Implementation of VRM on Three.js

W3C Developer Meetup 2019

pixiv Inc.
OBUCHI Yutaka

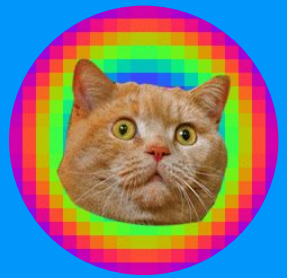
pixiv

2018.6.5

OBUCHI Yutaka

Engineer at pixiv Inc.
Specializing in CG

Twitter: @FMS_Cat



pixiv

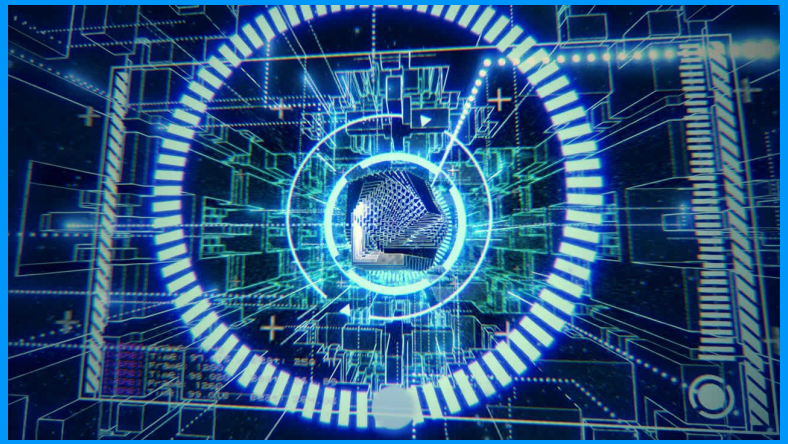
OBUCHI Yutaka

Working on 3D character platform called VRoid Hub

pixiv

OBUCHI Yutaka

I’m also a hobbyist demoscener!
Do 3D graphics experience on Web



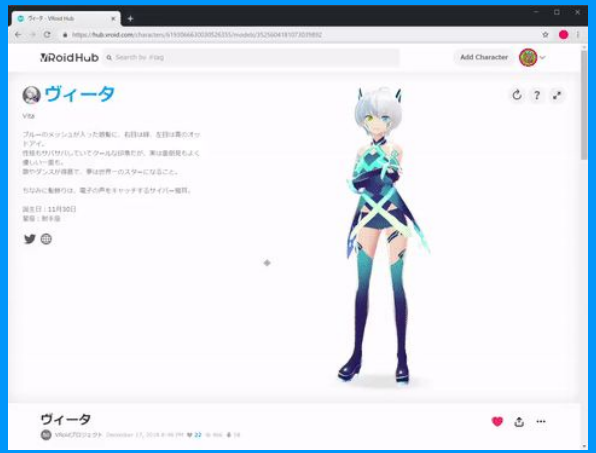
pixiv

Today I’m gonna talk about...

pixiv

Today I’m gonna talk about...

Implementation of 3D character viewer on Web browser



pixiv

Today I’m gonna talk about...

Especially, I’m gonna focus on:

- **Overview of VRM, A format for 3D avatar**
- **VRM Implementation on Web, especially materials**
- **Performance / Security issues we encountered**

pixiv

VRoid Hub?

pixiv

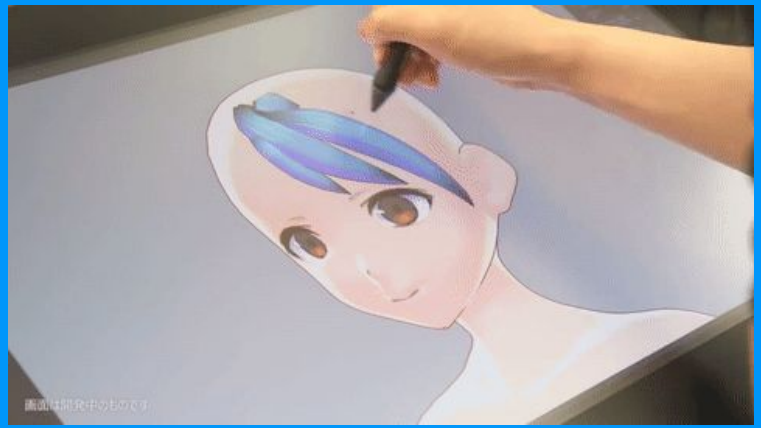
VRoid Hub?

