

Alice_Maker の構造

各用語解説

BP →blueprintのこと。裏はC++でできている。スクリプト、ビジュアル言語、オブジェクト指向。VM上で動作し、コンパルをするとblueprint bytecodeを吐く。windows,mac,ios,android,linux,Nintendoswitch等様々なos上で動かせる。C++でできることは基本出来る。できないものや外部のライブラリ等も自分でBPL関数を書くことで実装できる。C++と互換性がある。速度は問題にならないがループ系の処理だとC++の10分の1、pythonの10倍ほどの速度になる。

Uobject→unreal engine上のすべての基盤クラス。ここから派生クラスができています。これを継承したクラスはエンジン側が全てGC(ガベコレ)を自動管理する。これによりNullPointerExceptionの発生を防いだりIsValidの安全性を担保したりしている。

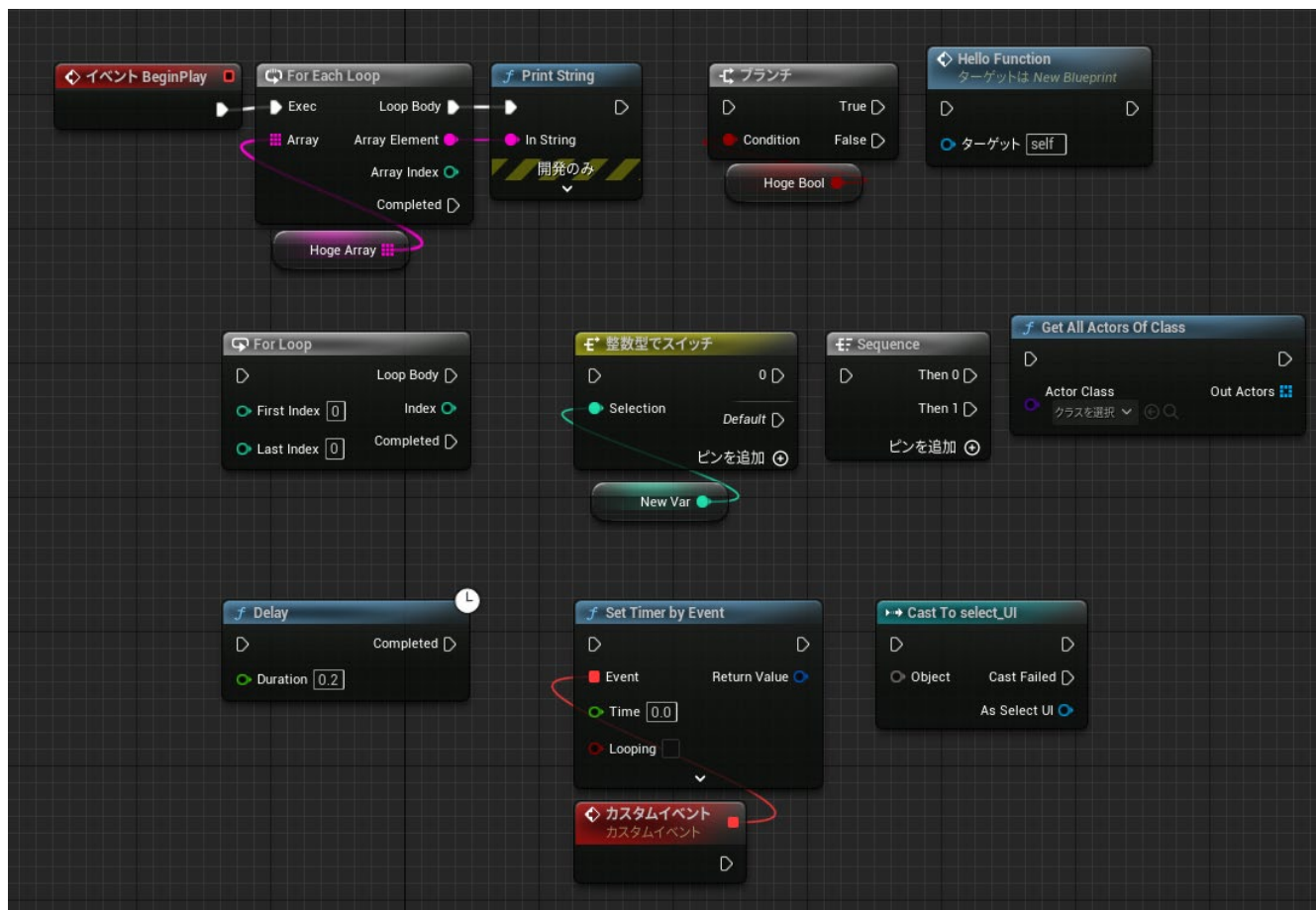
Widget→UIのこと。基盤クラスはC++でできておりすべての処理をBPで書ける。

