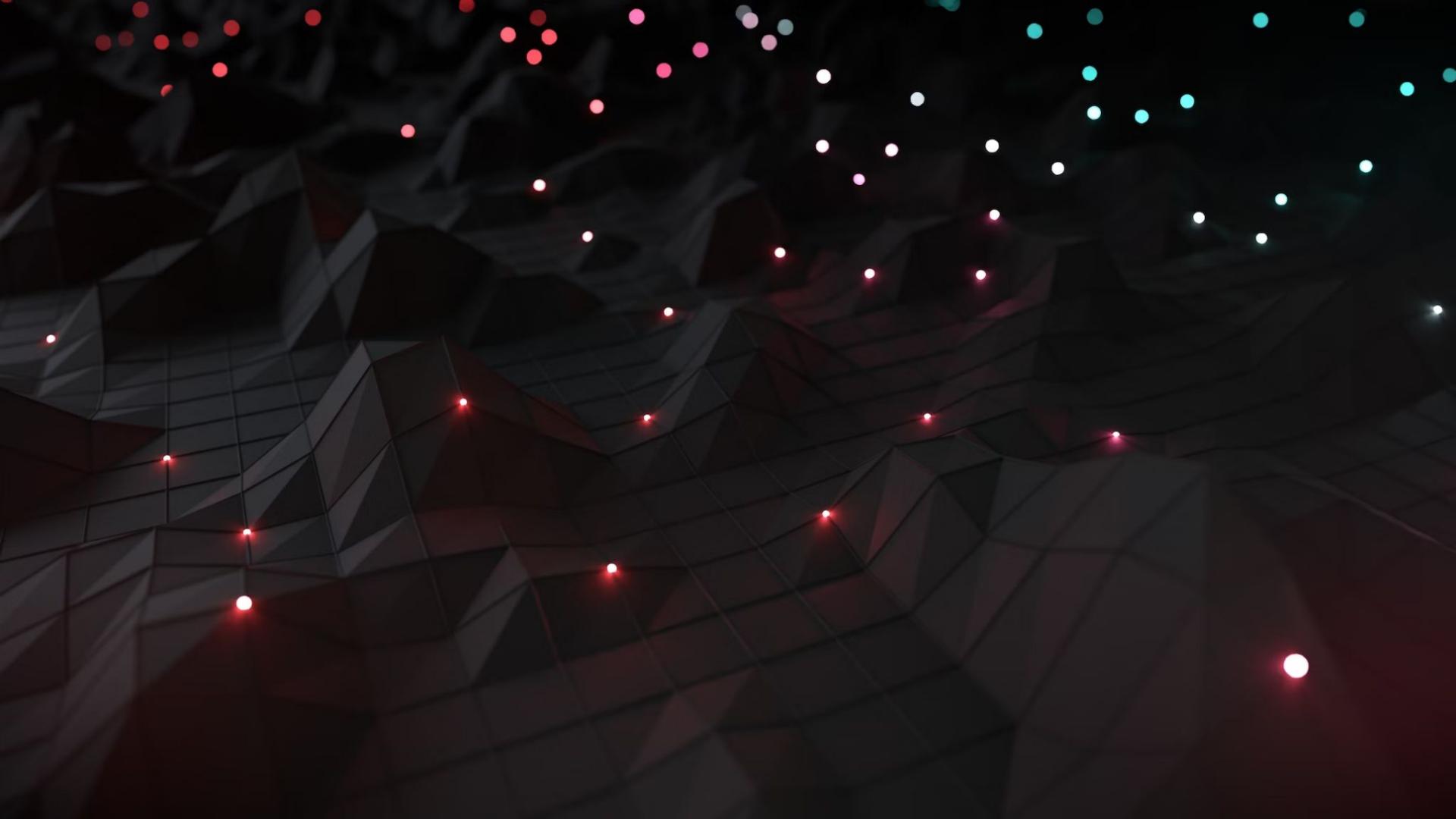


Flame & *(real)* 3D

A deeper dive



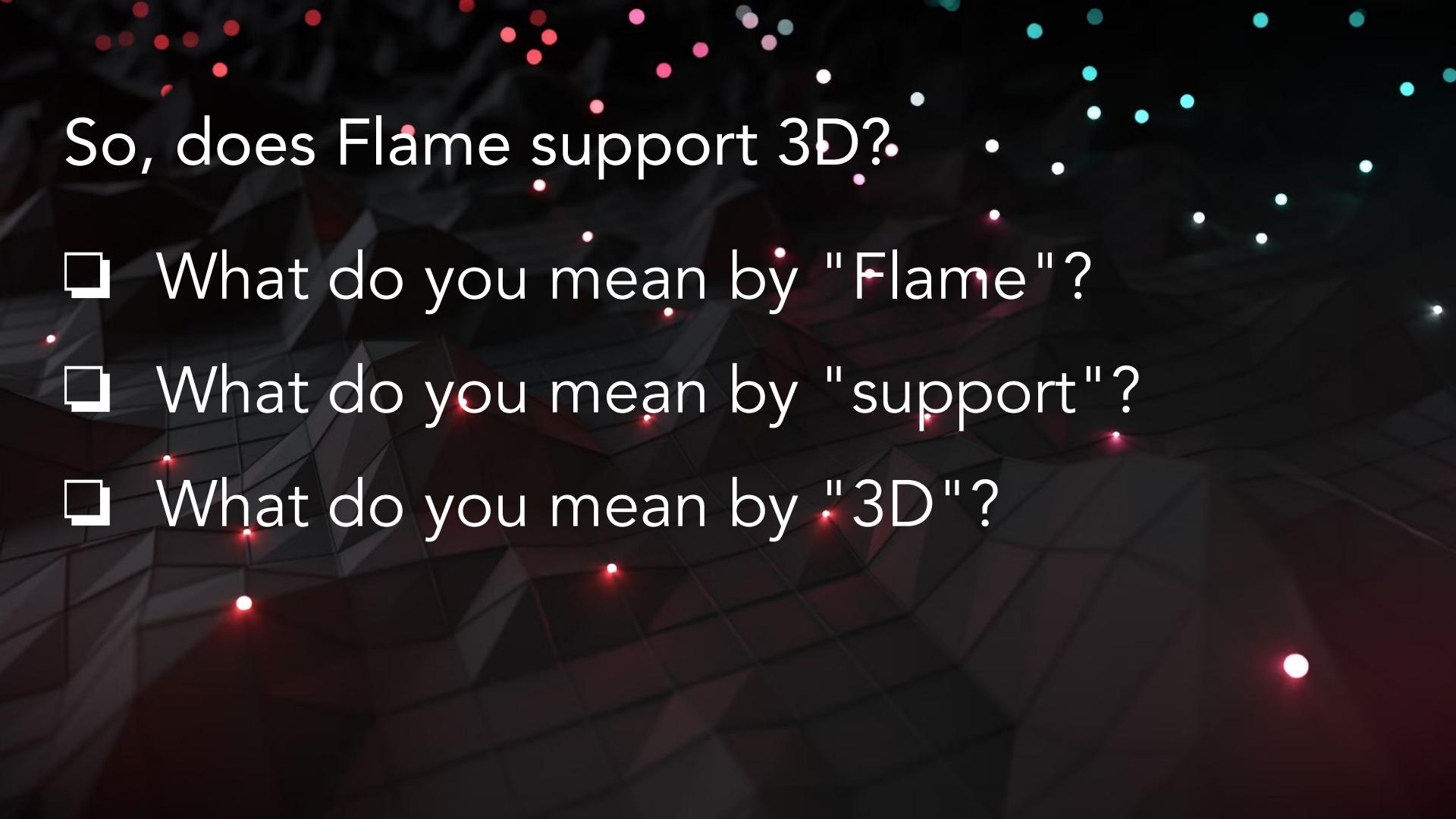
Luan Nico



⊕ Flame and (pseudo) 3d



Luan
Nico



So, does Flame support 3D?

- ❑ What do you mean by "Flame"?
- ❑ What do you mean by "support"?
- ❑ What do you mean by "3D"?

Can I run Dart code in the background of an Flutter app?

Yes, you can run Dart code in a background process on both iOS and Android. For more information, see the free Medium article [Executing Dart in the Background with Flutter Plugins and Geofencing](#).

Can I use JSON/XML/protobufs, etc. with Flutter?

Absolutely. There are libraries on [pub.dev](#) for JSON, XML, protobufs, and many other utilities and formats.

For a detailed writeup on using JSON with Flutter, check out the [JSON tutorial](#).

Can I build 3D (OpenGL) apps with Flutter?