First let me talk about
our Web service VRoid Hub and its motivation

pixiv

VRoid Hub?

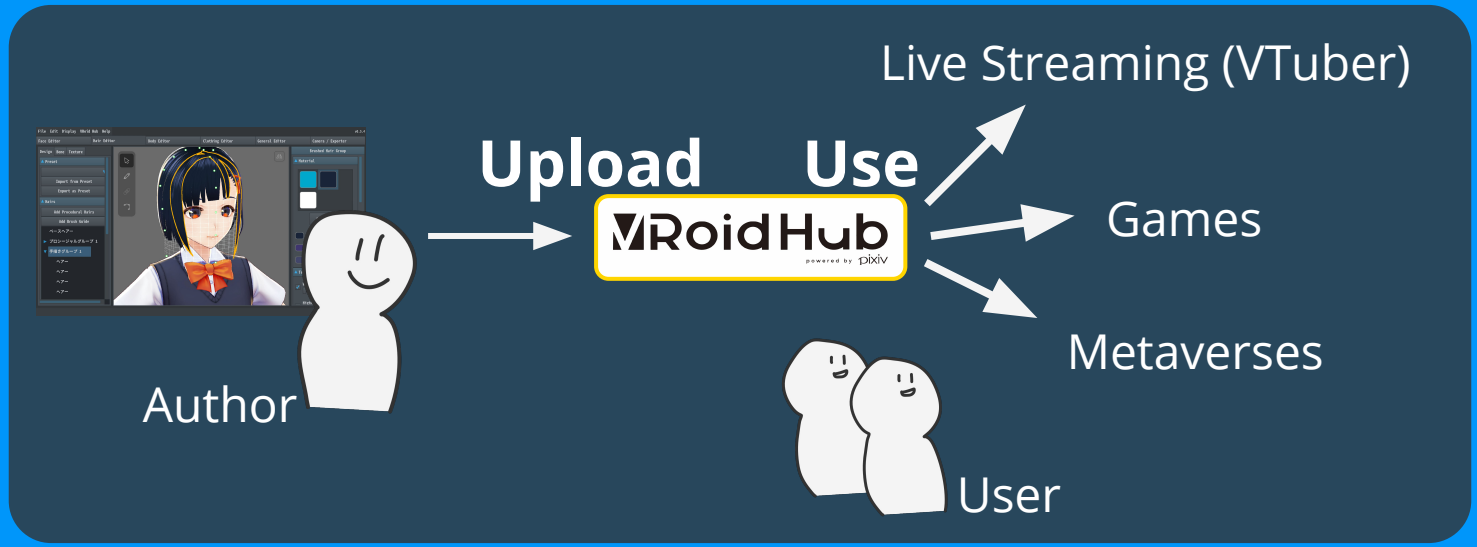
Nowadays it becomes easier
to make your own 3D character



pixiv

VRoid Hub?

