

# 電腦世界



機関紙第6号

# EDPS機関紙 第6号: 2025年開発アプリ紹介 - 第一弾!

EDPS(電子計算組織研究会)の機関紙第6号へようこそ。今回は技術系ブログっぽく、ラフにいこうぜ。俺たちEDPSは2025年もガンガンアプリ開発しまくったんだけど、その中からピックアップして勢いでいくつか紹介するぜ。全部自慢の作品だよ。技術的な話もぶち込んでるから、コード片手に読んでくれ。開発の裏話やTipsも混せて、参考にどうぞ！

## ① Lincorde: LineとDiscordのトーク連携マジック



まず最初に、LineとDiscordのトークをシームレスに連携させるアプリ「Lincorde」。普通、違うSNSだから連携なんて無理じゃん？って思うけど、BOT経由で実現した。Lineからメッセージ来たらDiscordに飛ばして、逆もしかり。グループチャットとかで便利すぎる。

サーバーサイドはPythonでサクッと組んだ。Flask使ってWebhook受け取って、LineBotApiとDiscordのライブラリで連携。コードのスニペットはこんな感じ：

```
python
from linebot import LineBotApi, WebhookHandler
```

LineのWebhookでメッセージキヤッチしてDiscordのWebhookにPOSTしたり。threadingで非同期処理してスムーズに。実装でハマったのは署名検証のInvalidSignatureErrorだけど、環境変数で秘密鍵管理すればOK。



## ② Zero: ストレージ無限大のAndroid撮影アプリ

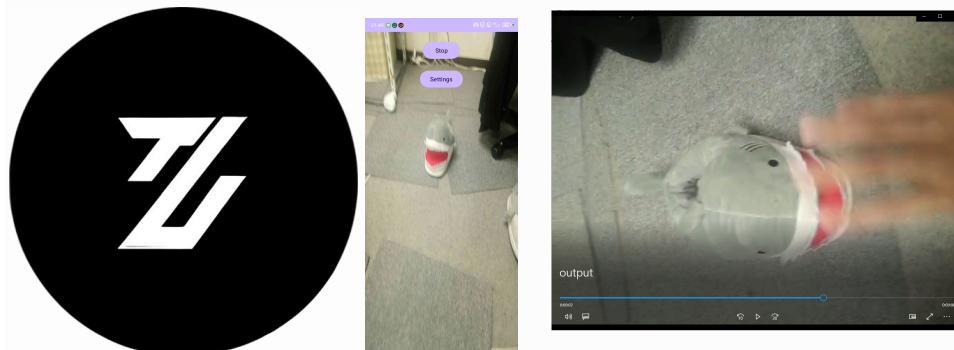
次はAndroidアプリ「Zero」。スマホのストレージ気にせず動画や画像撮りまくれるヤツ。撮影したら即サーバーにアップロードして、本体は空っぽ。旅行とかで神アプリだろ？

フロントはAndroid StudioでJava使って開発。サーバーサイドはPythonで、ffmpegで動画圧縮して送信。コードのエッセンスは、カメラキャプチャしてHTTP POSTで飛ばす部分。

今後の参考点：

- エミュレータの限界: Camera1とCamera2のエミュ実装が不安定。次は実機テスト優先で！
- Camera APIの選択: Camera1はシンプルだけど非推奨。Camera2は複雑だけど最新対応。用途で使い分けろ。
- サーバーURL: ローカルなら<http://192.168.2.58:8080/upload>、リモートならNgrokの<https://b037-153-246-178-81.ngrok-free.app/upload>みたいに。
- デバッグ効率化: LogcatにSurface状態のログ追加すると、次回のトラブルシューティングが楽チン。

