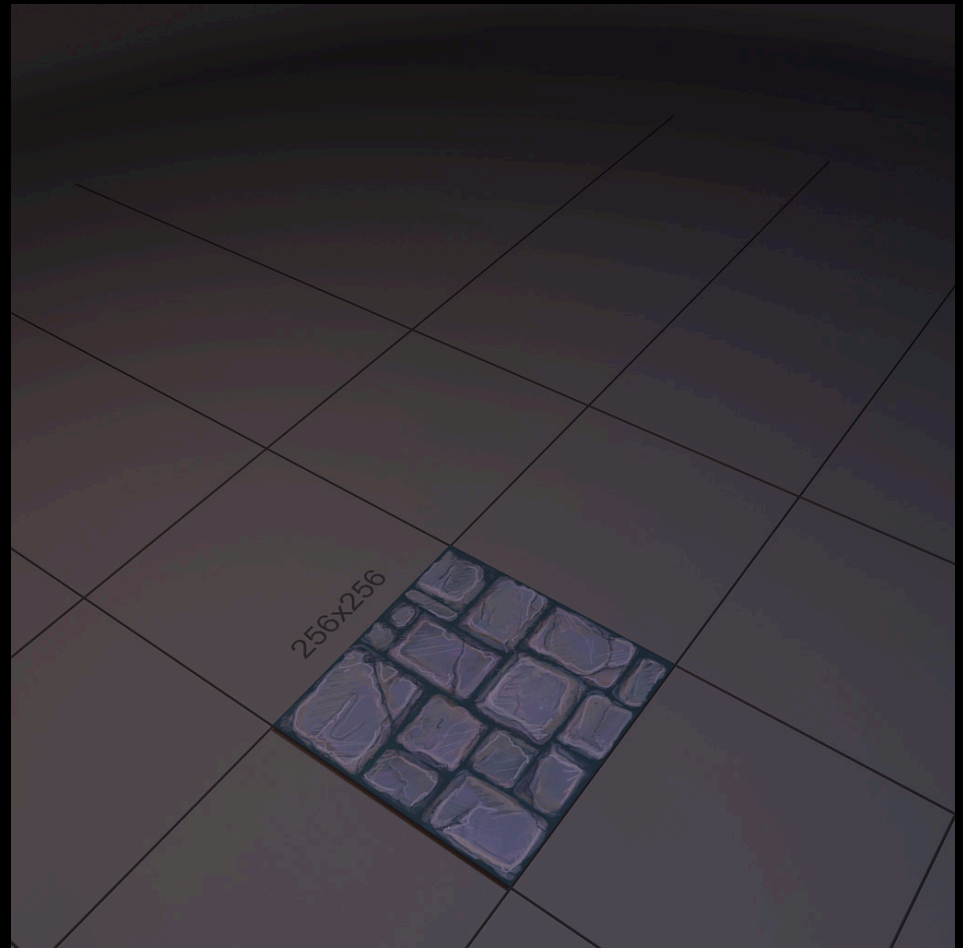
The Grid

- Power of 2 grid
- Used on all art assets
- Helps to keep texel density unified



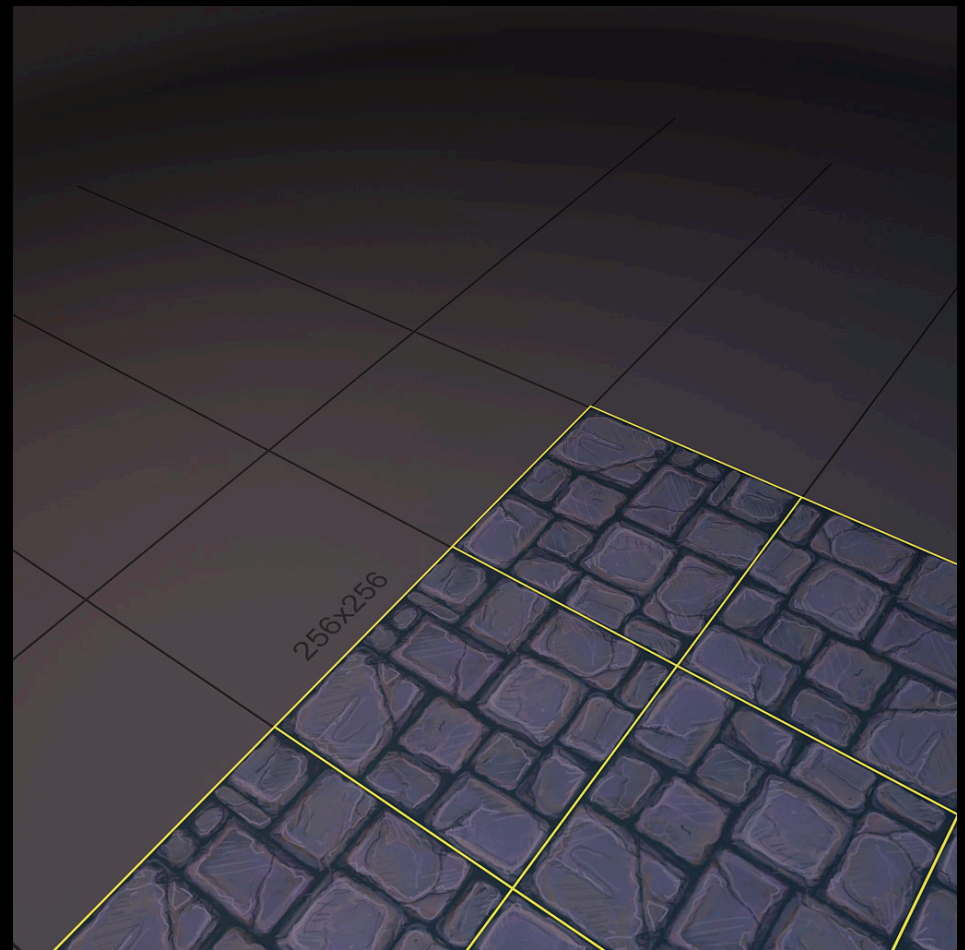
The Grid

- Putting it all together
- Simple floor geometry on the grid 256x256



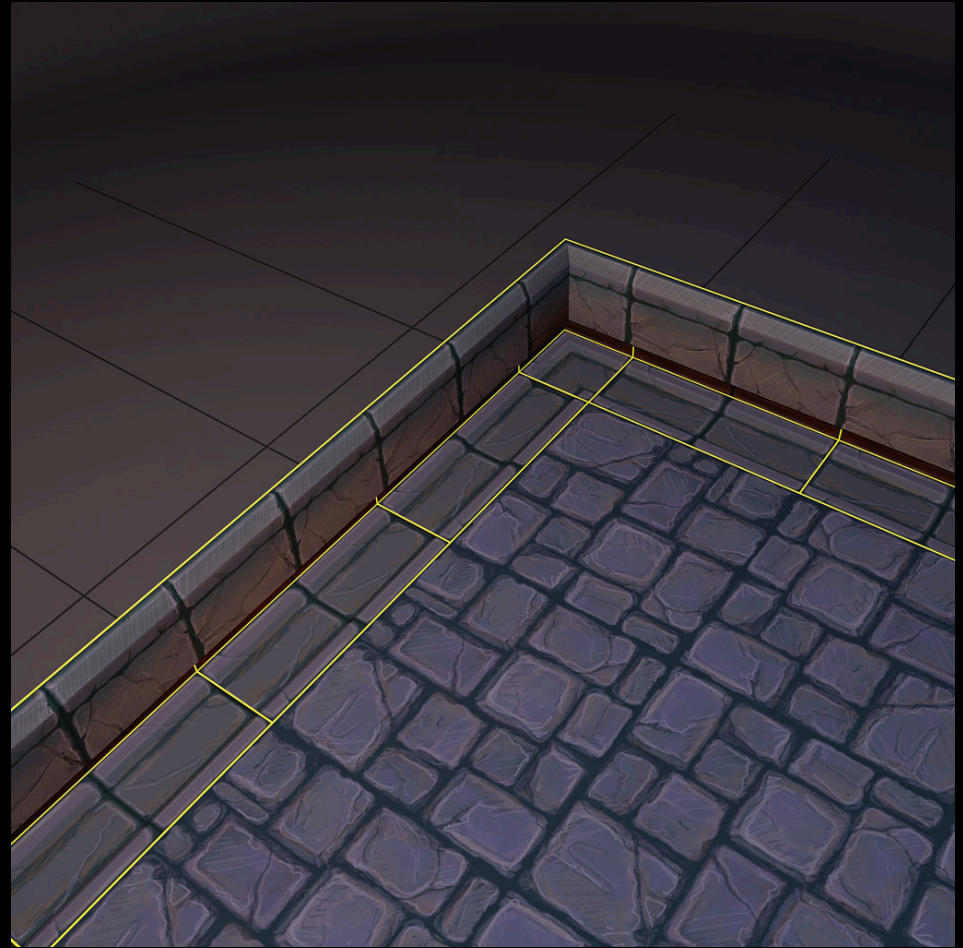
The Grid

- Tiled floor
- Rotated and mirrored



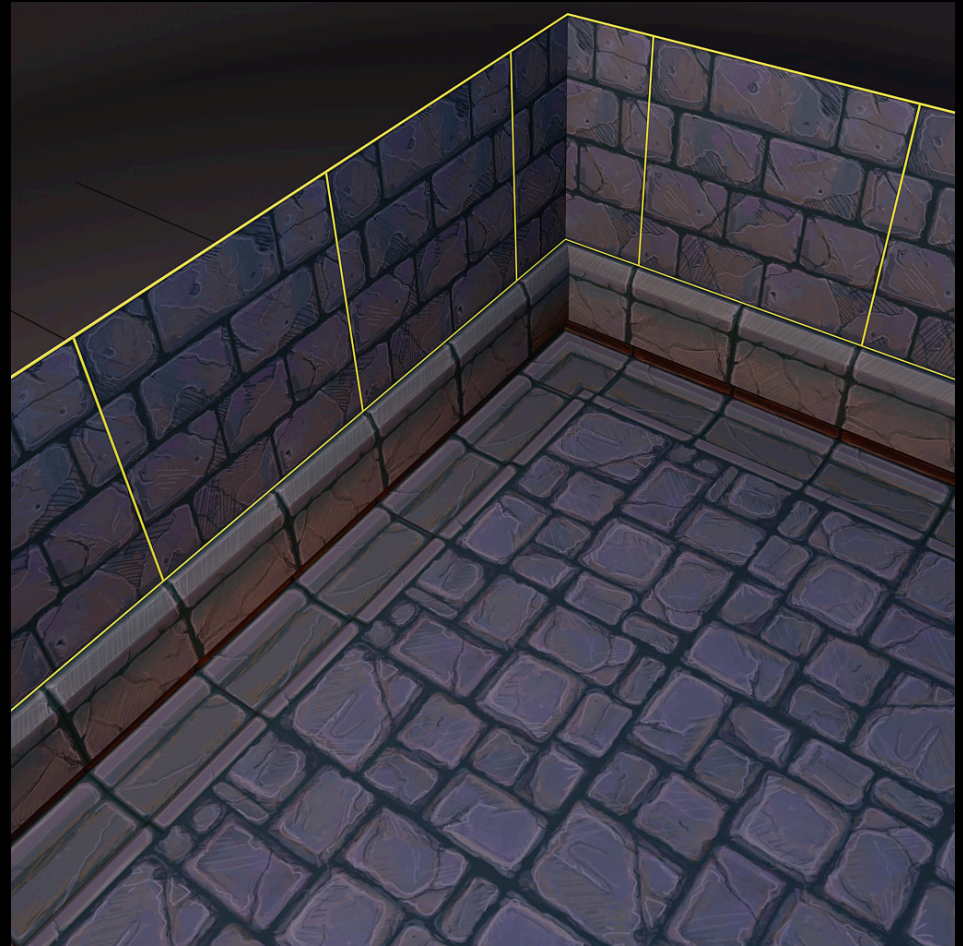
The Grid