Today we don't support for 3D via OpenGL ES or similar. We have long-term plans to expose an optimized 3D API, but right now we're focused on 2D.

natively

Why is my APK or IPA so big?

Usually, assets including images, sound files, fonts, etc, are the bulk of an APK or IPA. Various tools in the Android and iOS ecosystems can help you understand what's inside of your APK or IPA.

Also, be sure to create a *release build* of your APK or IPA with the Flutter tools. A release build is usually *much* smaller than a *debug build*.

Learn more about creating a [release build of your Android app](#), and creating a [release build of your iOS app](#). Also, check out [Measuring your app's size](#).

Getting started with Flutter GPU

Build custom renderers and render 3D scenes in Flutter.



Brandon DeRosier · [Follow](#)

Published in [Flutter](#) · 17 min read · Aug 6, 2024



--



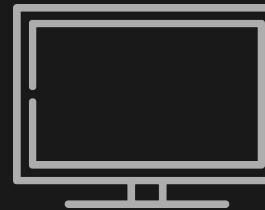
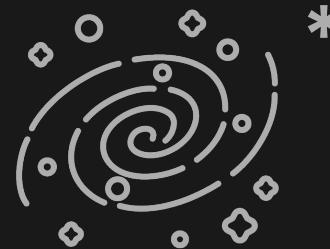
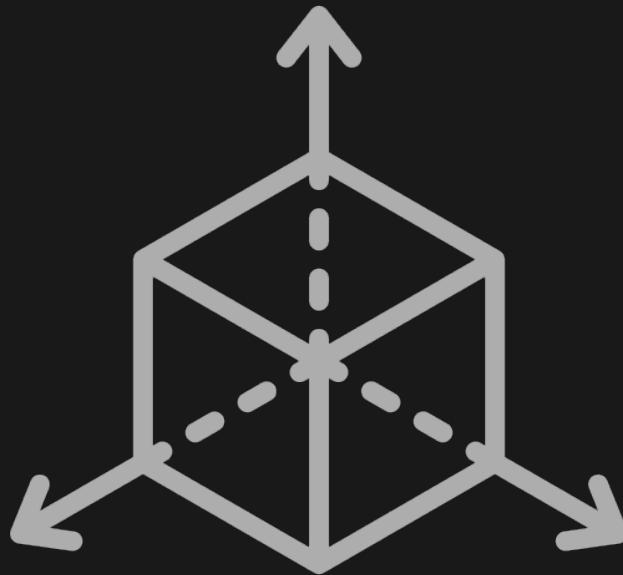
6

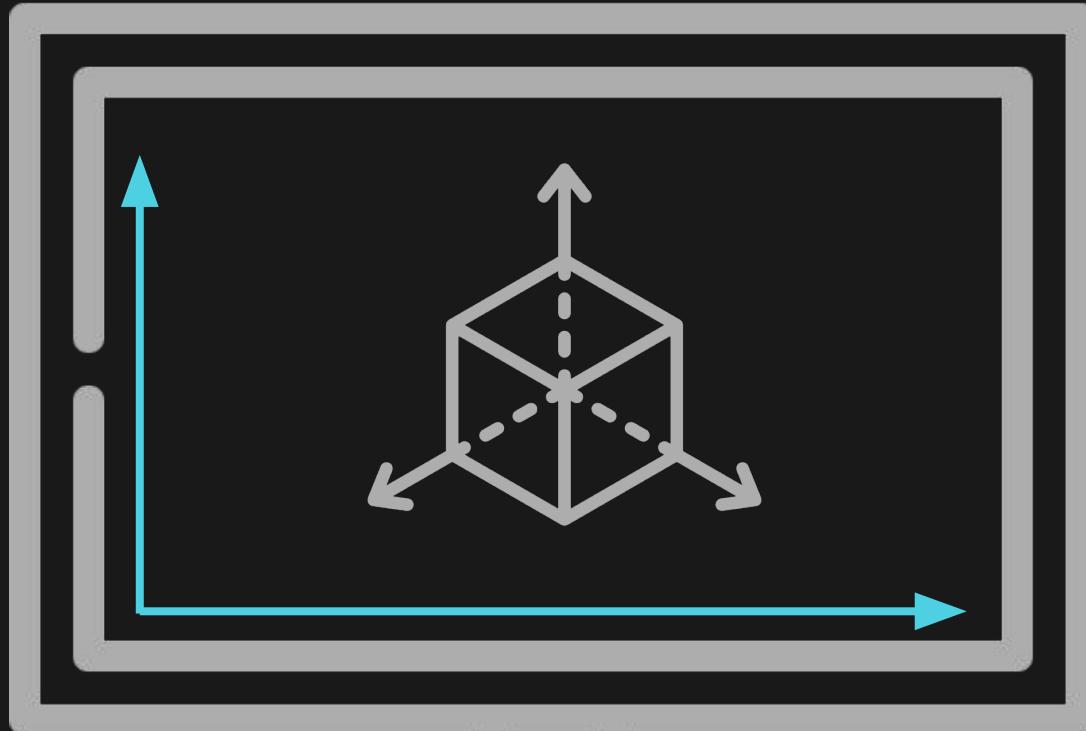


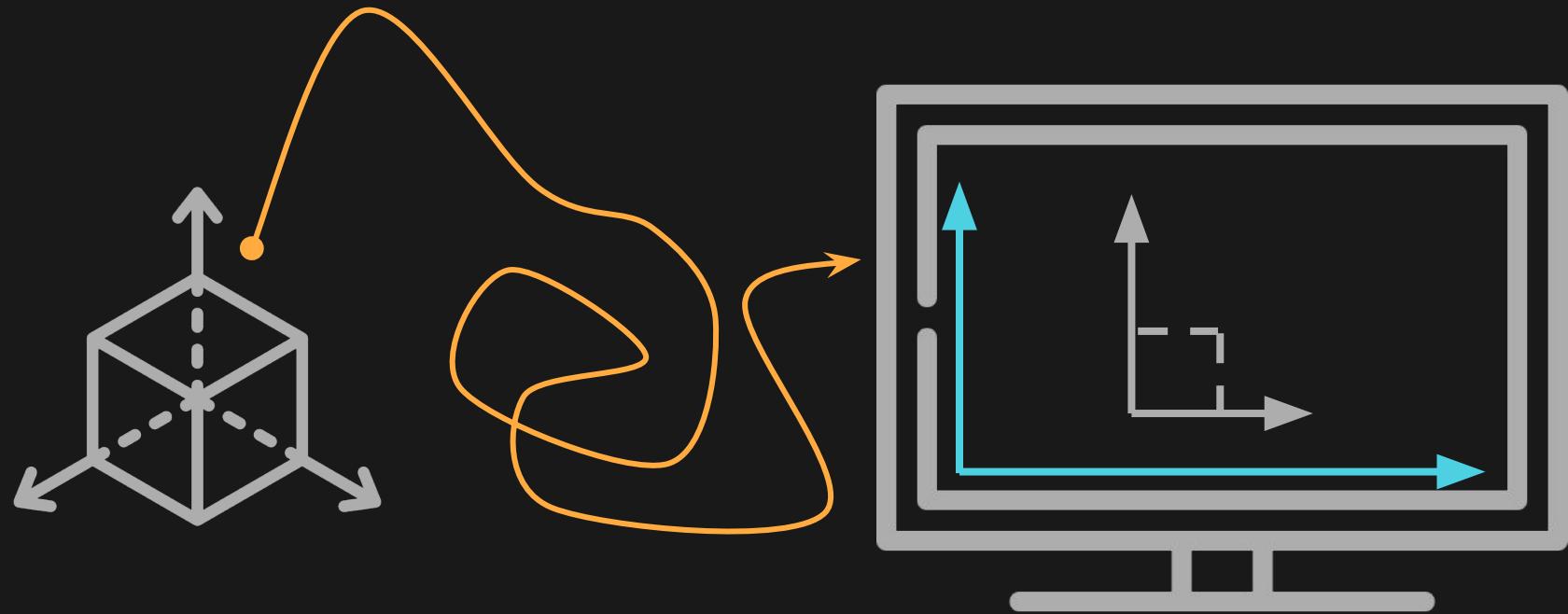
...



