

Voxel Airplanes demo

Oleksandr Popov

July 11, 2023

KHRONOS[®]
GROUP

WEBINARS
& MEETUPS



Voxel Airplanes WebGL demo

- Demo has stylized old-skool low-fidelity art direction - voxels, unfiltered textures
- Live page - <https://keaukraine.github.io/webgl-voxel-airplanes/index.html>
- Sources - <https://github.com/keaukraine/webgl-voxel-airplanes>
- Implemented with a low-level custom WebGL framework
 - Framework is identical to Java Android framework
 - There is the same Android live wallpaper app
- Optimizations of Web app:
 - Smallest data size
 - Fast async, lazy loading
 - Minimize overdraw
 - Shaders
- Optimizations of Android app:
 - Compressed textures (ASTC/ETC2)



Scene composition

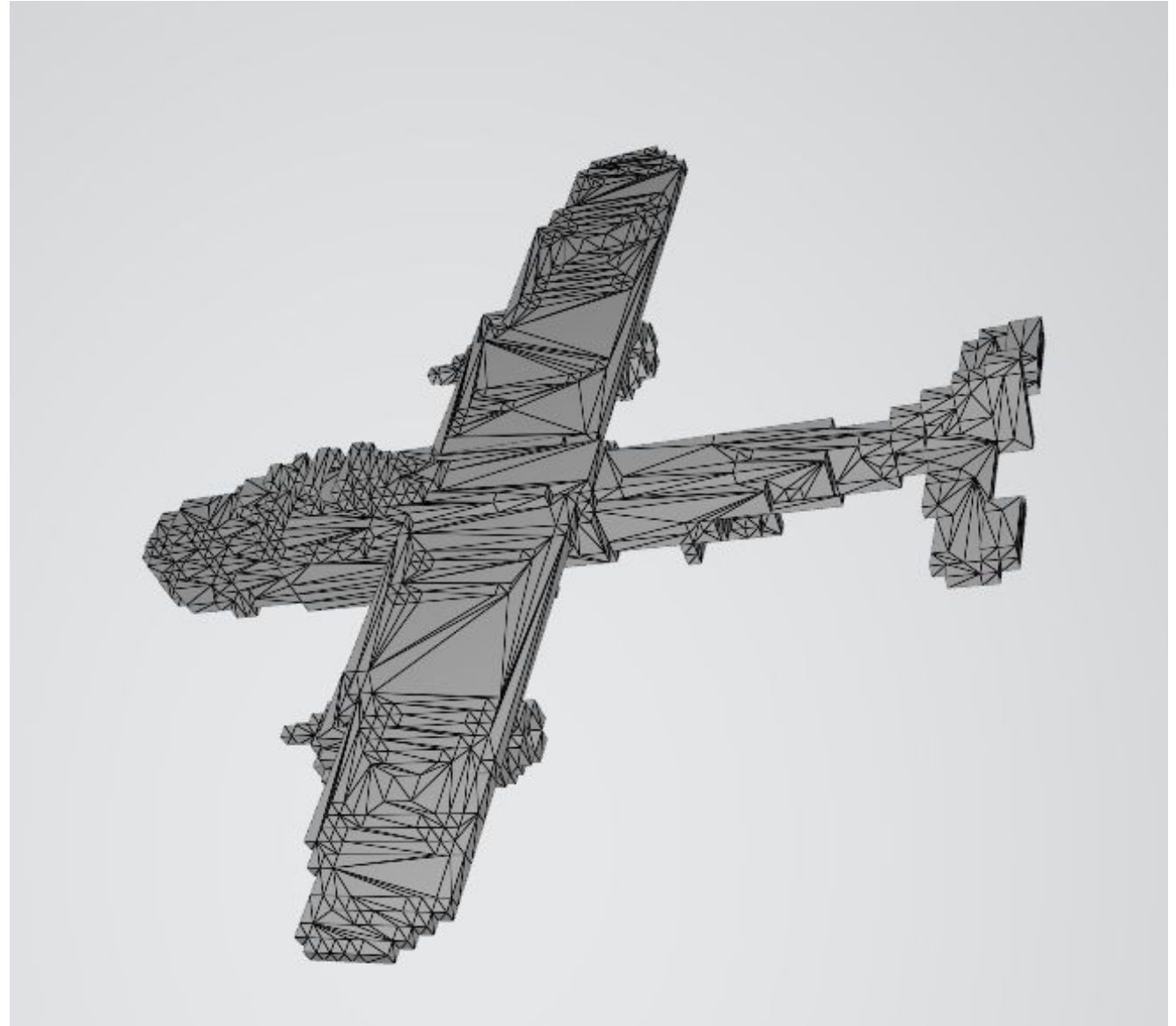
- Airplanes
 - Body
 - Propellers
 - Glass
- Ground plane
- Clouds
- Wind stripes