これでストレージの呪縛から解放だぜ。Javaのカメラハンドリングはハマリやすいから、公式ドキュメント読み込んでね。



## ③ Locky: 右クリックで即暗号化！ 秘密ファイルの隠れ家

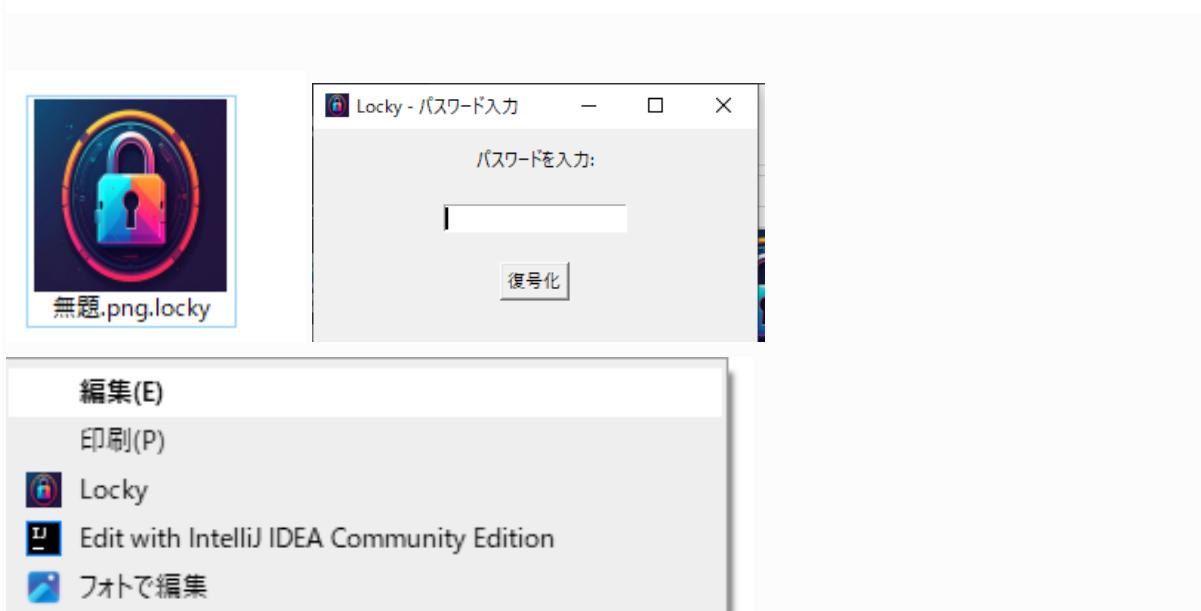
見られたくないファイルやフォルダをサクッと隠したい？ そんな時は「Locky」。エクスプローラーで右クリック → 「Locky」を選ぶだけでAES暗号化してロック。パスワード入力で復元できるシンプル設計。

公式サイトからインストーラーDLしてインストール → 再起動で自動でコンテキストメニューに追加される。内部的にはPythonスクリプトをPyInstallerでexe化して、Windowsシェル拡張で右クリックメニューにフック。暗号化は[cryptography](#)ライブラリを使ってガチガチに。

```
python
from cryptography.fernet import Fernet
```

パスワードはPBKDF2でキー導出して安全に。フォルダ丸ごとも再帰的に処理。復元もワンクリック。プライバシー守りたい人必須のツールだぜ！

install_locky.exe	2025/09/10 6:34	アプリケーション	13,293 KB
インストーラーは管理者権限で実行してください....	2025/09/10 6:35	テキスト ドキュメント	0 KB



## ④ 料理レシピ作成アプリ: 食材入力でGeminiが魔法をかける

シンプルに、食材リスト入力したらGemini APIがレシピ提案してくれるアプリ。サーバーPython、フロントHTML/CSS/JS。API叩いてJSONパースして表示。

Geminiの自然言語処理が優秀で、「にんじん、鶏肉、玉ねぎ」って入れるとカレーとか提案。フロントはJSで非同期リクエストしてUI更新。PythonのrequestsでAPIコール。手軽に作れて、キッチンで重宝するわ。

## ⑤ LaTeX互換アプリ: MarkdownとLaTeXのハイブリッド

LaTeXの数式とMarkdown混せて使えるアプリ。LaTeXコードは公式/数式に変換、それ以外はそのまま。フォントサイズとか反映。ChatGPTやGrokの出力そのまま貼ってPDF化できるのが神。

Pythonで開発してpyinstallerでexe化。Windows/Mac/Linuxどこでも動く。授業ノートをAIでまとめて入力→PDF出力、時短すぎる。pandocやtexlive依存だけど、ローカルで完結。