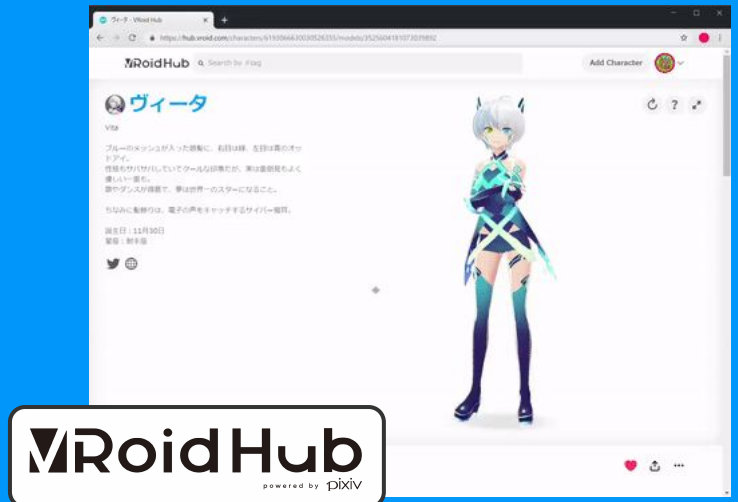
A “Hub” for 3D characters



pixiv

VRoid Hub?

We must present artists’ character on Web browser utilizing full potential of its expression

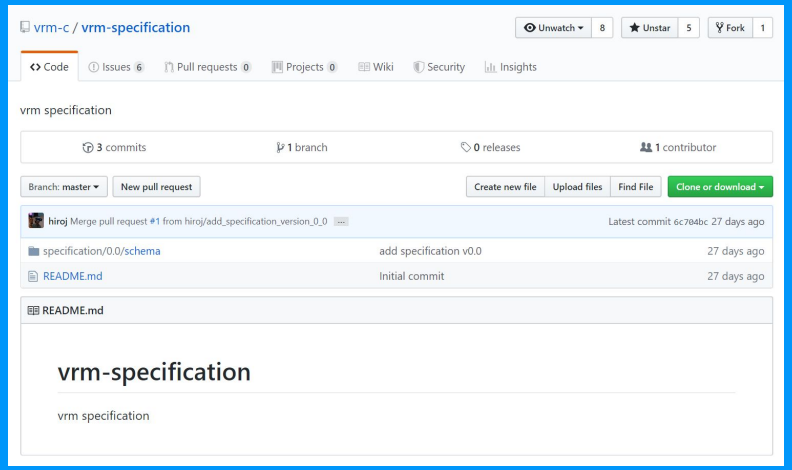


VRoid Hub powered by pixiv

pixiv

VRM: A format for 3D avatar

We use a 3D model format called **VRM**
that **specializes in 3D character expression**