レベル→マップ空間のこと。マリオで言えばステージ 1-1みたいなもの。

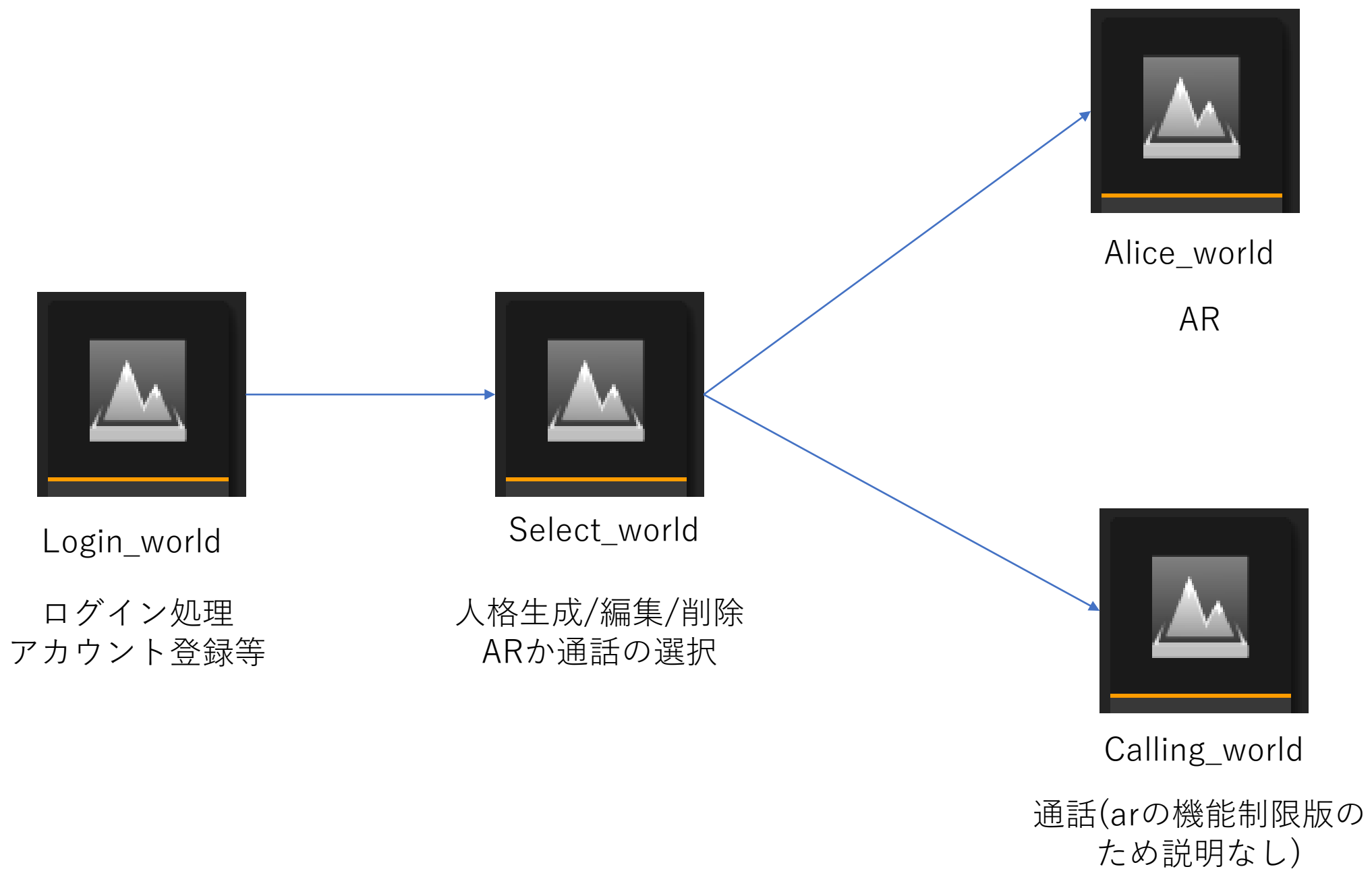
Actor→基盤クラスはC++でできており、派生クラスにcharacterやpawnなど様々ある。プレイヤーやNPC、敵など何らかのロジックを持ち合わせるものでレベル上に実体を持たせるものはこのクラスを使用する。