- Baseboard trim, corner, and edge pieces



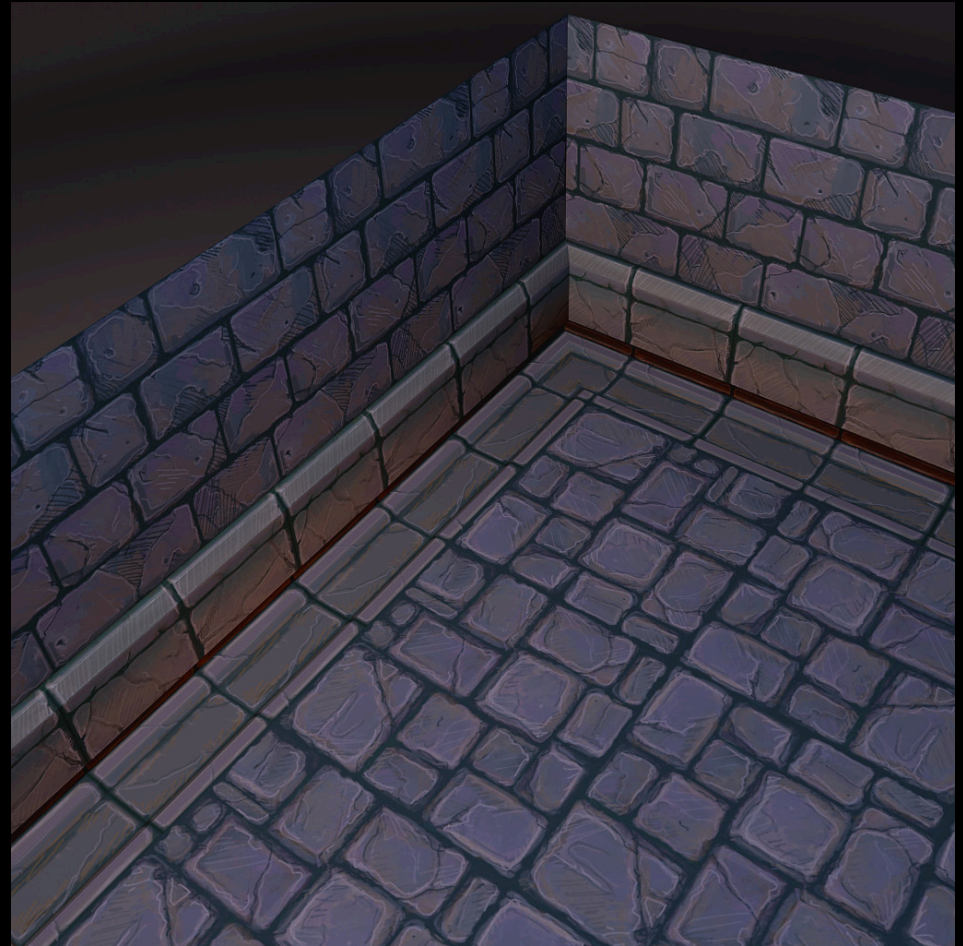
The Grid

- Tiled walls



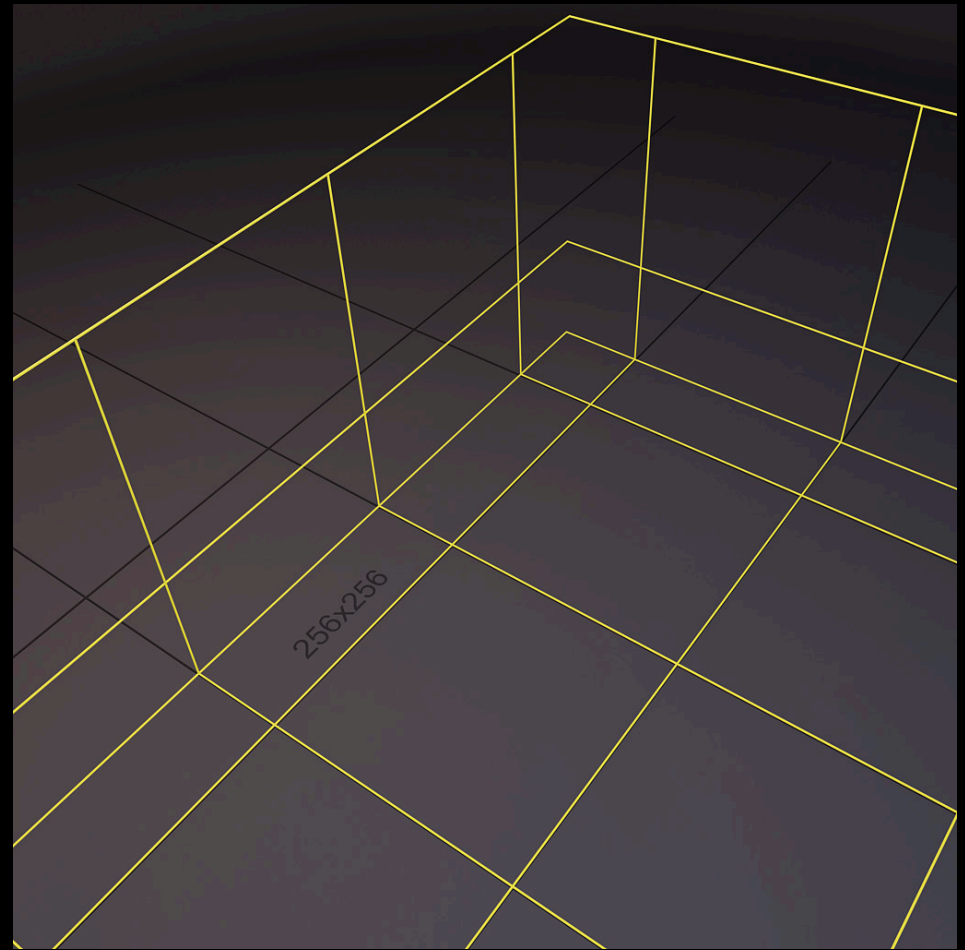
The Grid

- Everything together



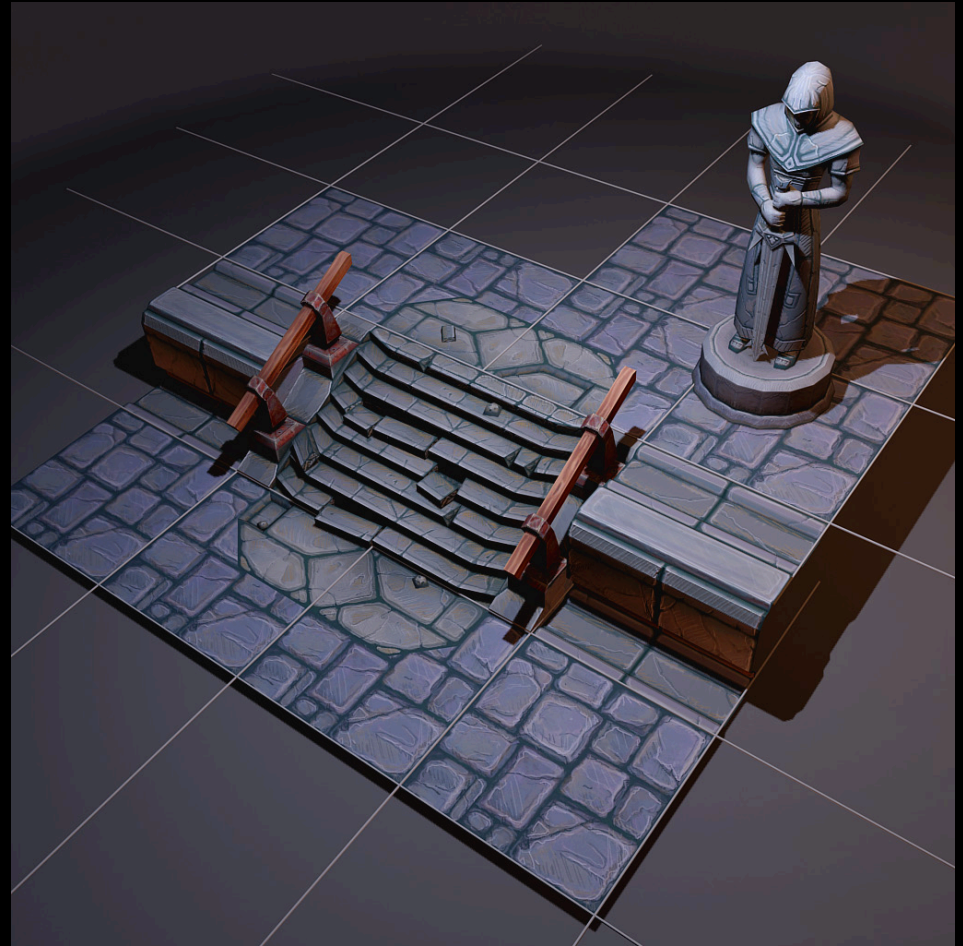
The Grid

- The pieces



The Grid

- Just a suggestion



Reusable Library

- Puzzle pieces



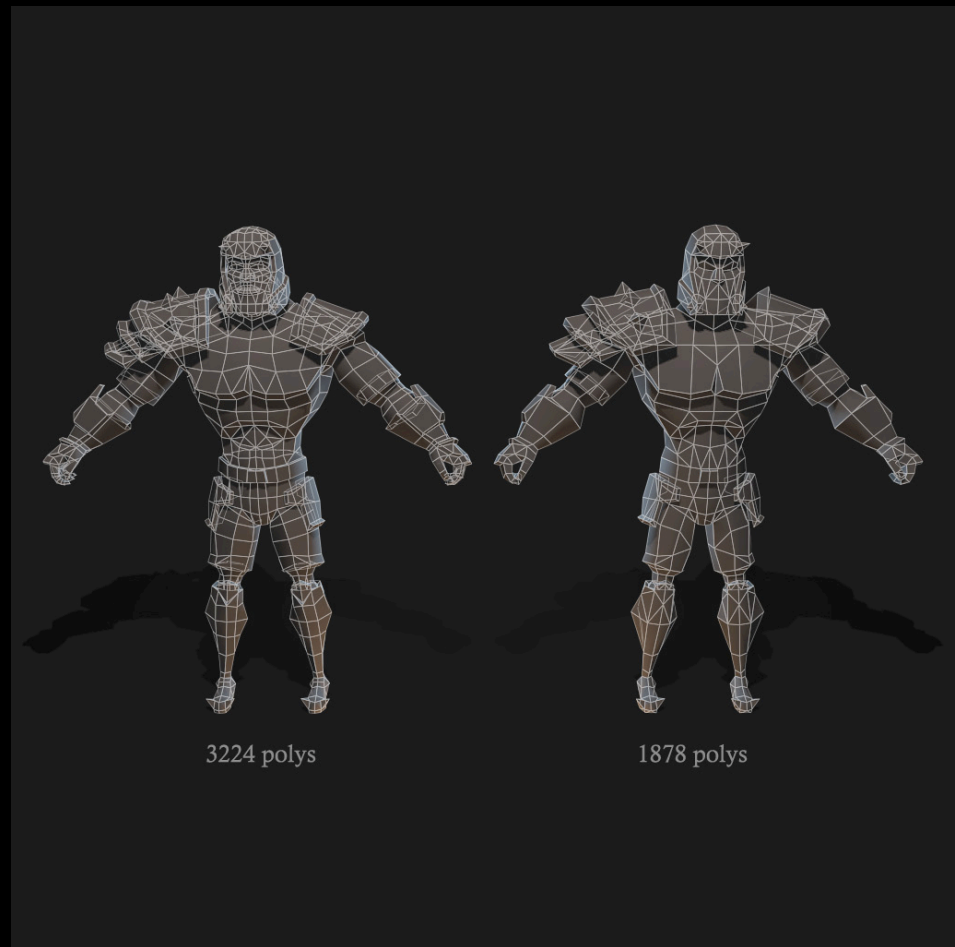
Art Efficiency

- Minimum amount of resources for the maximum amount of quality