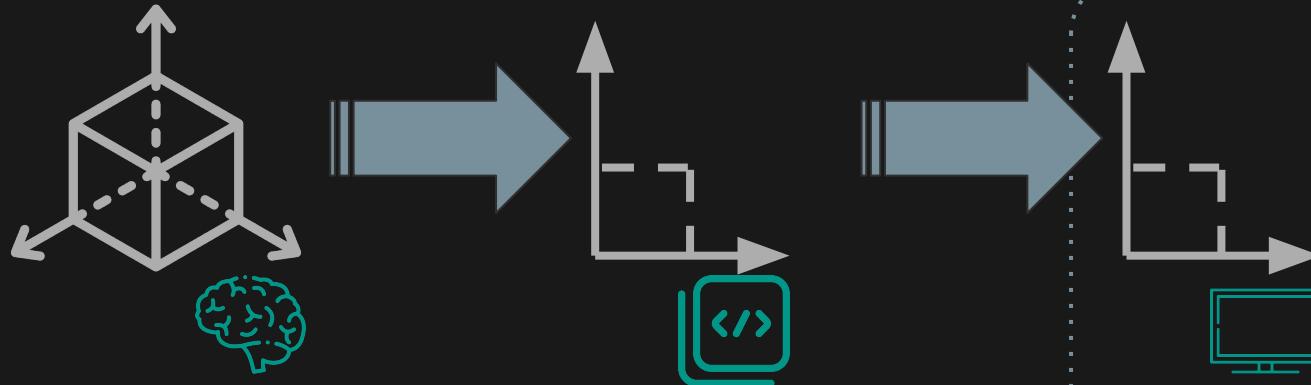
What is 3D?



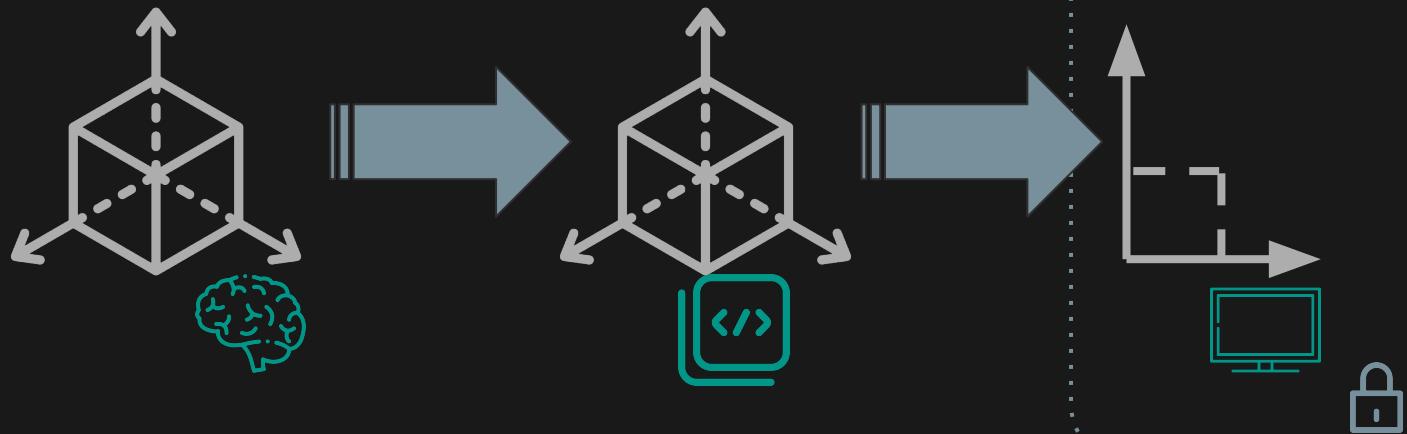




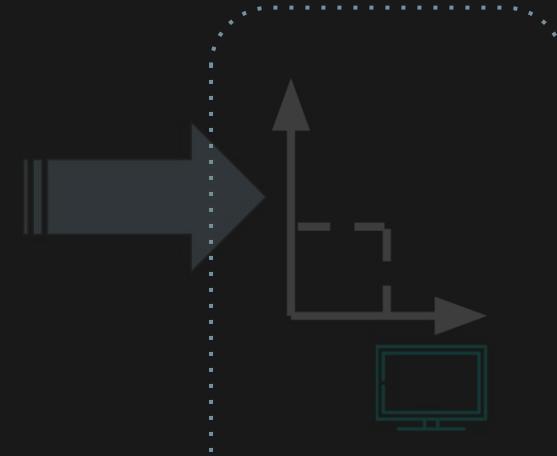
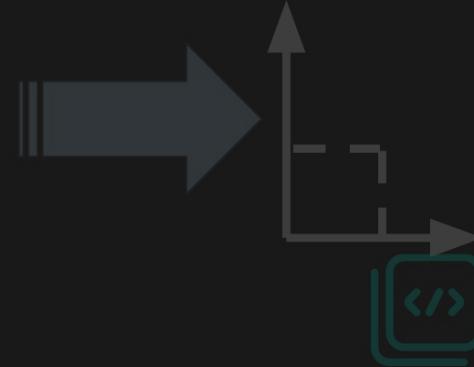
1



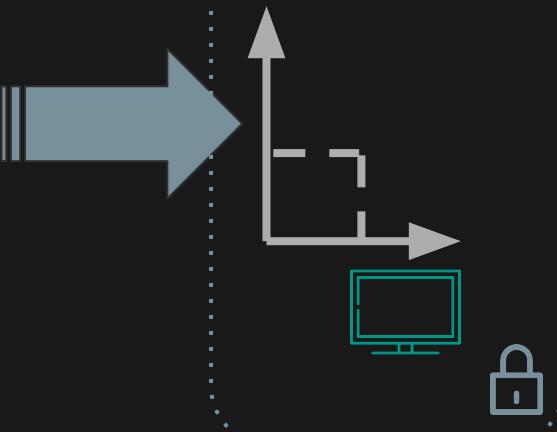
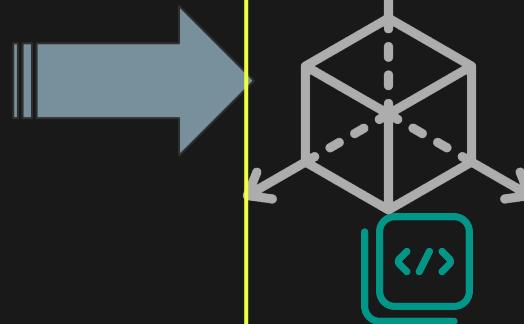
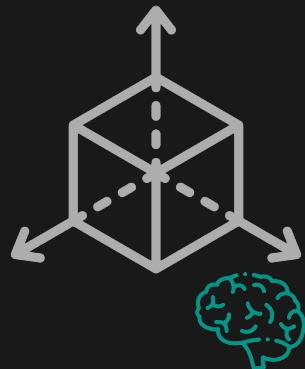
2



1



2



2D Worlds



Quick Recap - 2D Worlds

- **Camera:** World \rightarrow Screen Coordinates
- **Transform Matrices**

```
canvas.transform(  
    (Matrix4.identity()  
        ..scale(scale.x, scale.y, 1)  
        ..translate2(-cameraPosition + screenSize / 2))  
    .storage,  
);
```

$$M = \begin{pmatrix} S_x & 0 & 0 & | & T_x \\ 0 & S_y & 0 & | & T_y \\ 0 & 0 & S_z & | & T_z \\ 0 & 0 & 0 & | & 1 \end{pmatrix} \quad M \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix}$$

Quick Recap - 2D Worlds - "2.5D" Techniques

- Perspective
- Pre-rendered 3D sprites
- Parallax
- Isometric
- Sprite Stacking
- Raycasting



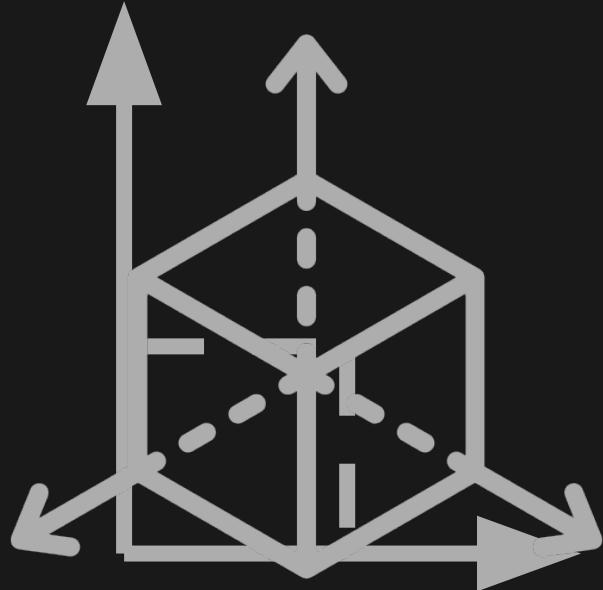
The (*Real*) 3D



The (*Real*) 3D?

3d rendering?

