

+Livecoding VJ

# コードから学ぶ GLSL

~暗黙知を習得する~

Renard

# はじめに

- 二部構成で、全体1時間30分ほどの講義になります(多分...)
  - ライブコーディング VJ:1時間 + コードから学ぶ GLSL:30分
- 講義内容はYoutubeライブでもご覧いただけます。Discordにリンクがあります
  - ライブコーディング VJのコードや音楽をなるべく綺麗に見せるためです！
- この資料やアーカイブ、ソースコードは共有しますのでご活用ください
- 講義中や講義終了後、何時でも質問を受け付けています。CommentScreenやDiscordなどで是非ご質問ください！

# 自己紹介

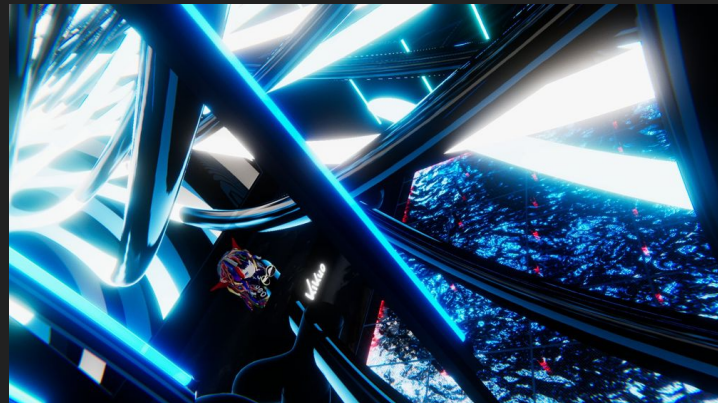
- Renardと申します
- バーチャルライブの演出や、  
Shaderを用いたVJなどをやっています
- 学士4年です



<https://scrapbox.io/RENARD>



[@Renard\\_VRC](https://twitter.com/Renard_VRC)



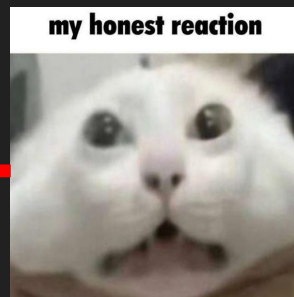
第一部

*+Livecoding VJ*

# ライブコーディングVJ？

VJ:DJの音楽に合わせて映像を出す

**GLSL Livecoding VJ:DJの音楽に合わせてリアルタイムに GLSLを書き、コードと共に映像を見せる**

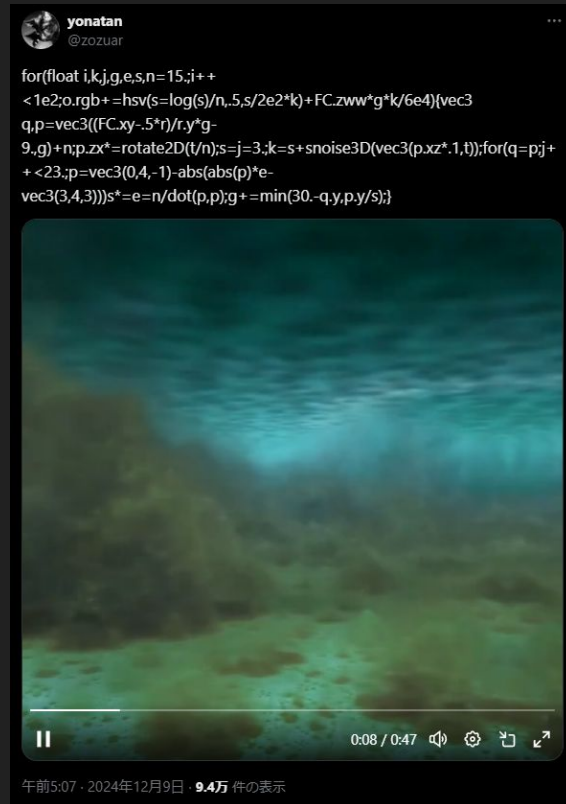


- 制限時間やコピペ無しに一からGLSLを書き、さらに音楽に合った映像を出すのは難しそうですが、幾つかのテクニックを抑えれば不可能ではありません
- この後にするライブコーディングはそんなテクニックが大量に出てくるので、気になった箇所は質問してください

# ライブコーディングの準備

## twiglをブラウザで開く！

- doxasさんが開発したライブコーディングエディタ
- つぶやきGLSLに使われたり、soundshader、broadcast機能もある
- ソースコードが公開されてるので、カスタム可能！
  - (今日は私が勝手に midi入力を使えるようにしたものを使います)