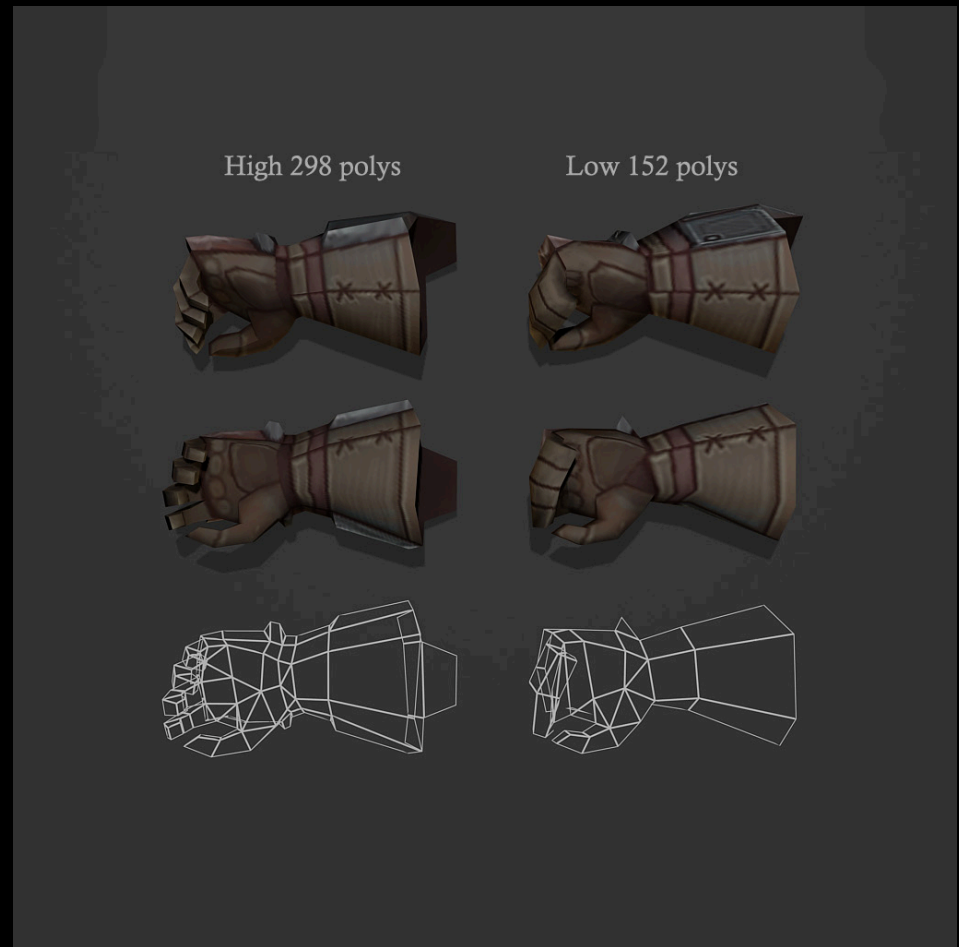
Art Efficiency

- Reduced polycount
- No visual difference



Art Efficiency

- Simplify



Art Efficiency

- Next gen tradeoffs



In Closing

- This is nothing new
- Do what works for you

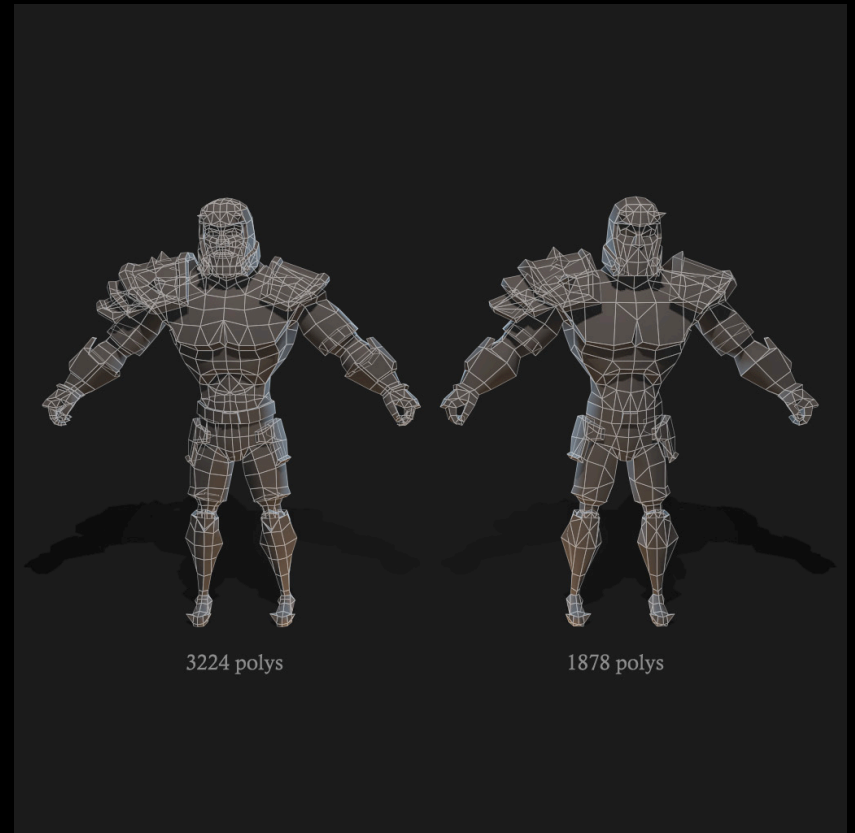


The Dozen Things We Learnt

Graeme Devine
iPhone Game Technologies

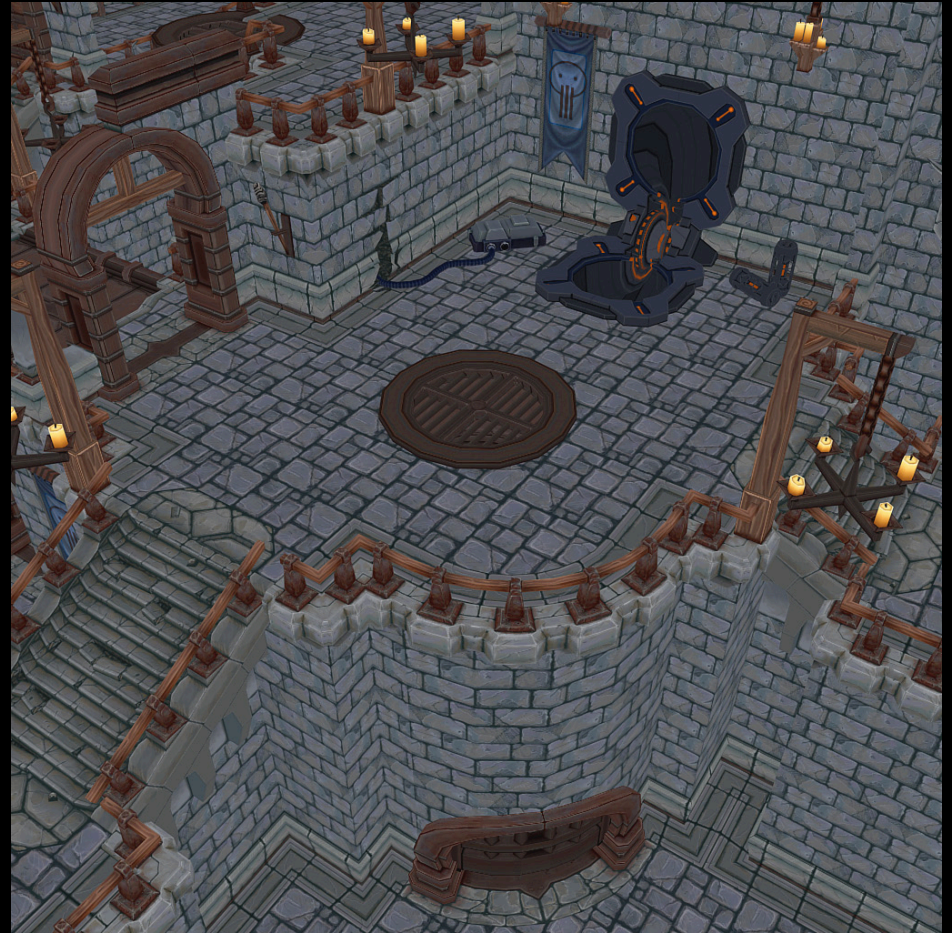
1 Optimization