3D World Representation - Basics

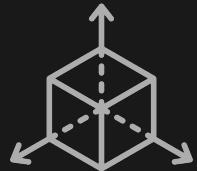


```
class Player {  
    Vector3 position;  
}
```



The (*Real*) 3D?

3D-world rendering?



The (*Real*) 3D?



Can I run Dart code in the background of an Flutter app?

Yes, you can run Dart code in a background process on both iOS and Android. For more information, see the free Medium article [Executing Dart in the Background with Flutter Plugins and Geofencing](#).

Can I use JSON/XML/protobufs, etc. with Flutter?

Absolutely. There are libraries on [pub.dev](#) for JSON, XML, protobufs, and many other utilities and formats.

For a detailed writeup on using JSON with Flutter, check out the [JSON tutorial](#).

Can I build 3D (OpenGL) apps with Flutter?

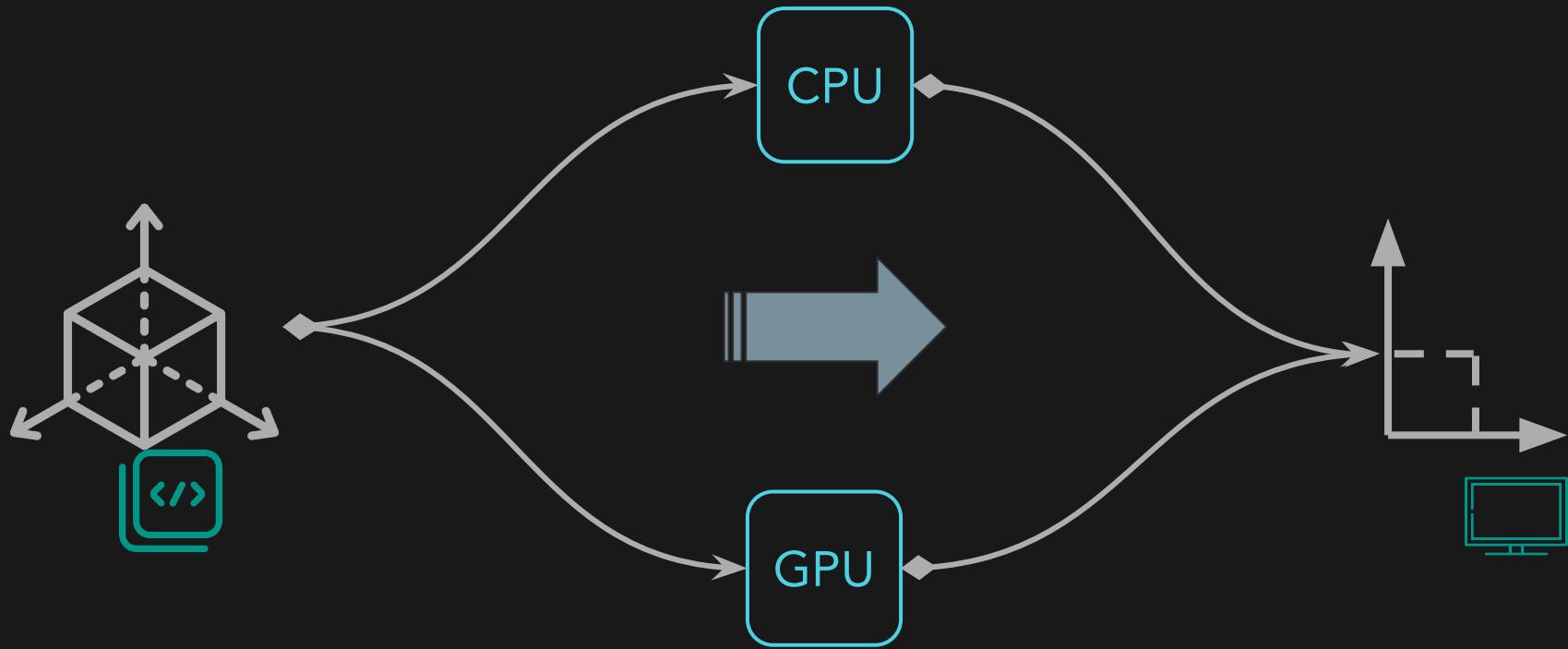
Today we don't support 3D via OpenGL ES or similar. We have long-term plans to expose an optimized 3D API, but right now we're focused on 2D.

Why is my APK or IPA so big?

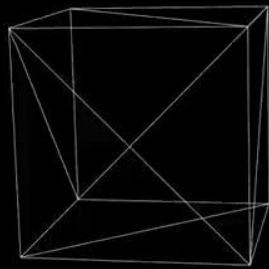
Usually, assets including images, sound files, fonts, etc, are the bulk of an APK or IPA. Various tools in the Android and iOS ecosystems can help you understand what's inside of your APK or IPA.

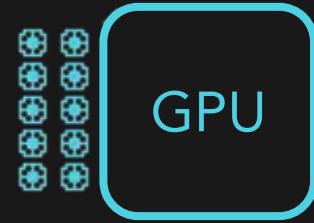
Also, be sure to create a *release build* of your APK or IPA with the Flutter tools. A release build is usually *much* smaller than a *debug build*.

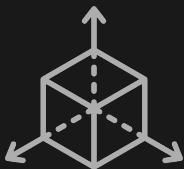
Learn more about creating a [release build of your Android app](#), and creating a [release build of your iOS app](#). Also, check out [Measuring your app's size](#).



Pitch: 0.00
Yaw: 0.00
Position: [0.00, 0.00, 0.00]







More Science, Less Programming

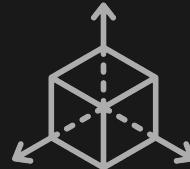


CUDA



The (*Real*) 3D?

gpu-accelerated 3D-world rendering



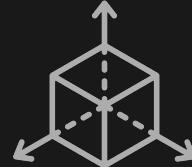
The (*Real*) 3D?

first-class gpu-accelerated 3D-world rendering



`flutter_gpu`

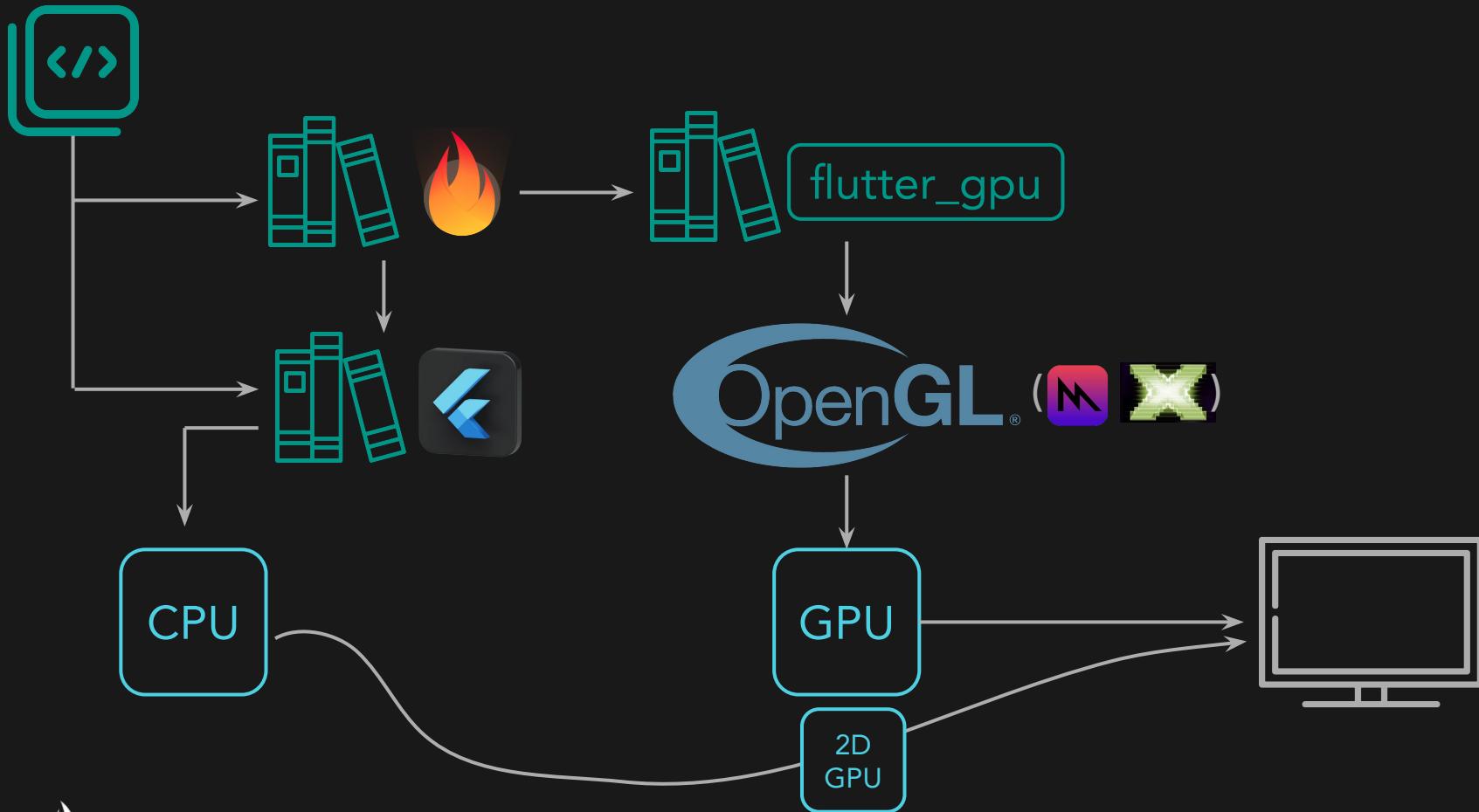
GPU



Can we do it?