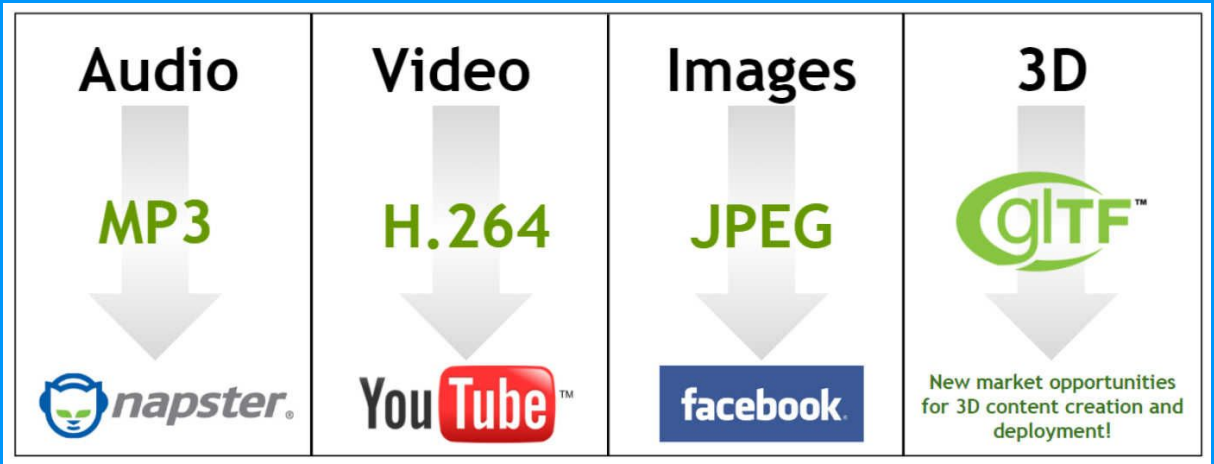


https://github.com/vrm-c/vrm-specification

pixiv

VRM: A format for 3D avatar

VRM is implemented as
a domain-specific **extension of glTF**



https://www.khronos.org/gltf/

pixiv

VRM: A format for 3D avatar

VRM has various features to deal with
Humanoid avatars

Humanoid
interfaces

Facial expressions

Toon materials

Features for
First person view

Dynamic bones

Metafield
e.g. licenses

pixiv

VRM: A format for 3D avatar

Humanoid
interfaces

basically, map of `{bone key: node}`

pixiv

VRM: A format for 3D avatar

Most of these specs are **meant to be used by modern game engines**

e.g. certain Humanoid specs completely relies on implementation of Humanoid component in game engines

→ We, Web developers, had to **adopt these game engine oriented spec** into how we deal with 3D stuff on Web

pixiv

VRM: A format for 3D avatar

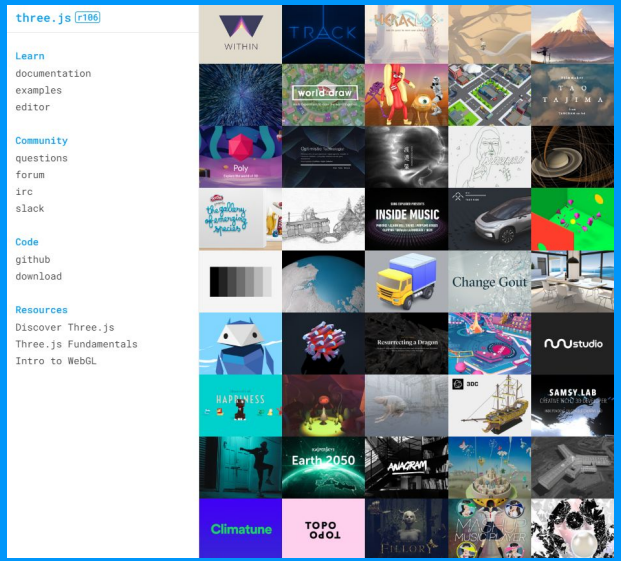
VRM Consortium:
An org to define the spec of VRM

https://vrm-consortium.org/

pixiv

Implementation

Implementation



https://threejs.org/

We implemented VRM importer as a module of Three.js

Three.js already has a glTF implementation!

pixiv

Implementation

The implementation is available under MIT License

https://github.com/pixiv/three-vrm

pixiv

Implementation

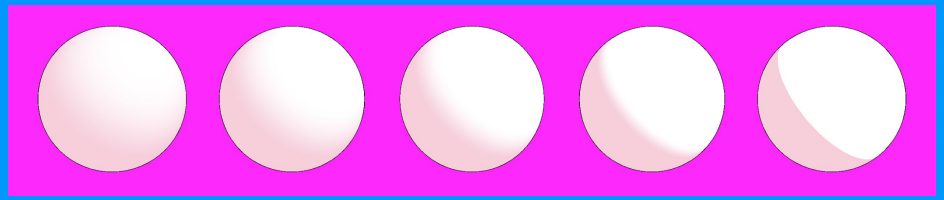
I’m gonna talk especially about

Materials

Implementation: Material

Toon material spec used in VRM: **MToon**

- **Minimal, efficient lightings** that can be adopted to various situation
  - From cell animation like to character art like
- Pseudo specular expression using **matcap textures**
- Line drawing like expression using **outlines**



pixiv

Implementation: Material

Unlit (no lightings)

MToon

pixiv

Implementation: Material

In 3DCG realm, we do program
how 3D objects should be rendered using **shaders**

A shader = **An implementation of a material**

Implementation: Material

How shader programs looks like:

```
uniform vec3 color;

void main(void)
{
    gl_FragColor = vec4(color, 1.0);
}
```