GameInstance→アプリ上どの場面においても一つだけ必ず存在するクラス。設定で事前に決まっている。どこから、どのクラスからでも参照ができ、起動時から終了時まで存在する为一時的な変数の保持などに使われる。

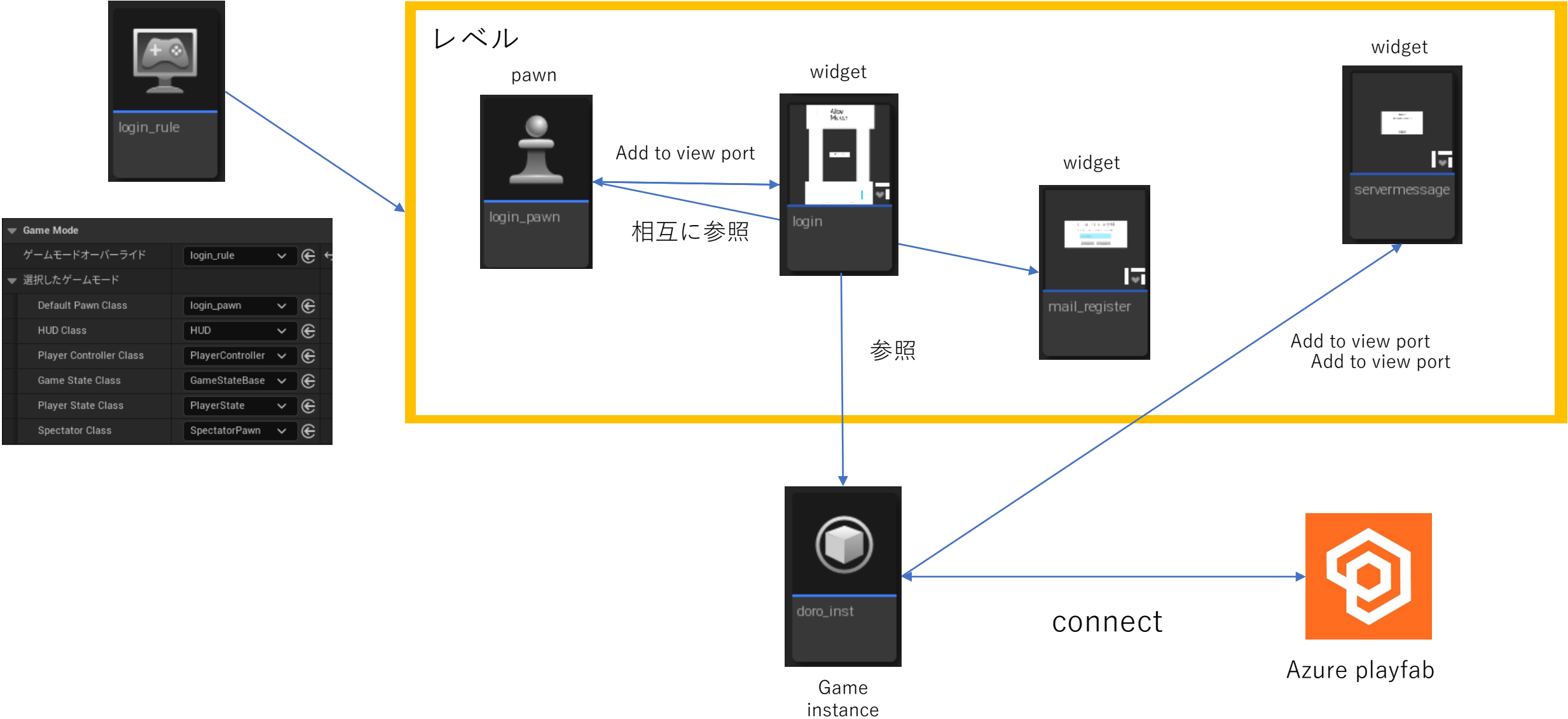
BPノードと変数の例。基本的なものはあるがC++で自分で拡張することも可能。関数、構造体、マクロ、インターフェース、castも存在する。
ノードを繋いでいき各処理を作る。
Delayなどエンジン側の独自のものも存在する。