Assets



1 Optimization

Shaders



1 Optimization

It's a game



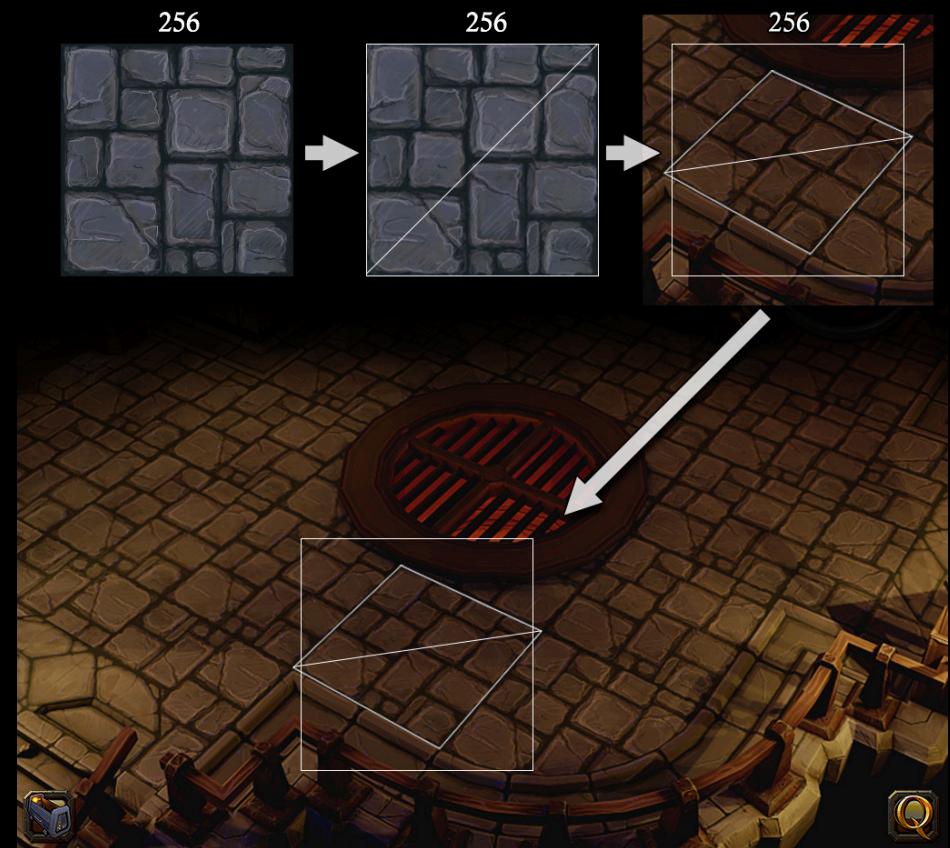
2 Memory

Purge assets not in use



2 Memory

Texture size



2 Memory

PVRTC



Uncompressed
32 bpp

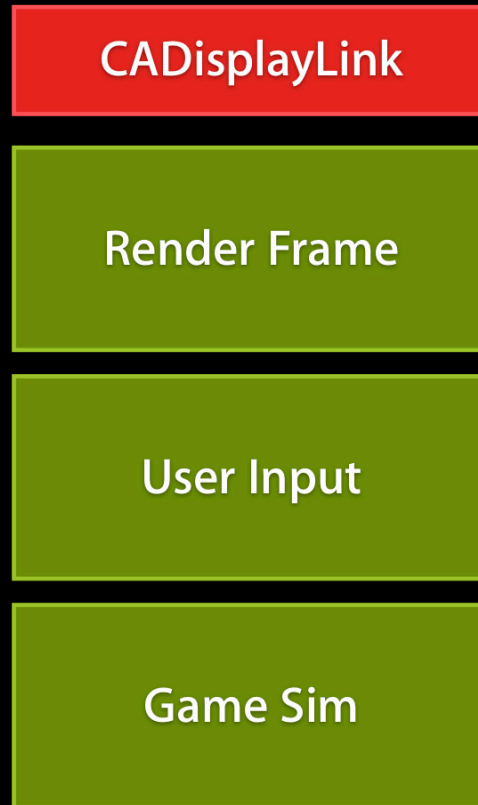


Compressed
4 bpp



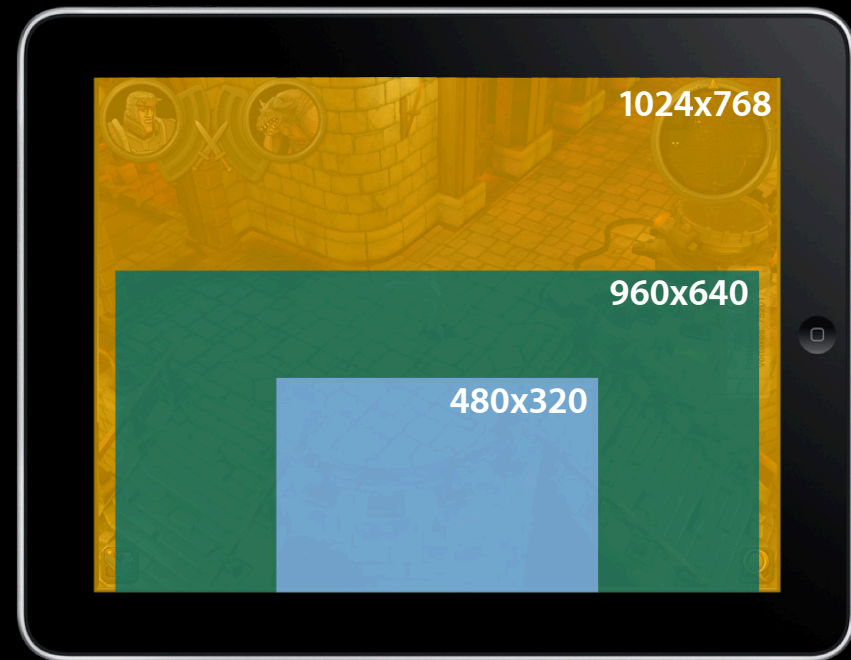
Compressed
2 bpp

3 Gameloops



4 Multiple Devices

- Screen size
- Resolution
- Capabilities



