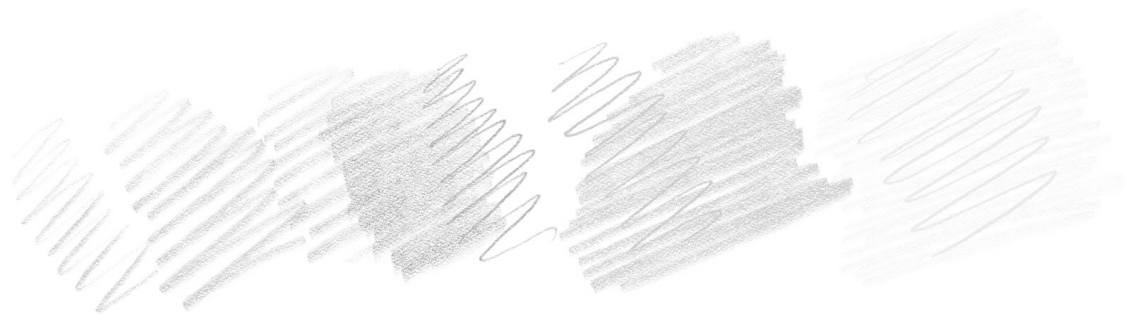


お絵描きチャットを Wasmで作った話

多人数同時編集対応ペイントツールとしてのお話とか



自己紹介

- ・Windows用のペイントツールや、Webブラウザで動くお絵描きアプリ「8bitpaint-web」を個人開発してました。
- ・現在はiPad向けのお絵描きアプリを開発中。
- ・メインの開発言語はc/c++

twitter : @minordaimyo

web : <https://minordaimyo.net/>

えんぴつチャットのURL : <https://minordaimyo.net/pencilchat/>

お絵描きチャットって何？ 1/2

- ・一つのキャンバスに多人数で同時にお絵描きして遊ぶサービス
- ・00年代くらいのインターネットで流行
- ・最近有名なものだと「MagicalDraw」 <https://draw.kuku.lu/>

お絵描きチャットって何？ 2/2

普通のお絵描きソフトと違うところ

- ・サーバーを立ててその上で動作させる

(一般的なペイントツールは個々の端末だけで動作する)

- ・複数人から描画操作が同時に発生したりする

→サーバーで一元管理しないと大変

「えんぴつチャット」の仕様 1/2

- ・キャンバスサイズ: 9921x7016pixel(A3 600dpi相等のビットマップ)
- ・色数: 256色
- ・レイヤー機能は無し
- ・ApplePencil等スタイラスペンの筆圧と傾き・方向検知に対応
- ・参加者は最大で同時に8人まで
- ・ROM参加(描画はできず、見るだけ)は40人まで
- ・Undo/Redo可能 ← **ここ大事!**

「えんぴつチャット」の仕様 2/2

- ・クライアントサイドは素のJavaScript+WebAssembly(主にc言語)
- ・サーバーサイドはNode.js+WebAssembly(コードはクライアントと共通)
- ・WebAssemblyへのコンパイルにはemscriptenを使用
- ・クライアント・サーバー間の通信にはWebSocketを使用
- ・サーバー側の消費メモリは、チャットルーム1部屋あたり約250MB