Yes. * * * * *





flame_3d

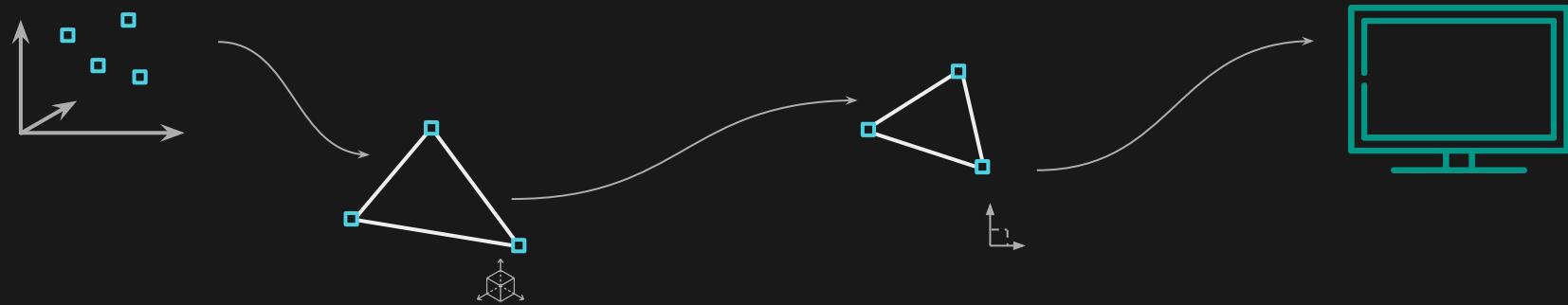
What we have

Preamble



Triangles

3D Projection



always has been

$$\begin{pmatrix} S_x & 0 & 0 & T_x \\ 0 & S_y & 0 & T_y \\ 0 & 0 & S_z & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

wait so it is
all matrices???



3D Projections - Transformations - TRS

$$M = \begin{pmatrix} S_x & 0 & 0 & T_x \\ 0 & S_y & 0 & T_y \\ 0 & 0 & S_z & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Vector3
translation;

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

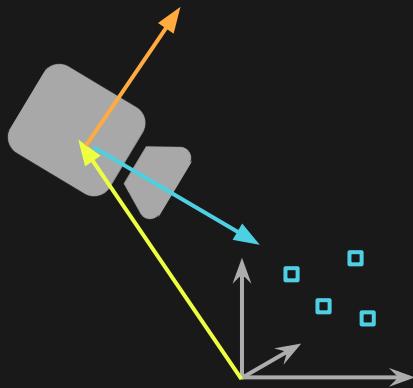
$$R_z(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Quaternion
rotation;

$$T = \begin{pmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

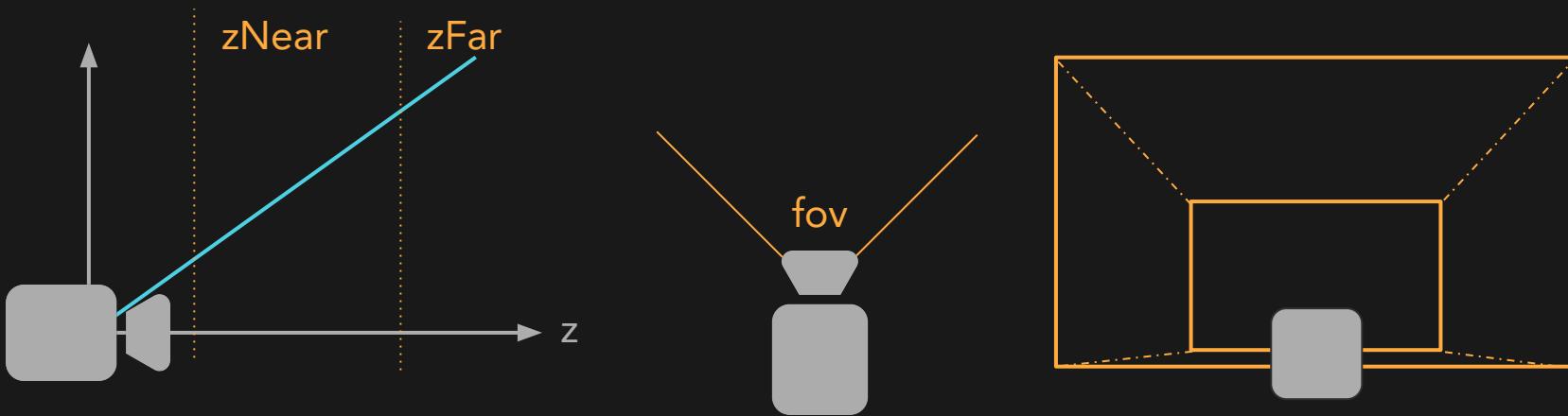
Vector3
scale;

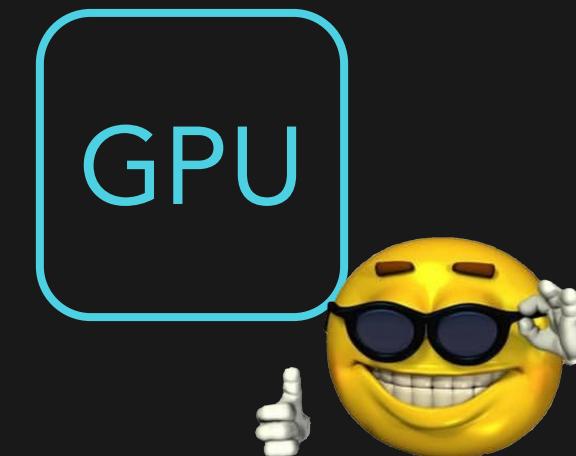
3D Projections - Camera



```
class Camera {  
    Vector3 position;  
  
    Vector3 direction;  
  
    Vector3 up;  
  
    // ...  
}
```

3D Projections - Perspective

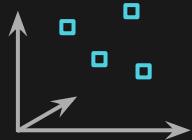




Meshes



Vertex



```
class Vertex {  
    Vertex({  
        required Vector3 position,  
        required Vector2 texCoord,  
        required Color color,  
        Vector3? normal,  
        Vector4? joints,  
        Vector4? weights,  
    }) {  
        // ...  
        this.storage = ...  
    }  
}
```



Shaders: The Key

shaders

GL spatial_material.frag

GL spatial_material.vert

~/projects/gamedev/flame/packages/flame_3d

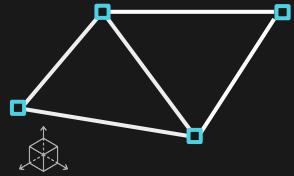
dart ./bin/build_shaders.dart

<https://learnopengl.com>

```
shaders > GL spatial_material.vert > ...
1  #version 460 core
2
3  in vec3 vertexPosition;
4  in vec2 vertexTexCoord;
5  in vec4 vertexColor;
6  in vec3 vertexNormal;
7
8  out vec2 fragTexCoord;
9  out vec4 fragColor;
10 out vec3 fragPosition;
11 out vec3 fragNormal;
12
13 uniform VertexInfo {
14     mat4 model;
15     mat4 view;
16     mat4 projection;
17 } vertex_info;
18
19 void main() {
20     // Calculate the modelview projection matrix
21     mat4 modelViewProjection = vertex_info.projection * vertex_info.view * vertex_info.model;
22
23     // Transform the vertex position
24     gl_Position = modelViewProjection * vec4(v0: vertexPosition, v1: 1.0);
25
26     // Pass the interpolated values to the fragment shader
27     fragTexCoord = vertexTexCoord;
28     fragColor = vertexColor;
29
30     // Calculate the world-space position and normal
31     fragPosition = vec3(value: vertex_info.model * vec4(v0: vertexPosition, v1: 1.0));
32     fragNormal = mat3(transpose(inverse(vertex_info.model))) * vertexNormal;
33 }
```