<https://x.com/zozuar/status/1865850795822104948>

実演

いかがでしたか？

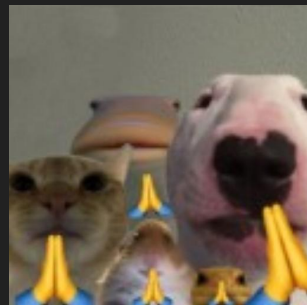


## 解説(割愛)

- 時間の都合上、解説は割愛します。すいません！
- 全てtwigl上で解説用のコードを書いているので、以下のリンクから手元で動かすことができます(twiglはコードの共有リンクが作れます)
- 分からないコードがあればいつでもご質問ください！

ライブコーディングのソースコード：

<https://twigl.app?ol=true&ss=-OGJOLmYRCg9cXLHSYRf>



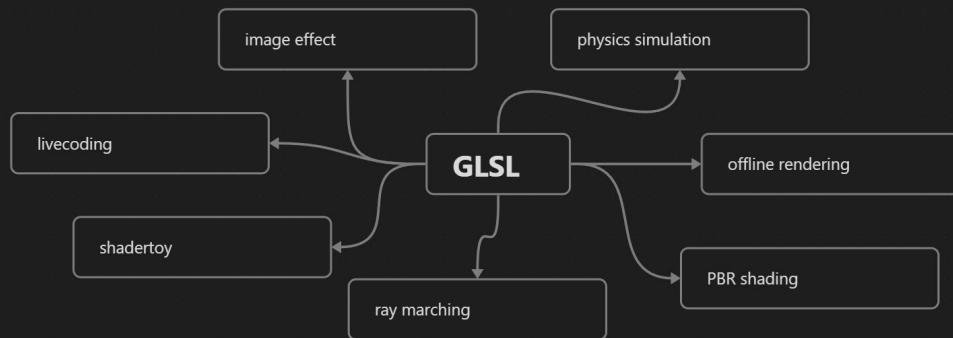
## 第二部

# コードから学ぶ GLSL

# GLSLの学習 どうやるのがいい？

- GLSLといっても様々なスキルがある
- 幸い今は各ドメインに対して多くの記事/サイトがあるので、尖ったことをしない限り学習には困らない (形式知)
  - 付録にドメイン別おすすめのリソースをまとめています
- 記事やサイトに無い、先程のライブコーディングのような暗黙知を学びたい場合は？