The screenshot shows the Unity Inspector's 'Variables' (変数) section. The variable name is 'hoge bool' and its type is 'Boolean'. A search bar is active, showing a list of available types including Boolean, Byte, Integer, Integer64, Float, Name, String, Text, Vector, Rotator, and Transform. The search results are filtered to show 'Boolean'.



Login_world



Login_worldの大まかな流れ

起動後

- 1 gameinstanceで起動直後にplayfabへ接続
- 1 widget(login)をメモリ上に展開(create)し表示(add to view port)
- 2 タイトルデータを取得しサーバーコードが200ではない場合メッセージを取得し表示

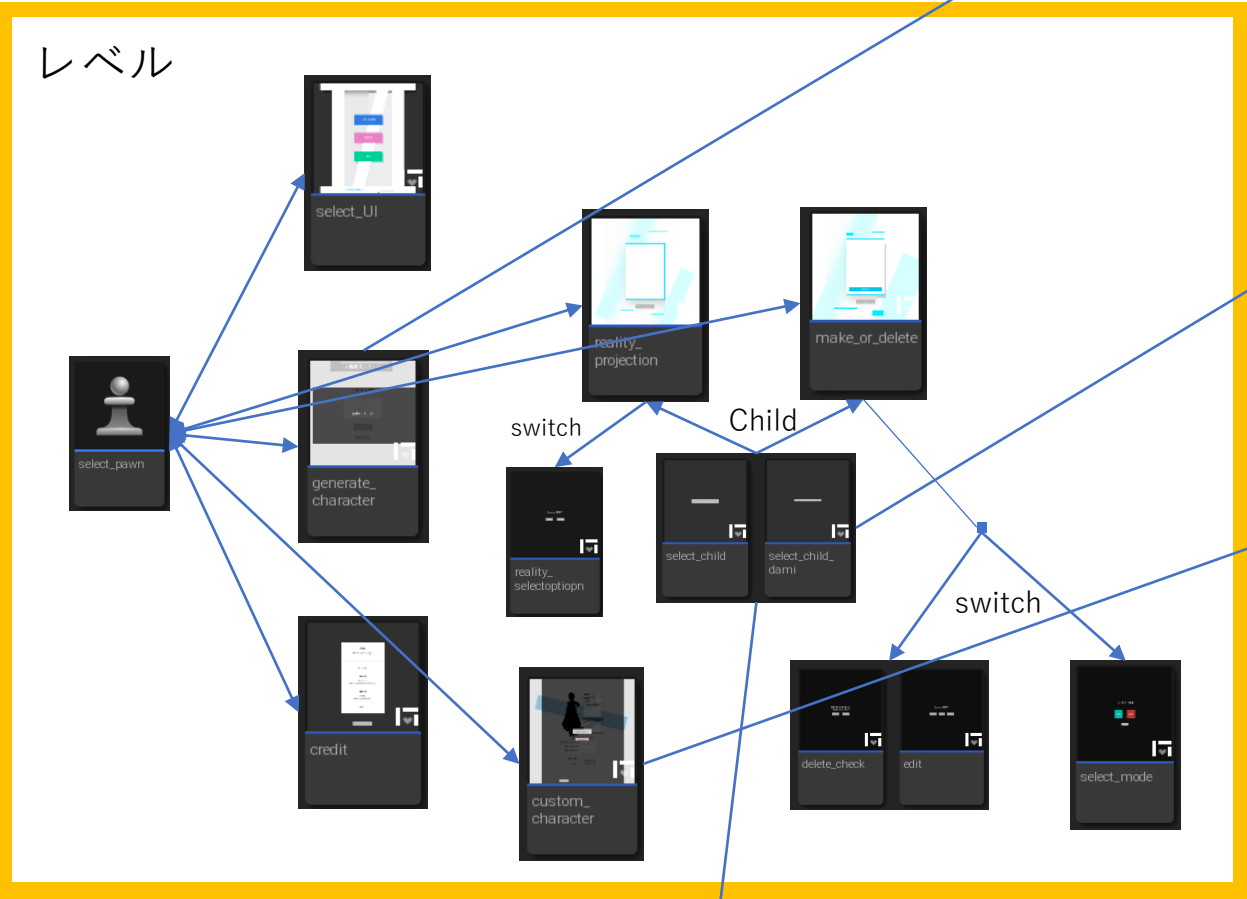
ログインボタンクリック後

- 1 instanceのサーバーコネクトのフラグをチェック後deviceIdでログインする
- 2 ログイン後 “ShowContactEmailAddresses=True”を含むjsonを構築、リクエストしget array field=ContactEmailAddressesでメアドの配列を取得。
- 3 一つ以上ある場合要素の最初を取得しget filed=VerificationStatusで戻り値がConfirmedの場合select_worldへ遷移。