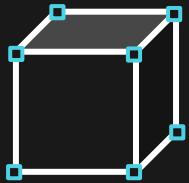
(no joints)

Surface



```
class Surface {  
    Surface({  
        required List<Vertex> vertices,  
        required List<int> indices,  
        Material? material;  
        Map<int, int>? jointMap;  
        /**  
         * If `true`, the normals will be  
         * calculated if not provided.  
         */  
        bool calculateNormals = true,  
    }) {  
        // ...  
    }  
}
```

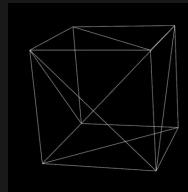
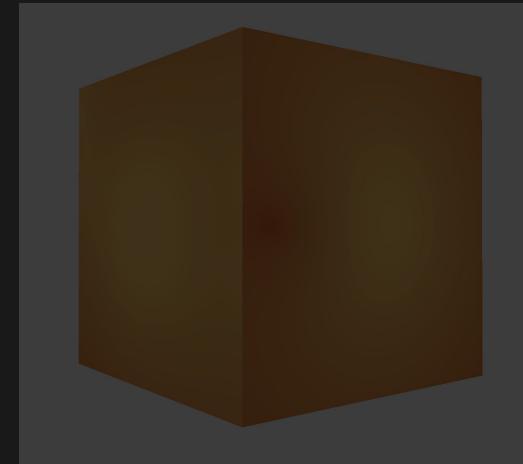
Mesh



```
class Mesh {  
    final List<Surface> surfaces;  
    Aabb3 get aabb;  
}
```

Simple Cube

```
await add(  
  MeshComponent(  
    mesh: CuboidMesh(  
      size: Vector3.all(8.0),  
    ),  
  ),  
);
```



```
final Vector3(:x, :y, :z) = size / 2;  
  
final vertices = [  
  // Face 1 (front)  
  vertex(  
    position: Vector3(-x, -y, -z),  
    texCoord: Vector2(0, 0),  
    normal: Vector3(0, 0, -1),  
  ),  
  vertex(  
    position: Vector3(x, -y, -z),  
    texCoord: Vector2(1, 0),  
    normal: Vector3(0, 0, -1),  
  ),  
  vertex(  
    position: Vector3(x, y, -z),  
    texCoord: Vector2(1, 1),  
    normal: Vector3(0, 0, -1),  
  ),  
  vertex(  
    position: Vector3(-x, y, -z),  
    texCoord: Vector2(0, 1),  
    normal: Vector3(0, 0, -1),  
  ),  
  
  // Face 2 (back)  
  vertex(  
    position: Vector3(-x, -y, z),  
    texCoord: Vector2(0, 0),  
    normal: Vector3(0, 0, 1),  
  ),  
  vertex(  
    position: Vector3(x, -y, z),  
    texCoord: Vector2(1, 0),  
    normal: Vector3(0, 0, 1),  
  ),  
  vertex(  
    position: Vector3(x, y, z),  
    texCoord: Vector2(1, 1),  
    normal: Vector3(0, 0, 1),  
  ),  
  vertex(  
    position: Vector3(-x, y, z),  
    texCoord: Vector2(0, 1),  
    normal: Vector3(0, 0, 1),  
  ),
```

Two Questions!

- Where does the lighting come from?
- Better way to define models?
- Your questions?

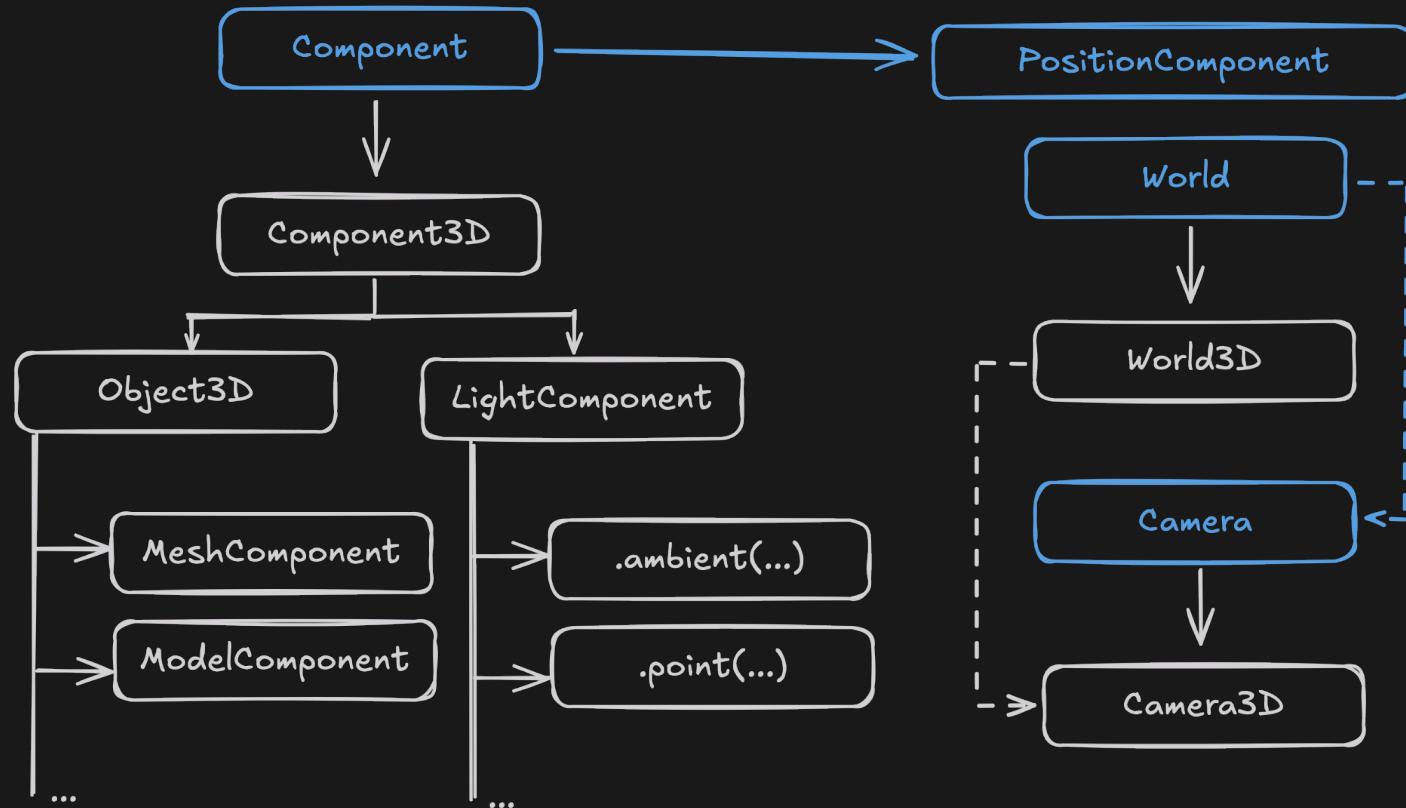
Lighting



LightComponent

```
await add(  
  LightComponent.ambient(  
    color: const Color(0xFFFFFFFF),  
    intensity: 0.75,  
  ),  
  
  LightComponent.point(  
    position: Vector3(...),  
    color: ...,  
    intensity: 20.0,  
  ),  
  // ...  
);
```





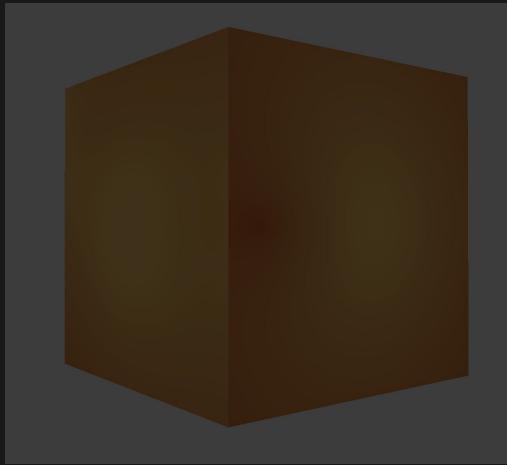
The Model Guides Us!

