

i1i2i3実験



第28班A

03-160441 土屋潤一郎

03-160471 井上友貴

←昨夜の発表者二名の様子(イメージ)

もくひょう

- ・最低限、通話したい
- ・電話をしながら、ファイルの送受信、チャットをしたい!
- ・もちろん、色々なことが簡単便利に出来たほうがいい

BUT

- ・時間は有限
- ・開発者の精神的な余裕も有限
- ・開発者の能力も有限
- ・目指せホワイト開発現場

このような感じになるのは避けよう
(結局こうなったんだが)



優先順位を決めよう！

開発戦略マトリクス

		難易度		難
		易	難	
重要度	高い	必ず作れる要素	おそらく作れる要素	作ってみたいと不明
	必要な要素	<ul style="list-style-type: none">・優先度中★★★・迷わず作る・他に影響を与えないのならば、むしろ後回しでもOK	<ul style="list-style-type: none">・優先度高★★★★・できるだけ早期に検証・実験する	<ul style="list-style-type: none">・優先度最高★★★★★・最優先で検証・実験し、バックアッププランを用意
	できれば欲しい要素	<ul style="list-style-type: none">・優先度低★★	<ul style="list-style-type: none">・優先度中★★★・優先的に検証・実験するかどうかを早期に決める・優先度を下げる場合は切り離し可能な状態を維持してゆとりの人員と時間で挑戦	<ul style="list-style-type: none">・優先度低★★・なるべく作らない・作る場合は切り離し可能な状態を維持してゆとりの人員と時間で挑戦
低い	無くても困らない要素	<ul style="list-style-type: none">・優先度最低★・人員や時間にとっても余裕があったら後で作る	<ul style="list-style-type: none">・優先度最低★・なるべく作らない・とても余裕があったら挑戦してみても良い	<ul style="list-style-type: none">・優先度×・作ってはならない

※あくまで概念の紹介として記載していますので各方針や優先度の記述は例として捉えてください
※いろいろな事情や戦略によっては優先度は変化します

そもそも電話とは…

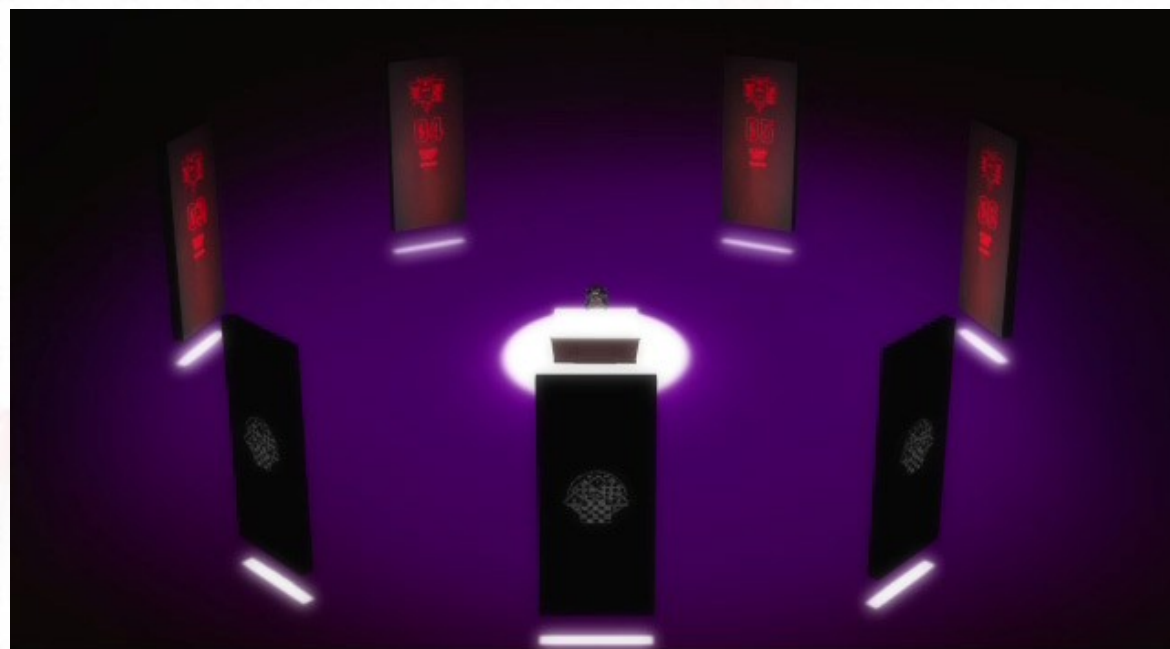
- 自分の”声”をそのまま伝えることが重要なのではない
音楽再生とは違う
- 必要なのは相手方のクリアな声ではないのでは？
話した内容が通話相手にある程度欠損無く伝われば良い
- どうも音質の追求は二の次で良さそう……
我々にとっては、遅延の排除、エコーの除去辺りが必要かつできるかどうか作ってみたいと不明な要素
→優先度最高
音質の追求とか多者間通話はできるかどうか作ってみたいと不明でできれば欲しい要素
→優先度低

