Pie - Ricoh

guisabran
gsabran
guillaume.sabran

Me :)

- Guillaume Sabran
- Co-founder & CTO @Pie
- Math - software
- French native

guisabran gsabran guillaume.sabran

What is Pie?

The 360 video platform for UGC content

- video
- mobile (get the iOS app!)
- daily usage
- large public

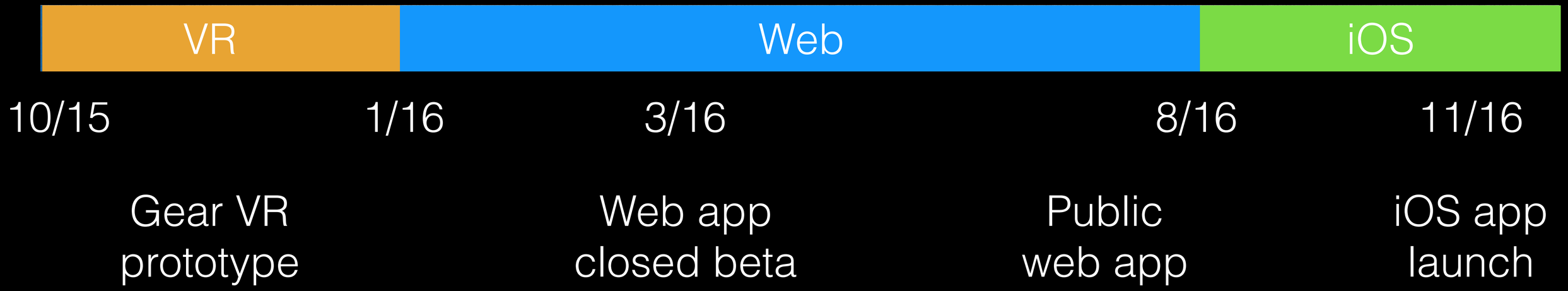
What is Pie?

demo

Pie - Ricoh

- From VR to desktop to mobile
- Insta style image / video filter

VR to mobile

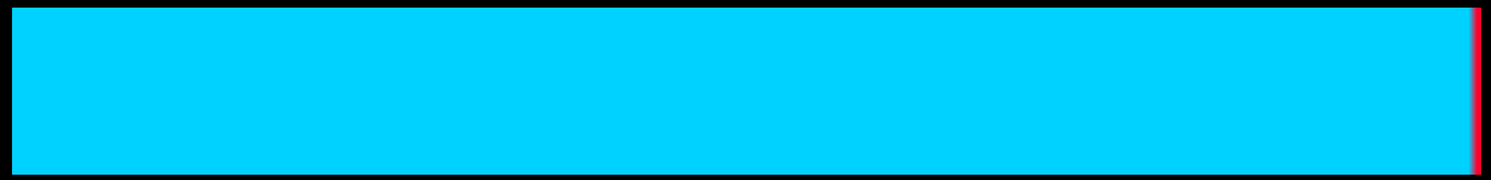


guisabran gsabran guillaume.sabran

VR to mobile

1) Why we gave up on VR (for now)

- 360 & VR are orthogonal
- VR market is small:
  - end 2016: ~12M devices sold, vs 2.1B mobile users



- We're a startup

VR to mobile

2) How we got to mobile

- followed trends in UGC cams
- better for UGC creation

VR	Web	iOS

10/15 1/16 3/16 8/16 11/16

guisabran gsabran guillaume.sabran

Filter

From

To

guisabran gsabran guillaume.sabran

Filter

From
- original image
- position in display

To
- color for pixel A

Filter

From
- original image
- position in display

To
- color for pixel A

Shader

Filter

From
- original image
- position in display

To
- color for pixel A

Shader

- color mapping:
  - B&W
  - sepia
- neighbor pixel interaction:
  - blur
  - edge sharpening
- frame effect:
  - gradient

guisabran gsabran guillaume.sabran

Filter

From

To

guisabran gsabran guillaume.sabran

Filter

- color mapping:
  - B&W: - color = vec3((R + G + B) / 3)
  - color map: - R = sample(colorMap, vec2(R, 0))

- neighbor pixel interaction:
  - blur - color = 4 * color - top - bot - left - right
  - edge sharpening - color = top - color

- frame effect:
  - gradient - color = color * sample(gradient, position)

guisabran gsabran guillaume.sabran

Filter

iOS - DDDKit

- open source: https://github.com/team-pie/DDDKit
- Swift (OpenGL backed) library for 3D rendering, with focus on video support.

```
// creates an element
let node = DDDNode()
// give it a shape
node.geometry = DDDGeometry.Sphere(radius: 20.0, orientation: .inward)
// pass the video input
let videoTexture = DDDVideoTexture(player: player)
videoNode.material.set(property: videoTexture, for: "SamplerY", and: "SamplerUV")
// add the node to the scene
scene.add(node: node)
// move it
node.position = Vec3(v: (0, 0, -30))
```

Filter

iOS - DDDKit

- open source: https://github.com/team-pie/DDDKit
- Swift (OpenGL backed) library for 3D rendering, with focus on video support.
- Simple support of filters:

```
let program = try! DDDShaderProgram(fragment: fShader, shaderModifiers: [
    .fragment: "gl_FragColor = vec4((col.x + col.y + col.z) / 3.0, 1.0);",
])
videoNode.material.shaderProgram = program
```

360 video filter

From
- original image
- position in display

To
- color for pixel A

360 video filter

From
- original image
- position in display
- time
- position in 360

To
- color for pixel A

360 video filter

Demo

Questions?
guisabran
gsabran
guillaume.sabran