- PBR - Physics Based Rendering

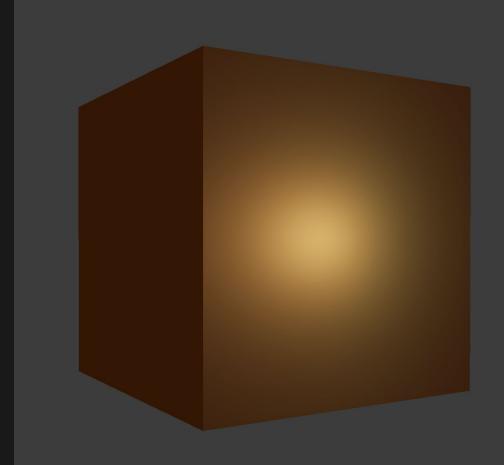
$$L_o(p, \omega_o) = \int_{\Omega} \left(k_d \frac{c}{\pi} + \frac{DFG}{4(\omega_o \cdot n)(\omega_i \cdot n)} \right) L_i(p, \omega_i) n \cdot \omega_i d\omega_i$$

- Arbitrary number of lights
- More advanced parameters
- Shadows!

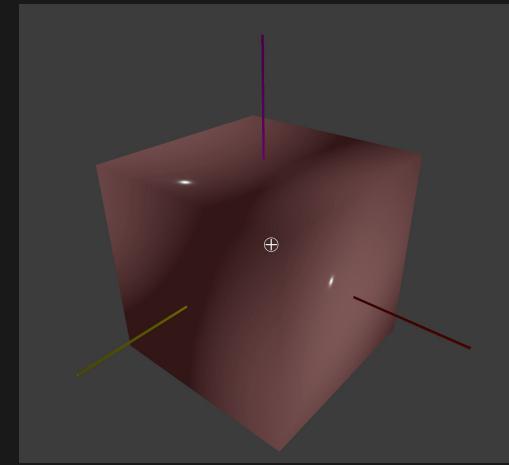
Face vs Vertex Normals



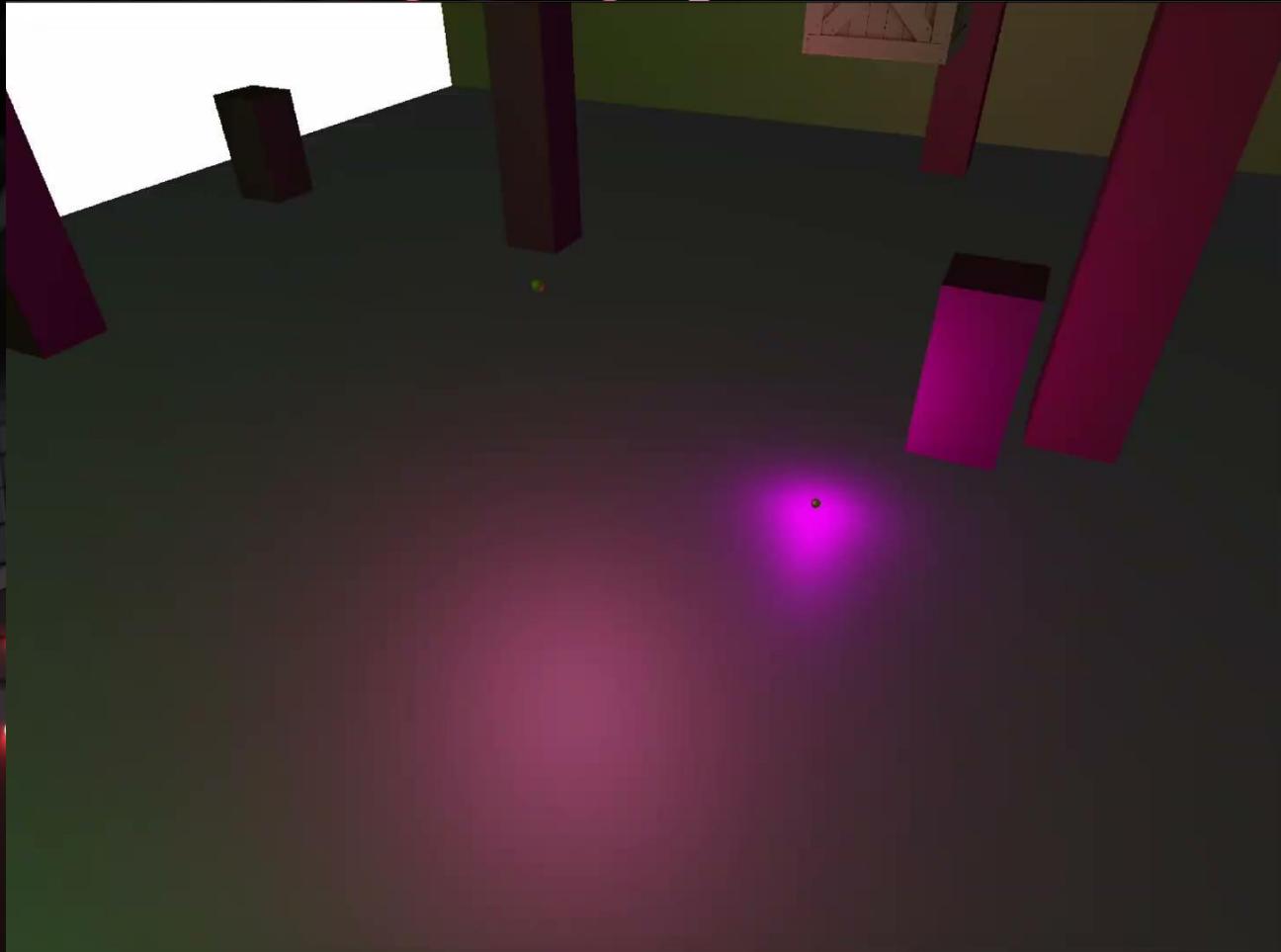
vertex normals



face normals

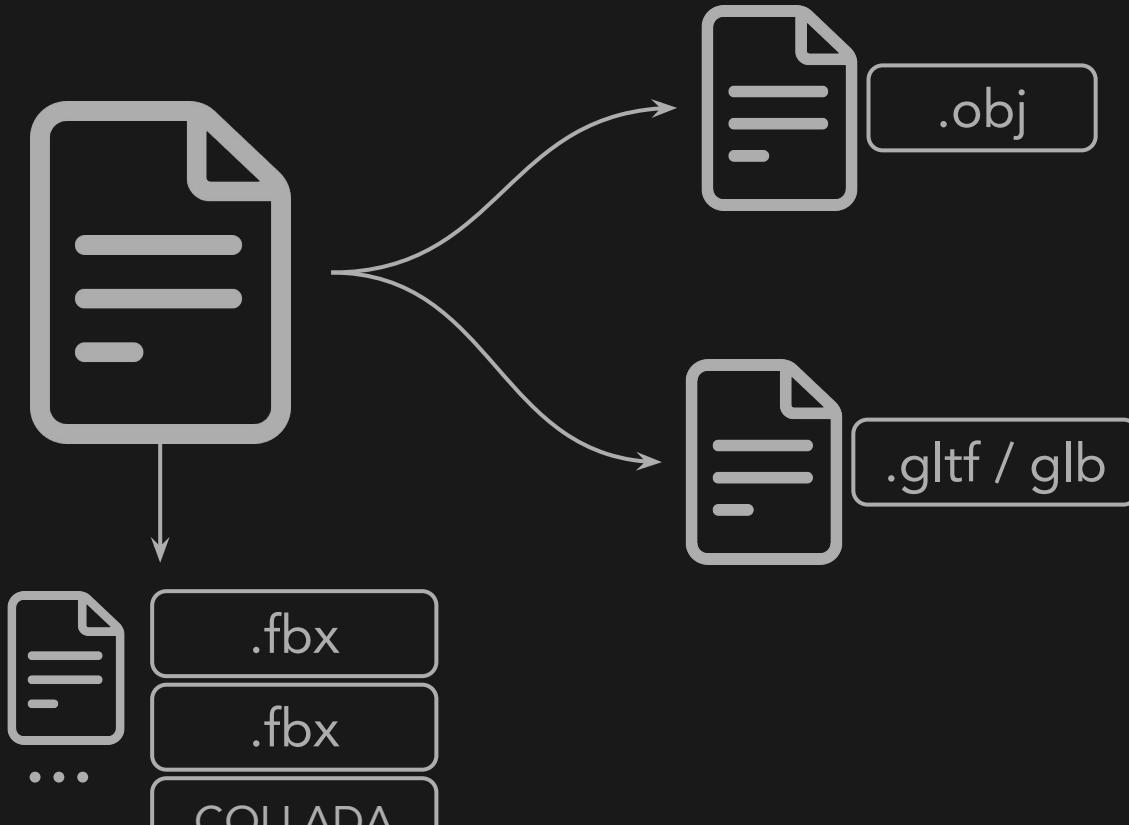


gltf model



Parsing Models





.obj file

```
# vertex (x y z)
v -0.000752473 0.044662 0.160067
# normal (x y z)
vn -0.146317 0.97078 0.194263
# texel (u v)
vt 0.123 0.456
# material
usemtl mat0
# surface (v/vt/vn)
f 1/1/1 2/1/1 3/1/1
```