できたこと（やれたこと）

- OpenAL, pthread, popen(, system)を使用した基本的な音声通話の実装
- エコーサプレッサもどきの実装
- 通話しながらのチャット、ファイル送信
→嘘。チャットはできてなかった
- CUIだけど、CUIなりに操作をわかりやすく

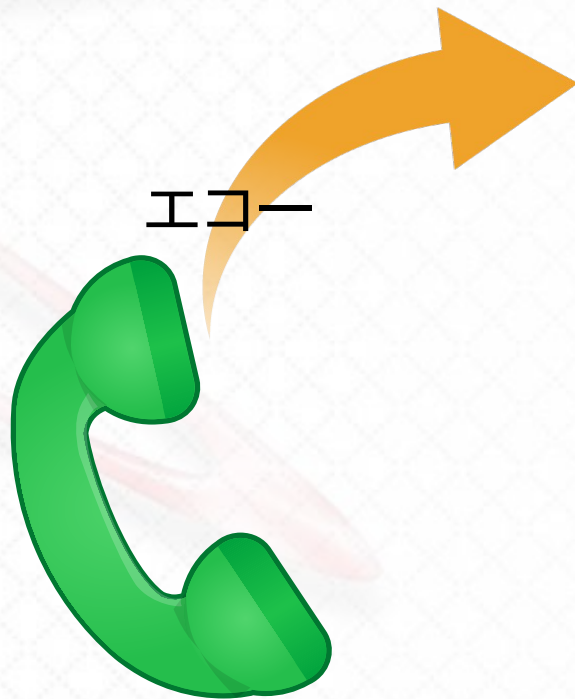
OpenAL #とは

- フリーの立体音響ライブラリ
- 録音・ストリーミング再生が可能
だったはずなんだけど.....
- もし三次元空間内での音作りを、多者間通話に利用
して面白いことができればなあ（反実仮想）
儚い夢と消えました（報告）



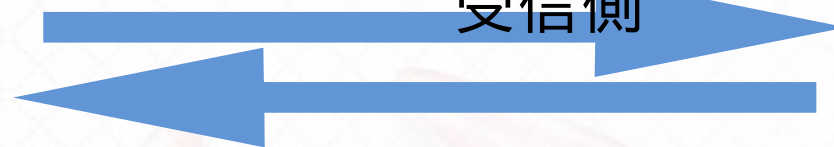
立体音響を用いた多者間通話（イメージ）

エコー
自分が話したエコーが聞こえる



送信側

受信側



エコーサプレッサ（として働くなにか）

- （本来は）受信データ、録音データを比較し、音が大きいほうを通過させるもの
- 今回は閾値以上の振幅のデータを含まないデータセットを送信しないことで擬似的に実装
- つまるところ滅茶苦茶原始的な無音判定をしているにすぎない
 - 閾値を大きめに設定することで音響エコーを無音判定しちゃおうというだけの話
 - 両者のマイクが一斉に有効な音声を捉えるとエコーが入ってしまう
 - それを解決する適応フィルタを用いたエコーキャンセラの実装は発表者の数学力がゴミクズ同然だったのでAbort

つかきのう

電話しながら、ファイル送受信とかもしたい

→pthreadで並列処理しつつ、popenやsystem
で別プロセスも立てる

→結果グローバル変数が多いし、スパゲティ
コードになった（と思う）

スパゲッティコードの神様（大嘘）



追加機能

- ファイル送受信

systemでncコマンドを開く

この演習の本義からは一番外れた実装方法

「多分これが一番簡単だと思います」

- チャット

```
$ system( "gnome-terminal --tab -e './chat  
<server/client>' " )
```

でチャット用のウィンドウを開きチャットする

- gnome-terminal使ってる人でないと使えないプログラムになってしまった
- 昨日の発表見てたらターミナルを使用するライブラリ（？）、あるんですね
 - ときどきググってたんだけど探し方が悪かったらしい

今後の課題（その１）

- ストリーミング再生をスレッドで処理できない！
 - 事前に、テスト用に標準入力のPCMデータを受け取って（OpenALで）ストリーミング再生するプログラムを書いたところ、エラー吐かずに動くところまで漕ぎ着けた
 - それをほとんどそのまま移植したところ.....
 - スレッド内ではリングバッファ的なものがきちんと働いてくれない
 - 原因不明
 - のでテスト用プログラムをpopenで開くことに
 - 「それってplayをpopenするのと一緒にだね」
 - と言われればそれまでだが、再生機構の中身がプログラマブルなので将来の拡張性は一応確保されてる（？）

今後の課題（その2）

- 電話をかけ直せない！
 - ソケットを閉じててもサーバーのポート番号が占有されたまま
 - なんかパケットが飛んでくるかもしれないので（教科書情報）
 - 一回電話が終わったらプログラム自体を閉じるように

まとめ

- 締め切りがあるときに、周りに聞ける人がいないものを新しく始めようとしない (OpenAL)
 - 公式ドキュメントは神なのでみんなも締め切りのないときに学ぼう
- ふわっとでも優先順位を付けるのはだいじ
 - つよいひとはどんどん実装できるけど我々はそんなことはできない
 - 結果として未実装だけだろうと思えばちょっとした改変だけで搭載できそうな機能が積み残されることにはなる
 - 弱者はそれで良いのでは
- 数学力は、だいじ