Airplanes

Models are exported from MagicaVoxel as conventional meshes:

- Position
- Normal
- UV (palette texture)



Planes vertex data optimization

- Plane models have the largest data size in scene:
 - ground texture is ~30 kB WebP
 - vertices + indices for each plane is ~250 kB unoptimized, ~50 kB optimized
- Voxel models allow to use some assumptions to:
 - omit some data
 - use more compact data types
 - pack data

Planes vertex data optimization

<i>Position</i>			<i>UV</i>		<i>Normal</i>		
X	Y	Z	U	V	X	Y	Z
FP32	FP32	FP32	FP32	FP32	FP32	FP32	FP32

32 bytes

Original unoptimized FP32 data

Planes vertex data optimization

<i>Position</i>			<i>UV</i>		<i>Normal</i>		
X	Y	Z	U	V	X	Y	Z
FP16	FP16	FP16	FP16	FP16	FP16	FP16	FP16

16 bytes

WebGL 2 supports half floats, use them

Planes vertex data optimization

<i>Position</i>			<i>UV</i>	<i>Normal</i>			<i>Padding</i>
X	Y	Z	U	X	Y	Z	2 bytes
FP16	FP16	FP16	FP16	FP16	FP16	FP16	

16 bytes

Palette texture is 1D, omit V coordinate:



Planes vertex data optimization

<i>Position</i>			<i>UV</i>	<i>Normal</i>			<i>Padding</i>	
X	Y	Z	U	X	Y	Z	1 byte	1 byte
FP16	FP16	FP16	ubyte	sbyte	sbyte	sbyte		

12 bytes

Convert normals+UVs to bytes

Texture resolution is less than 1x256, all normals are axis-aligned

Planes vertex data optimization

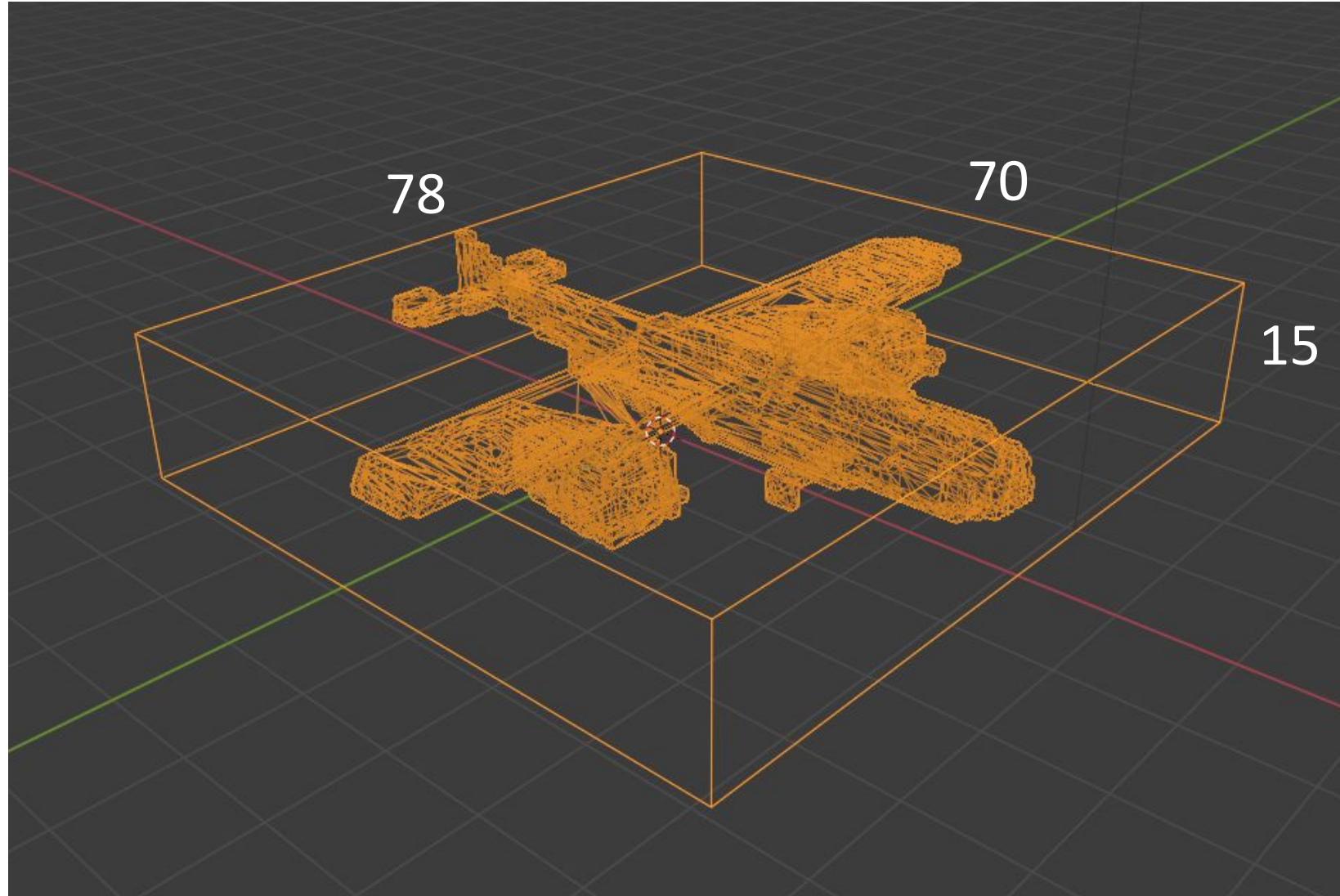
<i>Position</i>			<i>UV</i>	<i>Normal</i>			<i>Padding</i>
X	Y	Z	U	X	Y	Z	1 byte
sbyte	sbyte	sbyte	ubyte	sbyte	sbyte	sbyte	