Defend the Donut



.gltf / .glb files

- byte buffers!
- scenes, nodes, meshes
- textures
- animation
- skinning & skeletons

Model (and ModelComponent)

```
class Model {  
    final Map<int, ModelNode> nodes;  
    final Map<String, ModelAnimation> animations;  
    // ...  
}  
  
class ModelNode {  
    final int nodeIndex;  
    final int? parentNodeIndex;  
    final Matrix4 transform;  
    final Mesh? mesh;  
    final Map<int, ModelJoint> joints;  
    // ...  
}  
  
class ModelAnimation {  
    final String name;  
    final Map<int, NodeAnimation> nodes;  
    final double _lastTime;  
    double _clock = 0;  
    // ...  
}
```



What we don't have?

- Polish (flame_3d, joints, flame_3d_extras)
- SSBO (not even simple arrays)
- Shadows
- Physics
- Optimization
- A lot more...



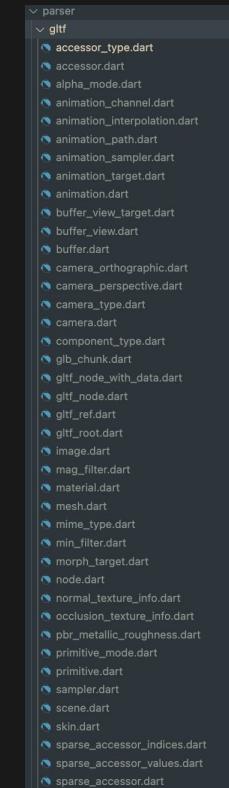
```
uniform JointMatrices {  
    mat4 joint0;  
    mat4 joint1;  
    mat4 joint2;  
    mat4 joint3;  
    mat4 joint4;  
    mat4 joint5;  
    mat4 joint6;  
    mat4 joint7;  
    mat4 joint8;  
    mat4 joint9;  
    mat4 joint10;  
    mat4 joint11;  
    mat4 joint12;  
    mat4 joint13;  
    mat4 joint14;  
    mat4 joint15;  
} joints;  
  
mat4 jointMat(float jointIndex) {  
    if (jointIndex == 0.0) {  
        return joints.joint0;  
    } else if (jointIndex == 1.0) {  
        return joints.joint1;  
    } else if (jointIndex == 2.0) {  
        return joints.joint2;  
    } else if (jointIndex == 3.0) {  
        return joints.joint3;  
    } else if (jointIndex == 4.0) {  
        return joints.joint4;  
    } else if (jointIndex == 5.0) {  
        return joints.joint5;  
    } else if (jointIndex == 6.0) {  
        return joints.joint6;  
    } else if (jointIndex == 7.0) {  
        return joints.joint7;  
    } else if (jointIndex == 8.0) {  
        return joints.joint8;  
    } else if (jointIndex == 9.0) {  
        return joints.joint9;  
    } else if (jointIndex == 10.0) {  
        return joints.joint10;  
    } else if (jointIndex == 11.0) {  
        return joints.joint11;  
    } else if (jointIndex == 12.0) {  
        return joints.joint12;  
    }  
}
```

Key Learnings

And why we need to refine flame_3d

Data Model is Everything!

- gltf as a "blueprint"
- typed data structure from the [docs](#)
- sample models from their [repo](#)



00:22



Donut Life
100 %

Shaders are Devilish

- Type-highly-unsafe
- "Optimize" the "fun" out of debugging
 - `return magenta;` trick
- Manual byte wiring

Cryptic Errors - Bytes Galore

Problem Report for example

example quit unexpectedly.

Click Reopen to open the application again. This report will be sent automatically to Apple.

Comments

Provide any steps necessary to reproduce the problem.

Problem Details and System Configuration

Translated Report (Full Report Below)

```
Process:           example [82751]
Path:             /Users/USER/*example.app/Contents/MacOS/example
Identifier:       com.example.example
Version:          0.1.0 (0.1.0)
Code Type:        ARM-64 (Native)
Parent Process:   launchd [1]
User ID:          501

Date/Time:        2024-08-30 16:18:08.3497 +0200
OS Version:      macOS 14.6.1 (23G93)
Report Version:   12
Anonymous UUID:  B5E23019-C94-C32F-885C-4D7360F97B1A

Sleep/Wake UUID: DC5C38A7-90C5-41D5-AC02-5732FAE18E94

Time Awake Since Boot: 1300000 seconds
Time Since Wake:    550 seconds

System Integrity Protection: enabled

Crashed Thread:   7  io.flutter.ui

Exception Type:  EXC_BAD_ACCESS (SIGSEGV)
Exception Codes: KERN_INVALID_ADDRESS at 0x0000000000000028
Exception Codes: 0x0000000000000081, 0x0000000000000028

Termination Reason: Namespace SIGNAL, Code 11 Segmentation fault: 11
Terminating Process: exc handler [82751]

VM Region Info: 0x28 is not in any region. Bytes before following region: 4336173016
REGION TYPE            START - END             [ VSIZE] PRT/MAX SHRMOD  REGION DETAIL
UNUSED SPACE AT START
-->  _TEXT             10274c000-102754000  [  32K] r-x/r-x  SM=COW  /Users/USER/*example.app/Contents/MacOS/
example
```

Hide Details ?

OK Reopen

Quaternion Interpolation

```
static Quaternion lerp(  
    Quaternion q0,  
    Quaternion q1,  
    double t,  
) {  
    return q0 + (q1 - q0).scaled(t);  
}
```



⊕

```
/// Some background on the correct shortest-path implementation can be found
/// here:
/// https://blog.magnum.graphics/backstage/the-unnecessarily-short-ways-to-do-a-quaternion-slerp/
static Quaternion _slerp(
    Quaternion q0,
    Quaternion q1,
    double t,
    double epsilon = 10e-3,
) {
    if (isEqual(q0, q1)) {
        return q0;
    }

    final dot = q0.dot(q1);
    if (1 - dot < epsilon) {
        // The quaternions are very close, so the linear interpolation algorithm
        // will be a good approximation.
        // This will prevent NaN from the slerp algorithm.
        return lerp(q0, q1, t);
    }

    final angle = acos(dot.abs());

    final q1Prime = dot >= 0 ? q1 : q1.scaled(-1);
    final a = sin((1 - t) * angle) / sin(angle);
    final b = sin(t * angle) / sin(angle);

    return q0.scaled(a) + q1Prime.scaled(b);
}
```

Order is Everything



🔥 FLAME

Thank you!
Questions?



- THE END -



Credits

<https://romannurik.github.io/SlidesCodeHighlighter/?theme=flutter-interact-19&font=Source+Code+Pro&tab=4&size=40&sel=focus>

itch.io assets

flaticon icons

Renan for the slide deck template

unsplash for the photos

<https://buttermilk.itch.io/tiny-wonder-farm-asset-pack>

Recommended Reading

<https://www.brainvoyager.com/bv/doc/UsersGuide/CoordsAndTransforms/SpatialTransformationMatrices.html>

<https://www.3dgep.com/understanding-the-view-matrix/>

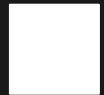
<https://www.youtube.com/watch?v=ih20l3pJoeU>

<https://www.scratchapixel.com/lessons/3d-basic-rendering/perspective-and-orthographic-projection-matrix/building-basic-perspective-projection-matrix.html>

<https://lodev.org/cgtutor/raycasting.html>



Color Palette



#FFFFFF



#78909C



#212121



#FFAB40



#303030



#4DD0E1



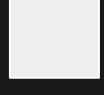
#ADADAD



#EEFF41



#009688



#EEEEEE



#191919

**More slides in case
people ask some
questions**

42



simple

modular

code first

open

hackable

FLAME



🔥 FLAME



VS



3D Projections - More Extras *

- ❖ Culling
- ❖ z-buffer
- ❖ Lights
- ❖ Textures
- ❖ Physics
- ❖ Much, much more