“Simply output the input color”

pixiv

Implementation: Material

```
varying vec2 uv;
uniform vec3 color;
uniform sampler2D map;

void main(void)
{
  vec3 tex = texture2D(map, uv);
  gl_FragColor = vec4(tex * color, 1.0);
}
```

plus “Do a texture mapping”

pixiv

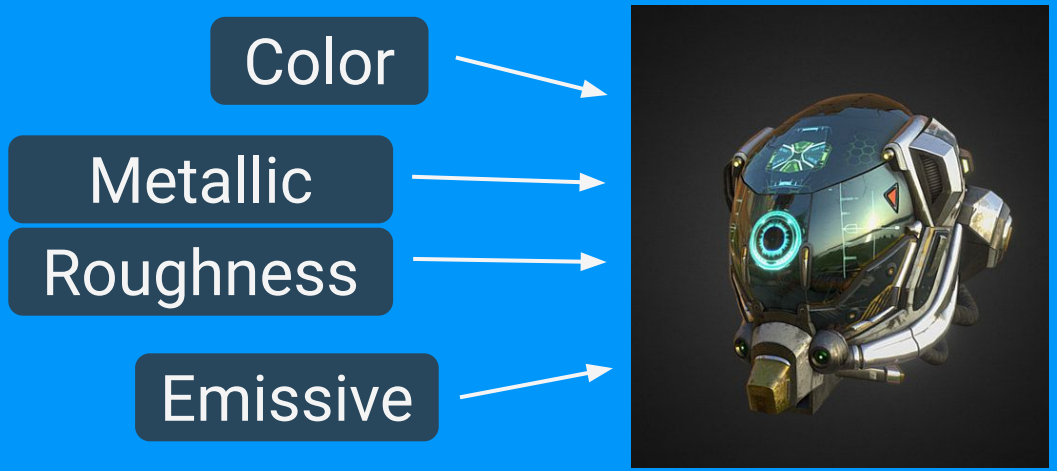
Implementation: Material

Since shaders are described in a program
it lacks portability

Well-described material specification is required
to use certain material in arbitrary context

Implementation: Material

To resolve this problem,
glTF uses an interface of **PBR material**



https://sketchfab.com/3d-models/battle-damaged-sci-fi-helmet-pbr-b81008d513954189a063ff901f7abfe4

pixiv

Implementation: Material

The specification of MToon is
completely aside from ordinary PBR materials

We had to convert Unity shader implementation
into GLSL that Three.js uses

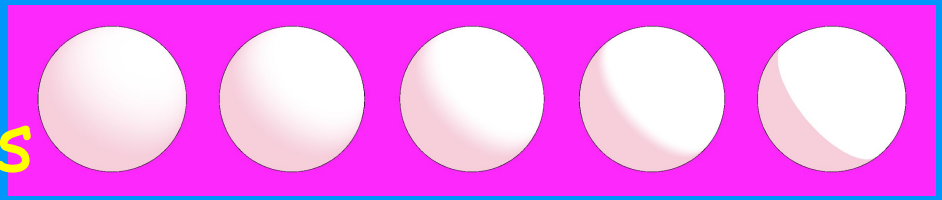
pixiv

Implementation: Material

Toon material spec used in VRM: **MToon**

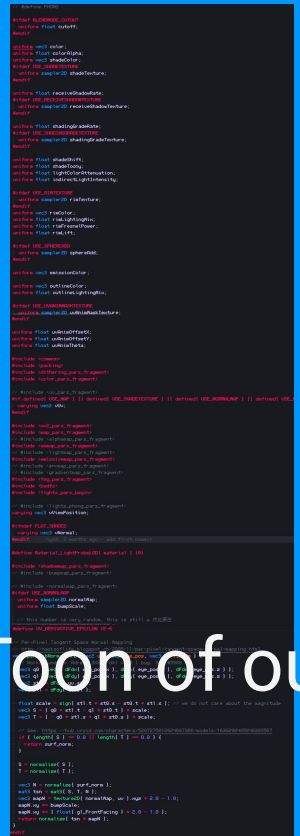
- **Minimal, efficient lightings** that can be adopted to various situation
  - From cell animation like to character art like
- Pseudo specular expression using **matcap textures**
- Line drawing like expression using **outlines**