8 bytes

Use bytes for position

Vertices are within -128...128 range and are snapped to voxel grid.

Planes vertex data optimization



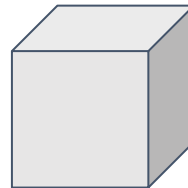
Planes vertex data optimization

<i>Position</i>			<i>UV</i>	<i>Normal</i>	<i>Padding</i>		
X	Y	Z	U	i	1 byte	1 byte	1 byte
sbyte	sbyte	sbyte	ubyte	ubyte			

8 bytes

Use index for normals

There are only 6 variations of normals for cube:



Planes vertex data optimization

<i>Position</i>			<i>UV</i>	<i>Normal</i>	<i>Padding</i>		
X	Y	Z	U	i	1 byte	1 byte	1 byte
sbyte	sbyte	sbyte	ubyte	ubyte			

8 bytes

Vertex data must be multiple of 4

If only we can eliminate 1 more byte...

Planes vertex data optimization

<i>Position</i>			<i>Normal Color</i>
X	Y	Z	i
sbyte	sbyte	sbyte	ubyte

4 bytes

Pack color and normal indices

Planes vertex data optimization

<i>Position</i>			<i>Normal</i> <i>Color</i>
X	Y	Z	i
sbyte	sbyte	sbyte	ubyte

<i>Color index</i>					<i>Normal index</i>		
4	3	2	1	0	2	1	0

Planes vertex data optimization

Unpacking color and normal indices in GLSL:

```
in uint rm_NormalColor; // Packed normal + color indices: CCCCNNN
...
const vec3 NORMALS[6] = vec3[6](...);
...
uint normalIndex = rm_NormalColor & 7u; // up to 8 normals
uint colorIndex = rm_NormalColor >> 3u; // up to 32 colors
vec3 normalValue = NORMALS[normalIndex];
```

Bitwise operators are available only in OpenGL ES 3.0+ and WebGL 2.

Planes vertex data size savings

8x

vs

naive FP32 data

2x

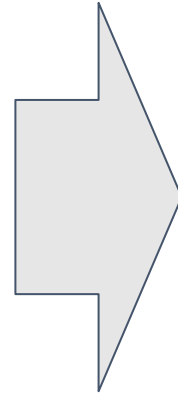
vs

best non-packed

Planes shader optimization

Vertex shader

- Unpacks vertex data
- Calculates position
- Gets color from palette
- Applies directional lighting