形式知: 明文化・形式化されており、共有や伝達が容易な知識  
暗黙知: 言語化や形式化が難しく、経験や直感に基づいた知識

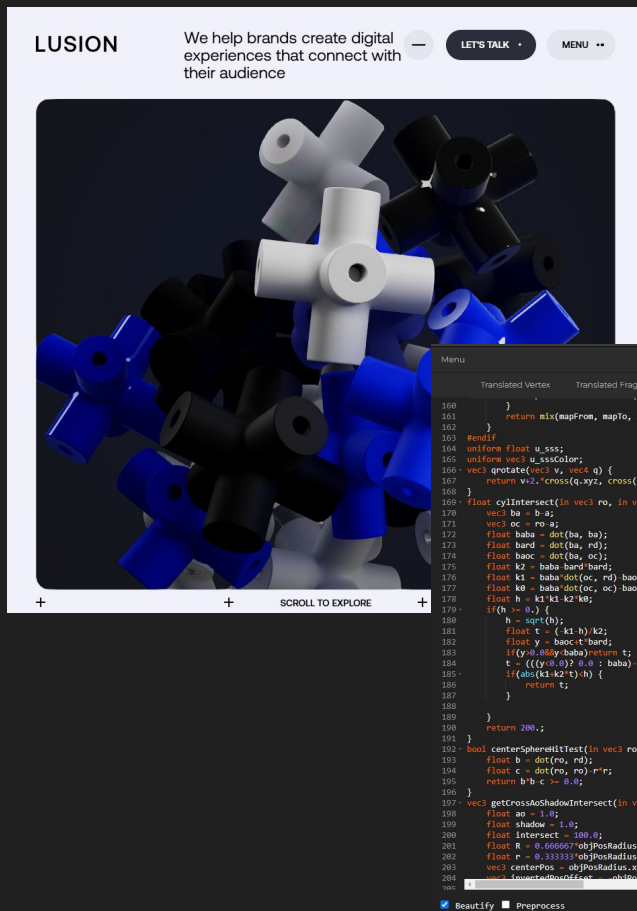


# GLSLの暗黙知

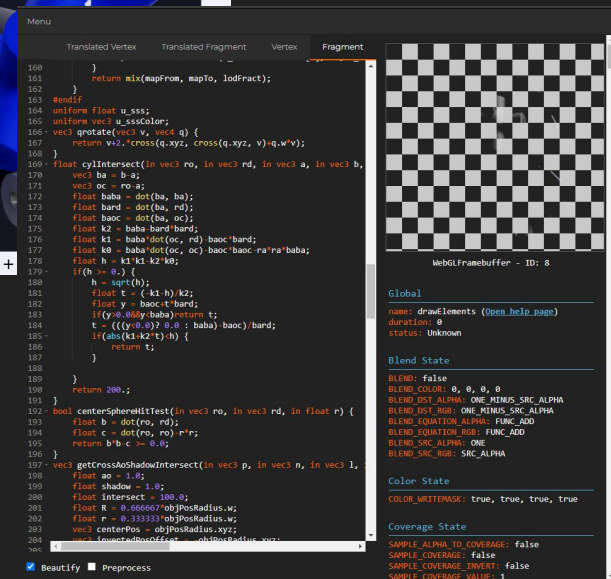
- GLSLの暗黙知とは
  - Shadertoy
  - いっぱい動くタイプのサイト
  - ライブコーディング
  - など...
- これらの明文化は少ないが、  
大抵ソースコードがある



コードから学べばいい！



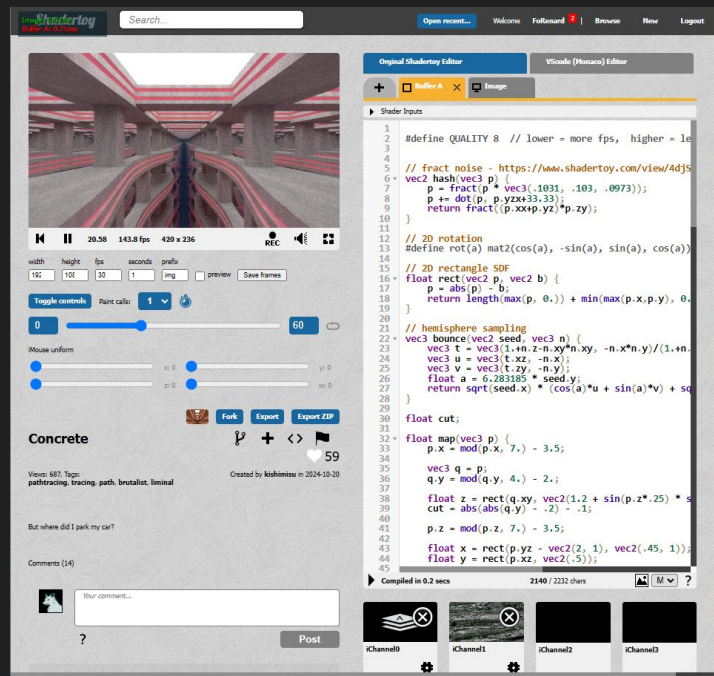
[Spector.is](https://www.spector.is/)を使うと、あらゆるサイトのシェーダーを覗き見ることが出来る



**ShaderToyの作品を読んでみよう**

# Shadertoyを読む - 1

- Shadertoyのシェーダーは以下の3種類
  - Image:最終的に画面に出力するパス
  - Buffer A/B/C/D: 中間処理を行えるパス
  - Common: Includeみたいなもの(出力はしない)
- 実行順序: Buffer A→B→C→D → Image
- 各パスごとにiChannel0~3で設定したバッファを使うことが出来る
  - 例) Buffer A, Imageの2パス  
ImageパスでiChannel0にBuffer Aを設定し、  
`texture(iChannel0,uv)`で利用できる



<https://www.shadertoy.com/view/l3SfDW>  
Concrete by kishimisu

# Shadertoyを読む - 2

- [Shadertoy unofficial plugin](#) を入れると、パスの情報含めソースコード全体をjson形式でexportできる
- ChatGPT(とか)に食べさせて、コメントを書いてもらったものを同じjsonで出力させる
- Shadertoyで新規にシェーダーを作成し、importする



以下はShadertoyのコードをjsonとしてエクスポートしたものです。GLSLコード内に分かりやすい日本語解説をコメントとして追記したものを、同じjson形式で出力してください。

```
json
{"ver": "0.1", "renderpass": [{"outputs": [{"channel":
```