5 Playtesting

- Watch people
- Play it yourself
- Do this all the time



6 This Is Not Reality

- Real physics!
- Real lighting!
- Real shadows!



7 Rendering the World

- Single texture bind
- You should be thinking less than 10 draw calls per frame



8 Things We Did Right

- Data driven pipeline rocks
- Collada as an export process
- Scene graph based 3D system
- Cheated in the right places
- Using Apple technologies



9 Things We Did Wrong

- Tools
- Asset pipeline still not perfect
- Scoped a teensy bit on the impossible side
- Could we make this into a real game?



10 Books and Magazines You Should Read

- Game Developer Magazine
- Graphic Gems series
- Game Programming Gems series



11 Sessions You Must Go To

Introduction to Game Center	Pacific Heights Tuesday 2:00PM
Game Center Techniques, Part One and Part Two	Pacific Heights Tuesday 3:15PM
OpenGL ES Overview for iPhone OS	Presidio Wednesday 2:00PM
OpenGL ES Shading and Advanced Rendering	Presidio Wednesday 3:15PM
OpenGL ES Tuning and Optimization	Presidio Wednesday 4:30PM
Core Animation in Practice, Part One	Nob Hill Thursday 11:30AM
Core Animation in Practice, Part Two	Nob Hill Thursday 2:00PM
Discovering AV Foundation	Nob Hill Thursday 4:30PM

This Is the Most Important
Thing I Can Tell You

12 Have Fun

Labs

Game Design for iPhone OS Lab

Graphics & Media Lab A
Wednesday 2:00PM

Game Design for iPhone OS Lab

Graphics & Media Lab A
Friday 11:30AM

More Information

Allan Schaffer

Graphics and Game Technologies Evangelist
aschaffer@apple.com

Documentation

OpenGL Dev Center
<http://developer.apple.com/opengl>

Apple Developer Forums

<http://devforums.apple.com>