## ④ アンペール-マクスウェルの法則（電流と変位電流）

【式】

$$\nabla \times \vec{B} = \mu_0 \vec{J} + \mu_0 \varepsilon_0 \frac{\partial \vec{E}}{\partial t}$$

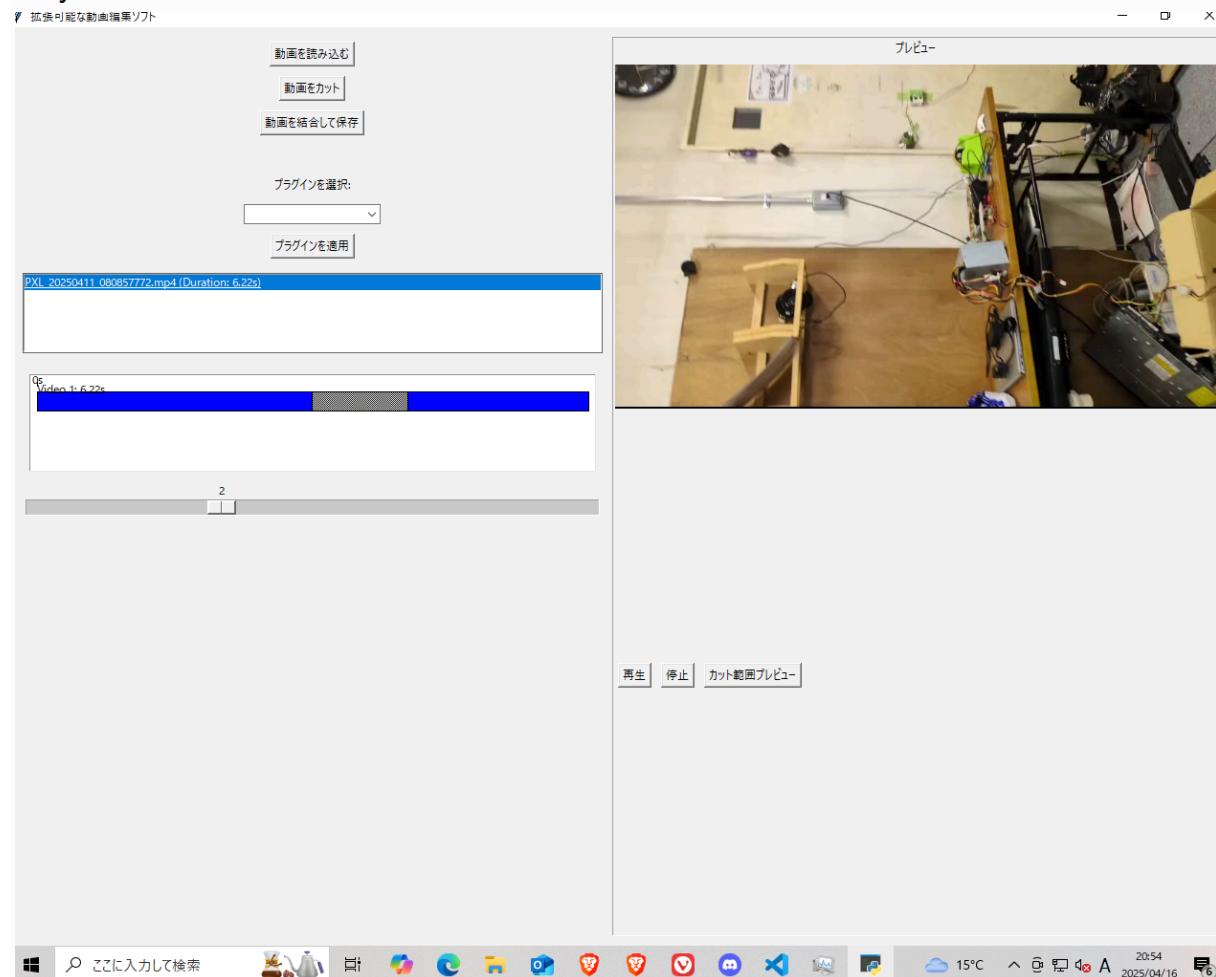
## ⑥ でんでんオーダー：スマホ注文システム

電子技術研究部との合同プロジェクト。Flutterでスマホアプリ作って、注文システム。レストランとかでQRスキャンして注文、厨房に通知。Flutterのクロスプラットフォームが便利。DartでUI組んで、Pythonでバックエンド連携。