多人数での描画とUndoの挙動 1/3

・描画領域が重なっていない場合、各人とも自由にUndo可能



ユーザー1

ユーザー2

12345

ABCDE

1. ユーザー1が”12345”と描画
2. ユーザー2が”ABCDE”と描画

この場合は、ユーザー1、2共に自由にUndoできる。

多人数での描画とUndoの挙動 2/3

・描画領域が重なっている場合、後から描いた人がUndoすれば、先に描いた人もUndoすることが可能



ユーザー1
ユーザー2

1 A 3 4 5
B
C
D

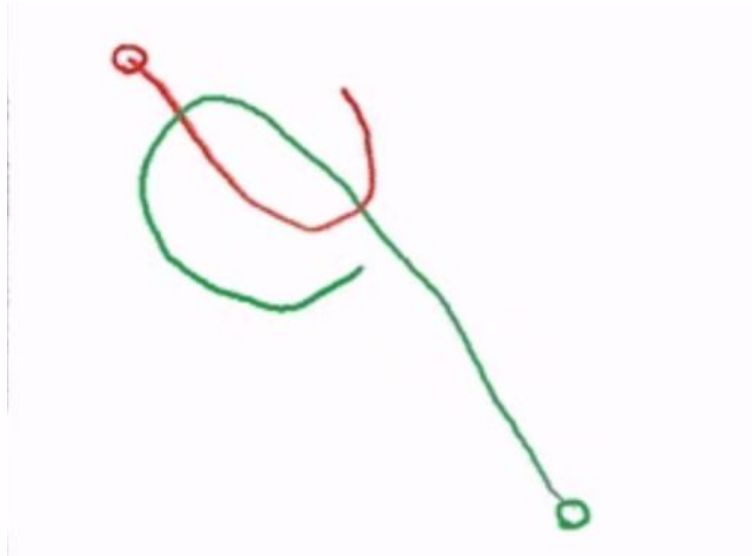
1. ユーザー1が”12345”と描画
2. ユーザー2が”ABCD”と描画

この場合ユーザー1は’5’,’4’,’3’まではUndo可能。

ユーザー2が’D’,’C’,’B’,’A’とUndoした後なら、ユーザー1は’2’,’1’もUndo可能になる。

多人数での描画とUndoの挙動 3/3

・複数人による描画が、空間的・時間的に同時に行われた場合、Undoはできない



1. ユーザー1とユーザー2が同時に描画を開始し、且つ同じ領域に重なるように線を描く

この場合は、ユーザー1、2共にUndo不可能になる。

描画レスポンス 1/4

描画処理の流れ

- 1.クライアントからサーバーにペンの筆跡データを送信
- 2.サーバーが各クライアントから受け取ったペンの筆跡データに対して、受信した順に描画処理番号を割り振る。(描画順序を確定)
- 3.描画順序番号を付加したペンストロークデータを、全クライアントに送信する。
- 4.各クライアントで順序通りに描画処理を行う。

描画レスポンス 2/4

問題点

- ・ペンを走らせてから実際に線が描画されるまでに、データがサーバーとの間で往復するのを待つことになる。
- ・その為、ペンの動きに対して描線が遅れて着いて来る事になる。

描画レスポンス 3/4

解決策

自身の描線だけは、サーバーからの返答を待たずに投機的に描画してしまう

描画レスポンス 4/4

解決策

自身の描線だけは、サーバーからの返答を待たずに投機的に描画してしまう

・サーバーからの返答を受け、キャンバスの内容に矛盾が生じた場合は、投機的に行った描画をキャンセルする。

矛盾の例)

自身が投機的に行った描画エリア内に、他のユーザーがわずかに先行して別の描画が行われていた事が、後から分かった場合など

公開した後

- ・2021年頃から一般に公開し、サービス開始
- ・当初は1日数人から多くて10人くらいの利用だった。
- ・2022年4月にTwitterでバズる(1.4万RT , 2.5万いいね)
 - 収容できる人数を遥かに超えるアクセスで、数日間常に満室状態が続いた
 - バグも山ほど発見された

なぜWasmなのか？

- ・以前にWindows用のペイントツールをc言語で個人開発しており、そのソースコードを流用できたから。

Wasmのメリット

- ・高速動作で省メモリ
- ・libpngやzlib等のライブラリをリンクした上でも尚バイナリサイズが小さい。(約560kB)

Wasmのデメリット

・デバッグが辛い

Chromeのデベロッパーツールで関数呼び出しの履歴をチェックしたり出来るが、printf()頼りになる事も多々.....

おわり

ご静聴ありがとうございました！