so many features



pixiv

Implementation: Material

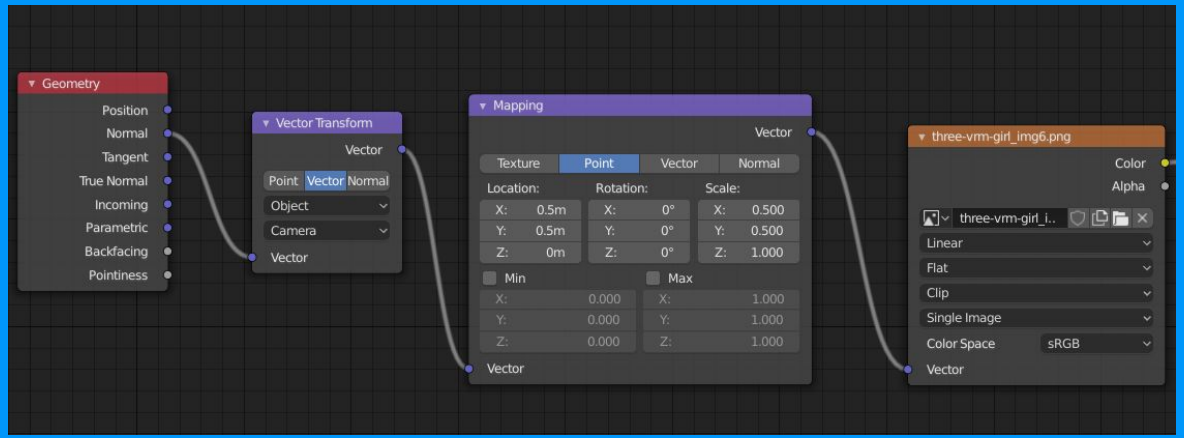


actual code of MToon of our implementation

pixiv

Implementation: Material

opinion: Make it nodable!



Blender

pixiv

Implementation: Material

Three.js : NodeMaterial

Can be converted into a JSON

```
const material = new THREE.StandardNodeMaterial();

// Basic material properties.
material.color = new THREE.ColorNode( 0xffffff * Math.random() );
material.metalness = new THREE.FloatNode( 0.0 );
material.roughness = new THREE.FloatNode( 1.0 );

const [ MUL, ADD ] = THREE.OperatorNode;
const localPosition = new THREE.PositionNode();
const localY = new THREE.SwitchNode( localPosition, 'y' );

// Modulate vertex position based on time and height.
// GLSL: vPosition *= sin( vPosition.y * time ) * 0.2 + 1.0;
let offset = new THREE.Math1Node(
  new THREE.OperatorNode( localY, time, MUL ),
  THREE.Math1Node.SIN
);
offset = new THREE.OperatorNode( offset, new THREE.FloatNode( 0.2 ), MUL );
offset = new THREE.OperatorNode( offset, new THREE.FloatNode( 1.0 ), ADD );

material.position = new THREE.OperatorNode( localPosition, offset, MUL );
```

https://www.donmccurdy.com/2019/03/17/three-nodematerial-introduction/

pixiv

Performance issues

Performance issues

VRoid Hub is meant to be visited
from both PC and mobile

Make our experience tolerable on mobile is
our primary challenge on VRoid Hub

pixiv

Performance issues

Outlines looks pretty grisly without antialias!