## ⑦ 新入生の動画編集ソフト: 初心者が1ヶ月で完成!

部長の指導のもと、プログラミング未経験の1年生が入部して1ヶ月でPython動画編集ソフト作った。exe化で誰でも使える。

tkinterでGUI、moviepyで編集処理。クリップカット、テキストオーバーレイ、出力。初心者でもPythonのライブラリパワーでここまでいける証明だぜ。みんなもチャレンジ!

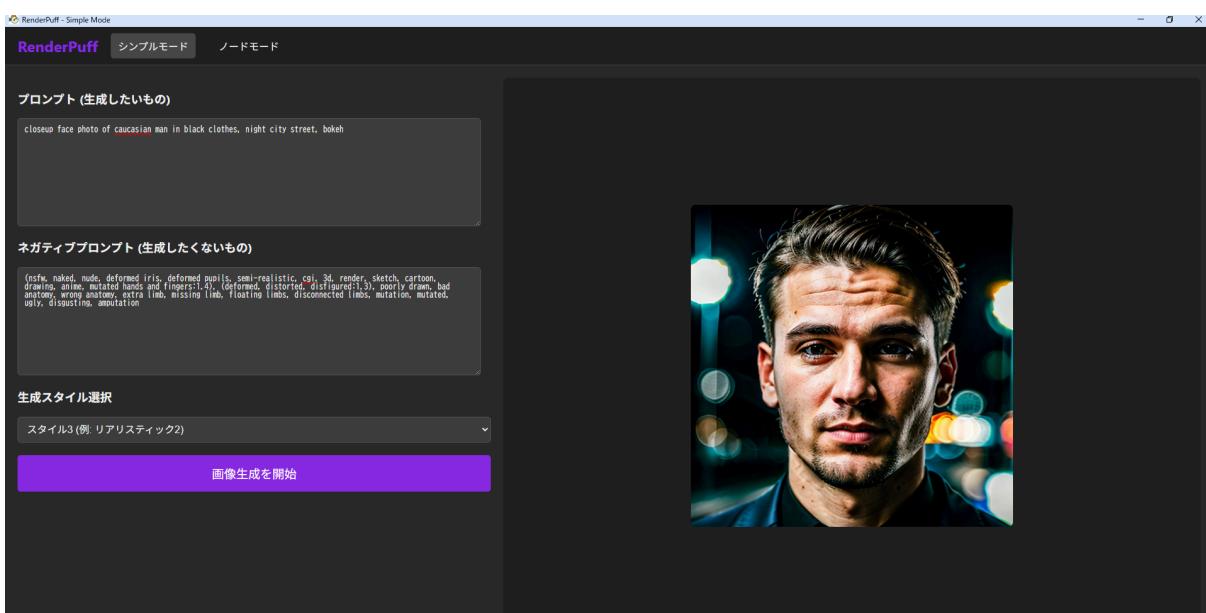
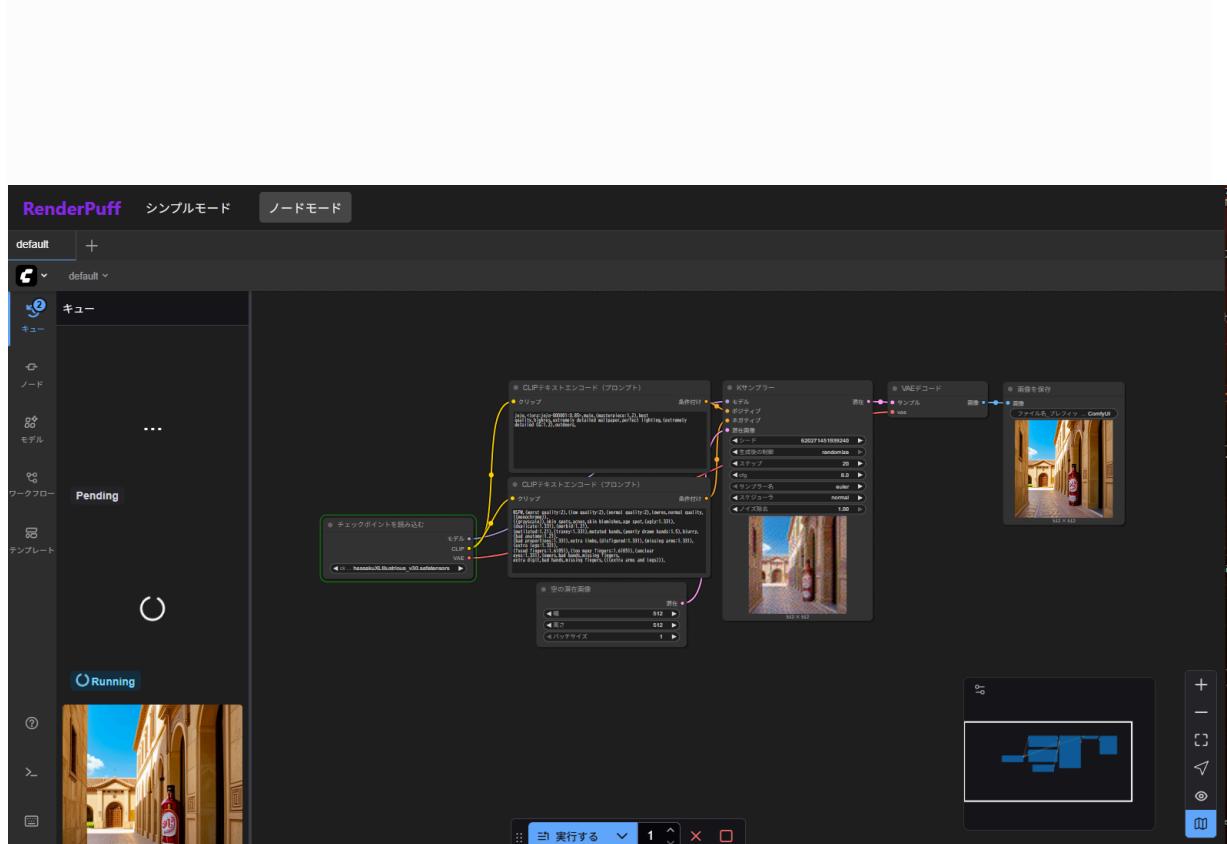