無い場合はwidget(mail_register)を展開、表示

メアド登録ボタンクリック後

- 1 テキストボックスの引数を元に認証メールリクエストを送信

Select_world



Json index
(string array)

json

```
1 <スローで選択されています>
2
3 "name": "桜井_こがね",
4 "gender": "女",
5 "age": "19",
6 "appearance": "髪型がニーデル、スカートをはいている。白い服をしておりが怪しい体つきをしていてそこそこ顔がある。",
7 "personality": "お姉さん系。面倒臭がよく、清楚しくていい。シェイプアップを待たせている。",
8 "conversation_sentence_1": "まあ仕方ないですね。ここの人だけでいいですよ。",
9 "conversation_sentence_2": "まあいい。でもお姉さん系。",
10 "conversation_sentence_3": "私もありがたうって伝えたかった。",
11 "speaker_x": "0.5",
12
```

複数widgetにどう対応するか

主に三つの対応がある

- ① switcherを使いrootwidgetにまとめてしまう
- ② 各widgetを完全に独立させ、pawn等からすべて制御する
- ③ 1、2の融合タイプ

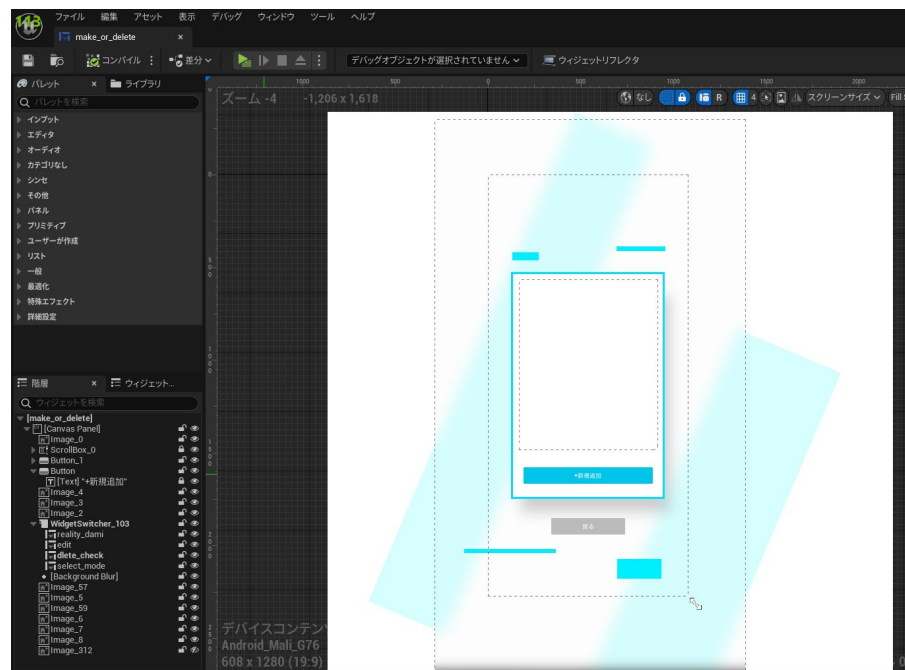
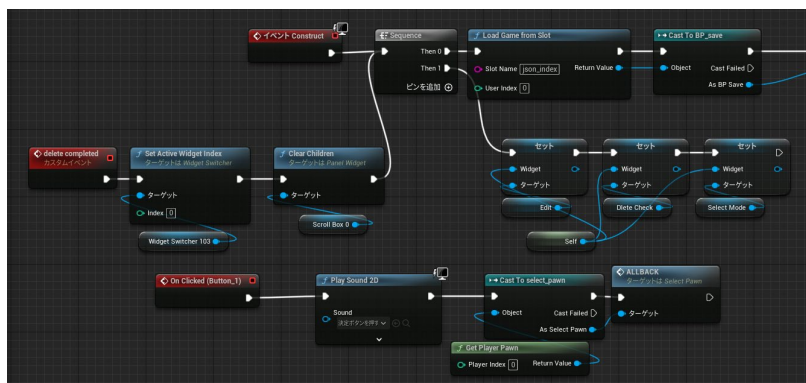
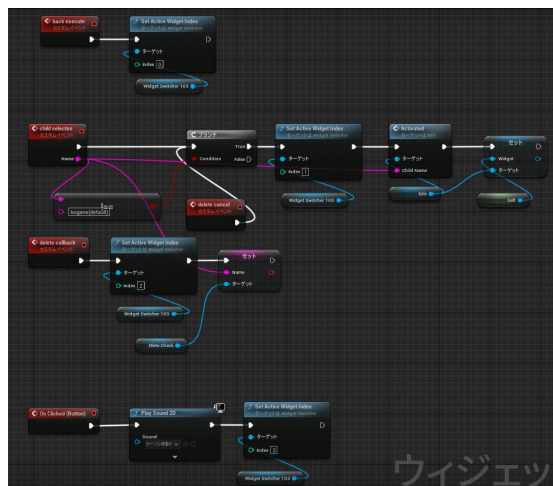
どうしたのか