7 秒の間、Thought about GLSLコード解説

json

コードをコピーする

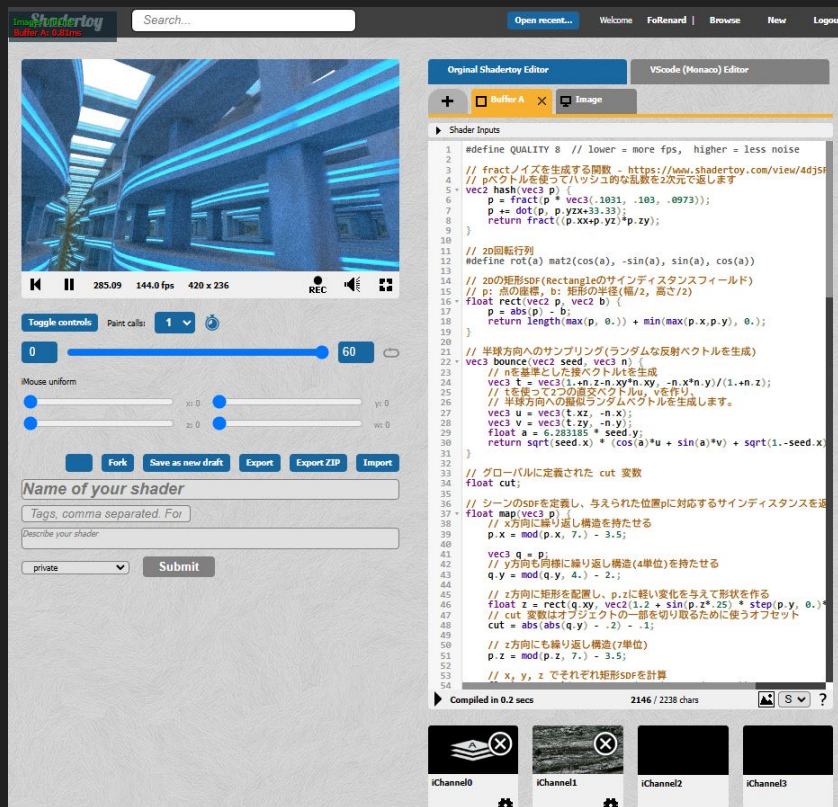
```
{
  "ver": "0.1",
  "renderpass": [
    {
      "outputs": [
        {
          "channel": 0,
          "id": "4dfGRr"
        }
      ],
      "inputs": [
        {
          "channel": 0,
          "type": "buffer",
          "id": "4dXGR8",
          "filepath": "/media/previz/buffer00.png",
```

<https://chatgpt.com/share/6781f13b-7db4-8010-a3e1-4db29371102b>

# Shadertoyを読む - 3

## 以上の手順を経ると、Shadertoy - ChatGPT間で相互に連携出来る！

- 先程のチャットに続けてコードについての質問を投げる
- 希望に沿った編集をしてもらう
- 等何でも出来るし、とても読みやすい





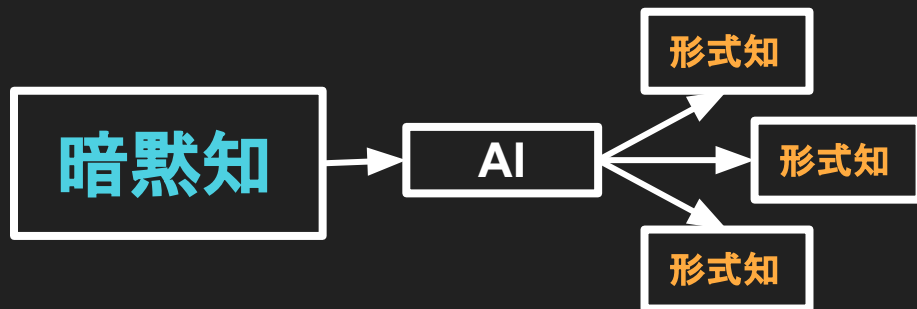
実演

# まとめ - GLSLを読む際の要点

- GLSLを読む際に衝突する壁は大きく二つ
  - コード全体の構造、流れの理解
  - コードブロック単位での理解
- コードの役割をAIに聞き、その際に知らない単語があれば検索
  - どんなに難しくとも、関連する単語は出してくれるはず
- コードの一部を変更→出力がどの様に変化したかを目で確認し、試行錯誤で学習する

これらはAIが得意とするタスク！

AIを活用して、暗黙知を細かい形式知にしてしまえばいい



ご清聴ありがとうございました！

&

質問コーナー

## 付録 - おすすめリソース: レイマーチング

- [iquilezles.org](http://iquilezles.org) - Inigo Quilez
- [wgld.org GLSL contents](http://wgld.org/GLSL%20contents) - doxas
- [球を出したあとのレイマーチング入門](#) - Butadiene
- [レイマーチング0から1](#) - 0b5vr
- [レイマーチング1から5](#) - 0b5vr
- [SDF for raymarching \(距離関数のスキル\)](#) - gaz

## 付録 - おすすめリソース: Realtime Rendering

- [Physically Based Rendering in Filament](#)
  - Filamentというgoogleが開発したリアルタイム物理ベース レンダリング エンジンの解説
- [リアルタイムレンダリング 第4版](#)

## 付録 - おすすめリソース: Offline Rendering

- [Ray Tracing in One Weekend](#)
- [FADIS PRESS](#) - fadis
- [memoRANDOM](#) - shocker
- [MochiMochi3D](#) - kinakomoti

## 付録 - おすすめリソース:その他

- [The Book of Shaders](#)
- [Sketching with Math and Quasi Physics](#) - kynd.info