Fragment shader

- Uses color from VS

Ground filtering

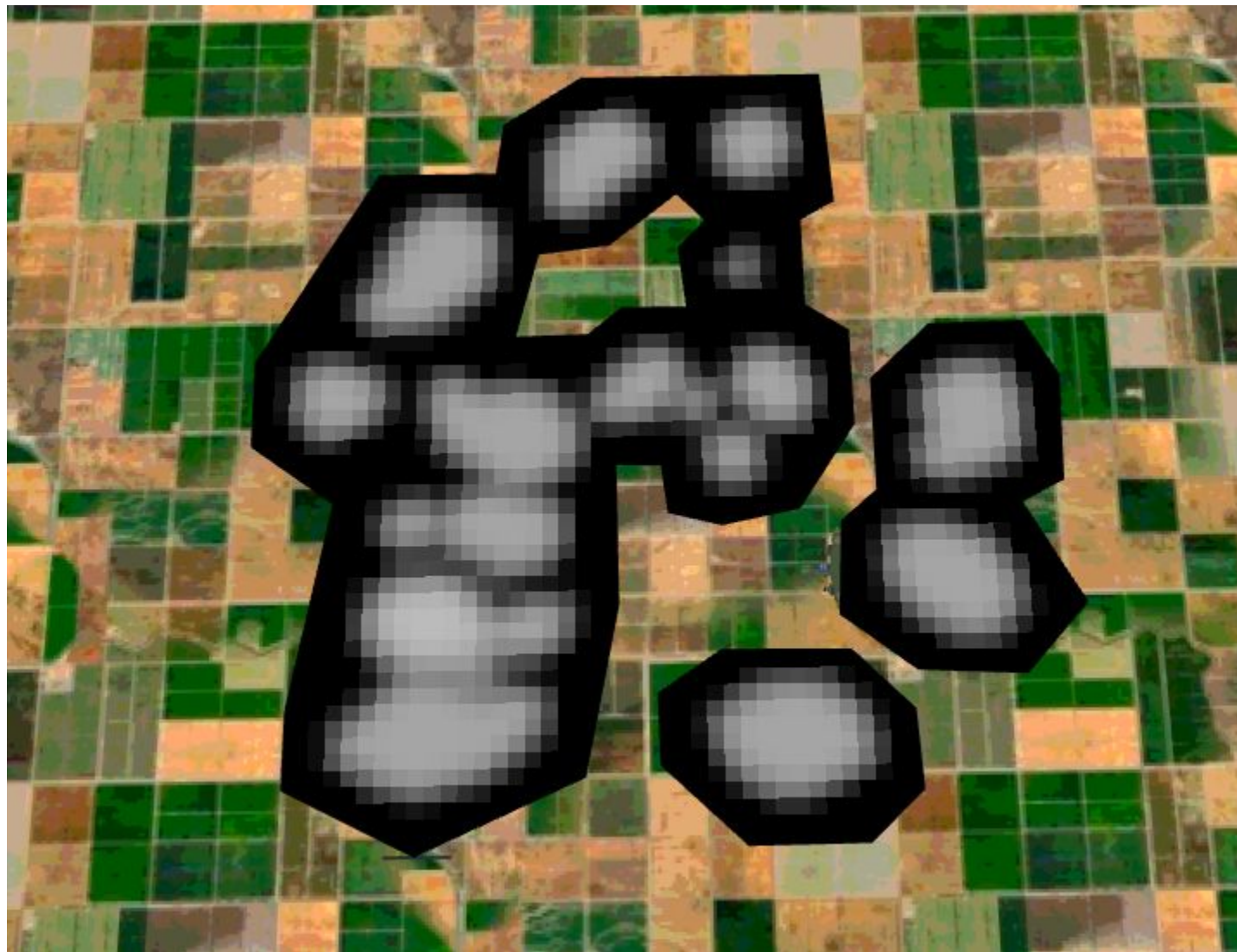


Nearest neighbour




Custom antialiased blocky filtering

Reducing clouds overdraw



Data size

- Initial web page load size - 155 kB.
- All models and textures (lazily) loaded - 1.05 MB.
- Fits on a floppy disk 

Thank you for watching!

Links to materials used in this presentation:

- Live demo page - <https://keaukraine.github.io/webgl-voxel-airplanes/index.html>
- Source code - <https://github.com/keaukraine/webgl-voxel-airplanes>
- Article with additional and more detailed explanations of this demo rendering pipeline - <https://keaukraine.medium.com/voxel-airplanes-3d-webgl-demo-d210d5dfa54e>
- Original airplanes models are by Max Parata - <https://maxparata.itch.io/voxel-plane>
- Blocky filtering shader - <https://www.shadertoy.com/view/ltfXWS>

A recording of this presentation will be available at
<https://www.khronos.org/events/webglwebgpu-meetup-july-2023>

For more information on WebGL, please visit
<https://www.khronos.org/webgl>

Email: public_webgl@khronos.org



WEBINARS
& MEETUPS