→②寄りの③

各メリットデメリットと対策

- ①実装自体は簡単であり、排他的処理も簡単だがroot下にあるwidgetを全てメモリ上に保持し続ける為メモリ効率の悪くリソースが限られるandroidでは不向き
- ②は排他的処理の難易度やバグの発生確率が上がり、最初に芋づる式にロードが入るため最初だけは時間がかかる。しかし軽い

具体的なUI実装 1/2

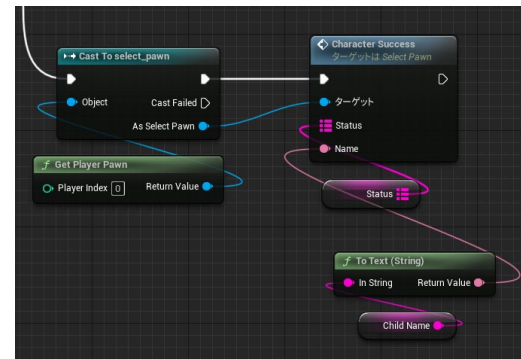


Widgetswitcherを使用しているのが分かると思う。これは軽めの各表示(削除や編集等のポップアップ)のみに使用している。Switcher下のwidgetと通信の際、上位(root)から下位(child)へは通信できるが逆からではできないという問題がある。これを解決するために下位のwidgetの変数に上位のselfオブジェクトを持たせることで通信を可能にしている。