(with AA)

Performance issues

We cannot use multisample AA on WebGL
(can use in WebGL2 though)

→ We're rendering our entire thing
on 2x resolution + using post process FXAA

→ 🔥 on mobile devices 😩

pixiv

Performance issues

We wanted to implement
adaptive quality control on our 3D renderer

PCs w/
decent GPU

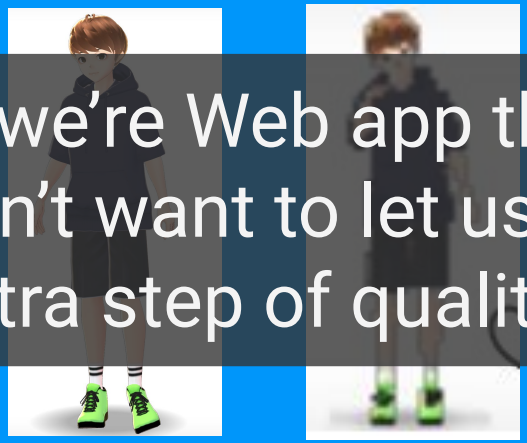
poor mobile
devices

(yes, this it exaggerated a bit)

pixiv

Performance issues

We wanted to implement
adaptive quality control on our 3D renderer



Since we’re Web app then
We don’t want to let users
takes one extra step of quality settings

PCs w/
decent GPU

poor mobile
devices

pixiv

Performance issues

Inspecting performance on user end is hard!

An option: See an actual FPS on the end

→ FPS can be subject of
various condition other than actual performance
e.g. orientation changes, spikes of other apps...

pixiv

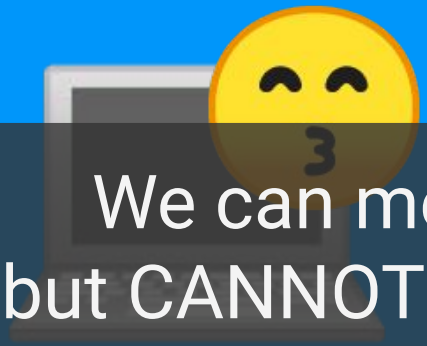
Performance issues

60FPS
(1ms)

60FPS
(16ms)

pixiv

Performance issues



We can measure CPU time
but CANNOT measure GPU time

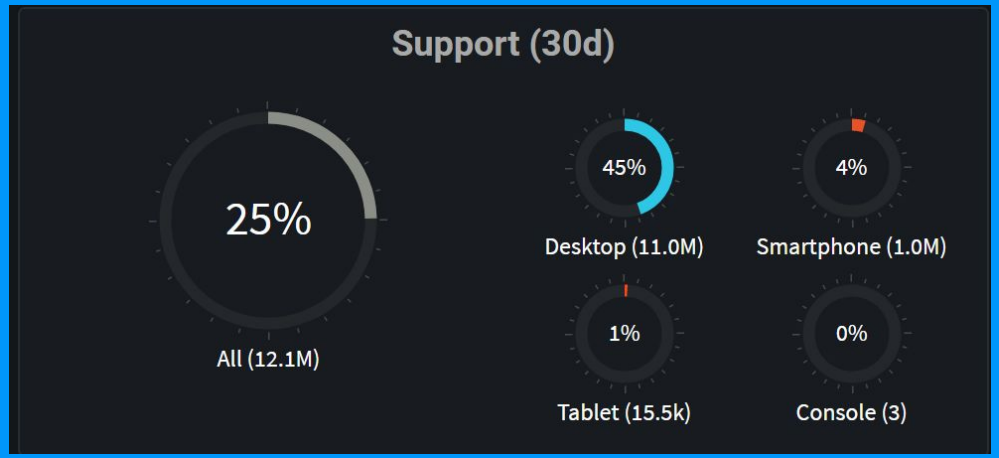
60FPS
(1ms)

60FPS
(16ms)

pixiv

Performance issues

There is a WebGL extension called EXT_disjoint_timer_query but...



http://webglstats.com/webgl/extension/EXT_disjoint_timer_query

pixiv

Performance issues

We are currently seeing FPS
to perform adaptive quality regression
with extremely low threshold (30FPS) 😩

pixiv

Security issues

Security issues

It’s relatively harder than Native applications
to protect users creation on Web

We are trying to protect
creators stuff at all costs!