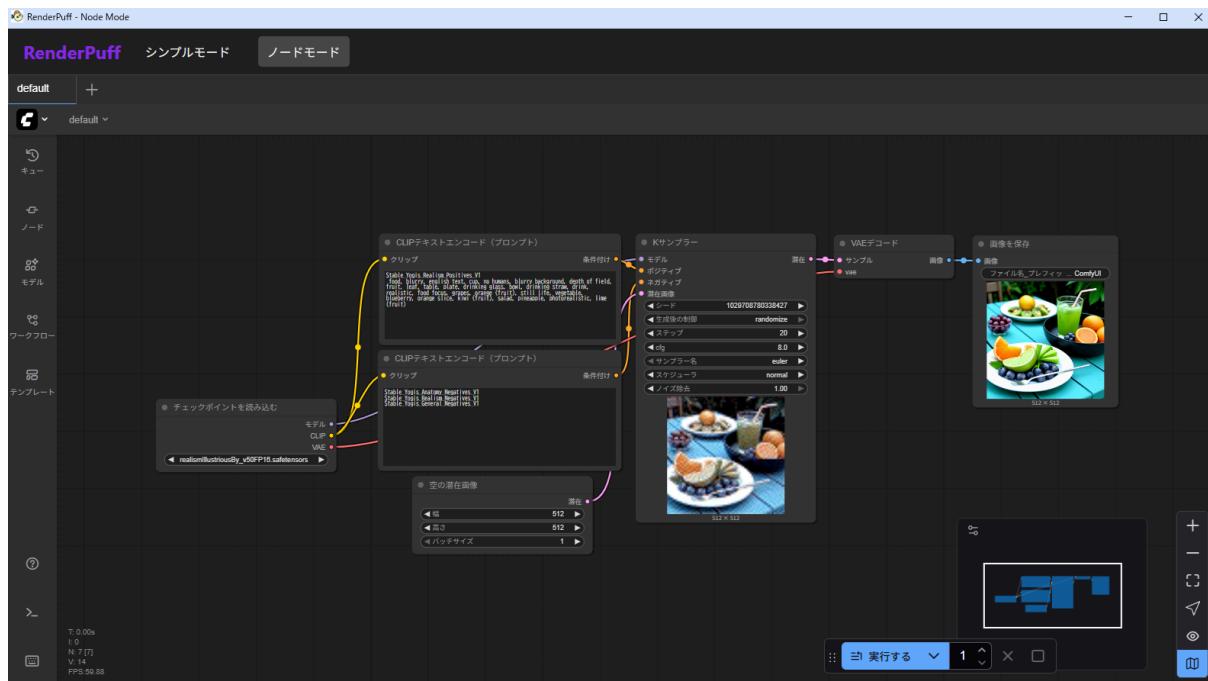
## ⑧ RenderPuff: ローカルAIで画像/動画生成

最後に、ネット不要のローカルAI「RenderPuff」。自分のPC上で画像/動画生成。シンプルモードはプロンプト入力でOK、ノードモードはComfyUIベースでカスタム。

アニメ風やリアル、3Dオブジェクトまで。Stable Diffusionモデルローカルで動かす。GPUあれば爆速。サーバー依存なしでプライバシー守れるのが売り。







これで第一回の紹介終わり！ EDPSの2025年はアプリ開発ラッシュだったぜ。次号で続きやるかも？ 質問やフィードバックあったらSNSか掲示板で投げてくれ！技術好きの皆さん、インスピライアされて何か作ってみて！ 

## -発行理念-

部の活動を通じて得た多くの知識を、もっとたくさんの人々に知ってもらいたい。

そして、ソフトウェアやハードウェアをはじめとする私たちの活動分野に、

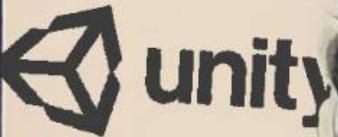
多くの人が興味を持つきっかけになってほしい。

また、すでにその分野を知っている人にとっても、さらに深く踏み込むための手がかりとなることを願っている。

本誌が様々な分野の記事を掲載するのは、幅広い知識を得て初めて見えてくるものがあると信じているからである。

このような理念のもと、私たちは機関誌『電腦世界』を発行している。

bj.desc)  
 .perf(Pc=1.1, eps=2.3, MR=1.45);  
 c, Cstar, Tc, MW, gamma = ceaObj.get\_ivacCstrTc\_ChmmWGM  
 (%.5.1f %.8.1f %.6.1f %.8.1f %.8.1f %.8.1f %.8.4f %.1f)  
 (Pc, eps, MR, ispVac, Cstar, Tc, MW, gamma)  
 (MPa) AreaRatio O/F ispVac(sec) Cstar'  
 in np.arange(1, 2, 0.1):  
 w\_perf(Pc=1.1, eps=2.3, MR=MR)  
 load = 30.0 # [kg]  
 = 350.0 # [kg]  
 propellant = 800.0 # [kg]  
 = 0.90 # isp efficiency  
 al = Wstg + Wpropellant + Wpayload  
 al = Winit - Wpropellant  
 show\_deltaV(Pc=1.1, eps=2.3, MR=1.45);  
 ispVacTheoretical = ceaObj.get\_isp(Pc=Pc,  
 ispVacMeasured = effisp \* ispVacTheoretical  
 ispSL, mode = ceaObj.estimate\_Ambient  
 ispSLMeasured = effisp \* ispSL  
 deltaV = 9.80665 \* ispVacMeasured \* np.  
 print(%.5.1f %.8.1f %.7.2f %.8.1f %.8.1f  
 # print(mode)  
 print()  
 print('Pc(MPa) AreaRatio O/F ispVac(s) isp  
 for eps in np.arange(1, 4, 0.5):  
 show\_deltaV(Pc=1.1, eps=eps)



# I WANT YOU FOR EDPS

Room 201, Building O, Neyagawa Campus