pixiv

Security issues

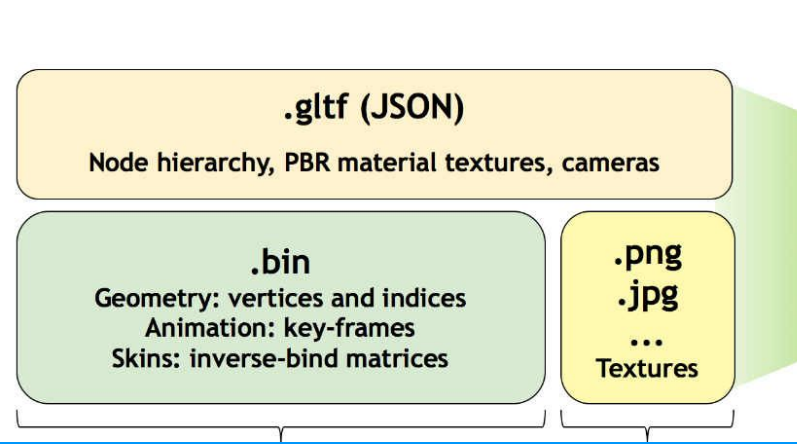
Example: We have to deal with models
that is on sale / has paid texture

There are cultures that artists sells
those own texture on store
since textures are sometimes compatible
between same authoring tool (e.g. VRoid Studio)

pixiv

Security issues

We’re using .glb instead of .gltf
+ it’s encrypted



https://www.khronos.org/gltf/

pixiv

Security issues

What we need to do
to load a texture from glTF (glb):

- Fetch the glTF binary from server
- Extract PNG/JPEG chunk
- Set the image to a texture

Security issues

We don’t want to expose
texture request on DevTools!

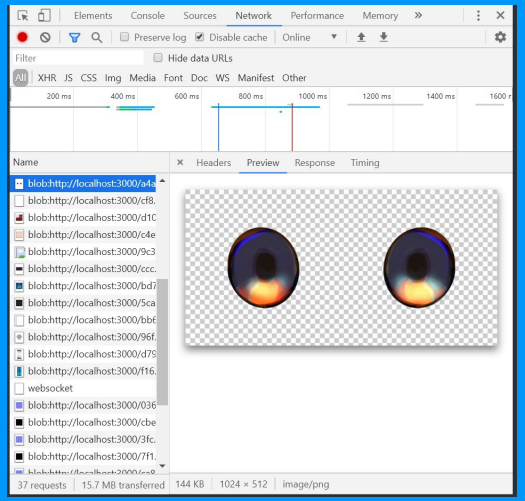
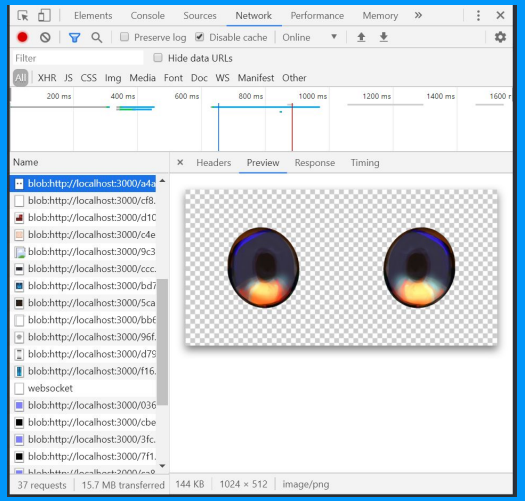


Image requests
are exposed on it
even if it was
loaded as a blob
(via TextureLoader of Three.js)

pixiv

Security issues

We don’t want to expose
texture request on DevTools!



→ We decoded these textures by ourselves

pixiv

Security issues

WebCodecs API has been proposed
several months ago

https://github.com/pthatcherg/web-codecs

```
{Audio, Video}{TrackReader, TrackWriter, Encoder, Decoder}
```

It would be nice if we can have
the image decoder as an API

pixiv

Wrap Up

- We are doing the Web service VRoid Hub and promoting use of 3D avatars on Web
- Implementation of materials are hard and it needs more portability
- There are still many issues to do 3D platform as a Web application

pixiv