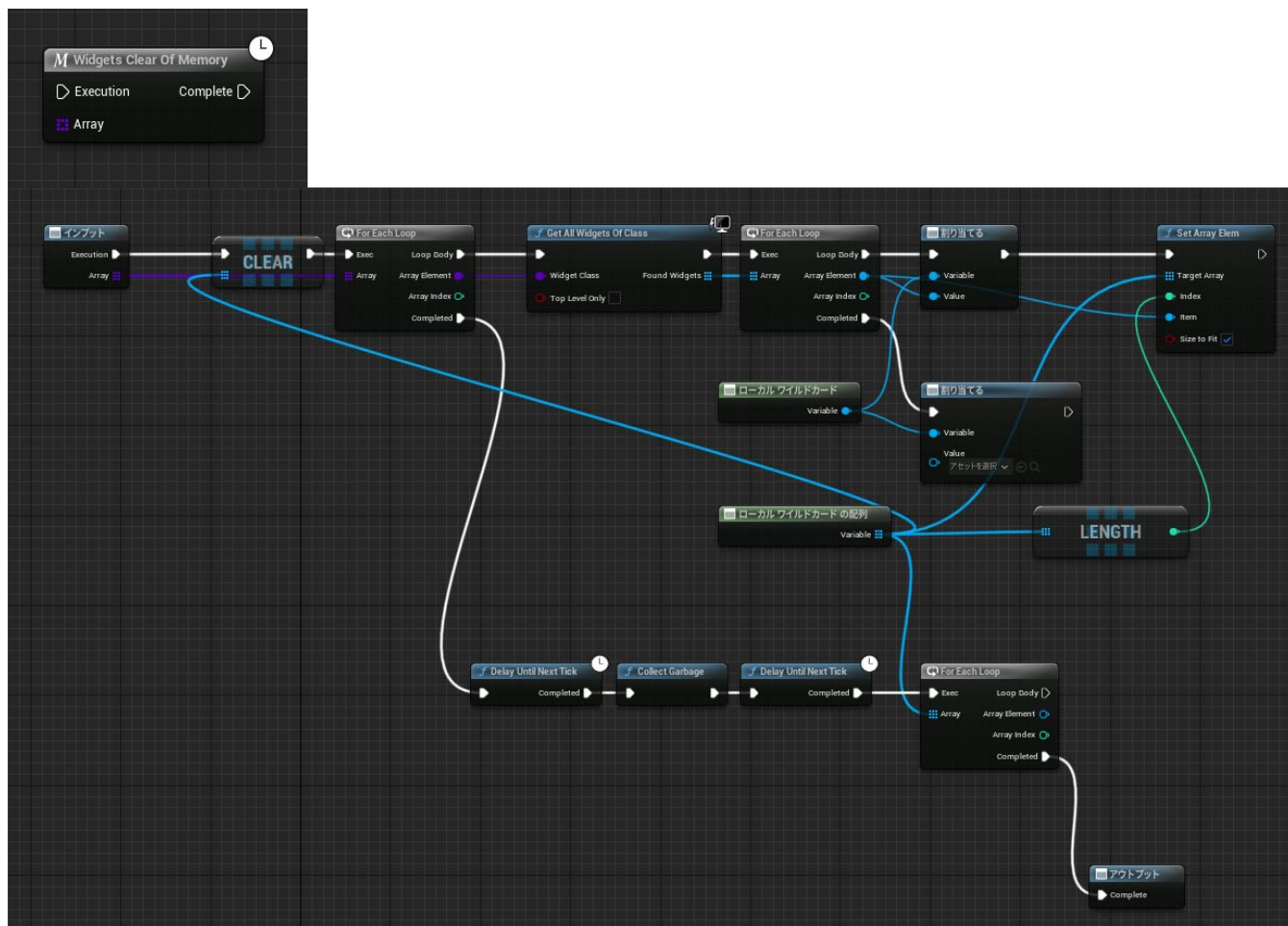
具体的なUI実装 2/2



これはselect_pawnのBPの一部である。
各widgetのアクティベート/ディアクティベートはカスタムイベントを契機に行われる。
カスタムイベントとはBPにおける手続き実装の基本的なものである。
BPの通信にはクラスの依存度を下げるためにインターフェースを使用するのが良い場合が多い。
しかし今回はselect_worldにおいてplayercontrollerが変わることが無いことが前提の設計のため、実装しやすく通信が容易なcastを使用した。



UIとGC



UIは基本”remove…”系の関数で消すのだが、最初これはdestroy系と同様にメモリ上からも消されると思っていた。しかしこれは間違っており非表示になっているだけであった。これが原因でandroid上でヒッチが発生していた。

これを解決するためにactorを基盤クラスとするマクロ”BP_clear_widgets”を作成した。引数にwidgetを基盤とするクラスの配列を取り、for each loopで各要素ごとのクラスのwigtresの配列を取得しループで各要素に、none(nullptr)を代入し、完了したら1f後にGCを呼び出して破棄させる。

UIとGC



<https://qiita.com/aoharuisgod/items/848ad61f86f542eab196>

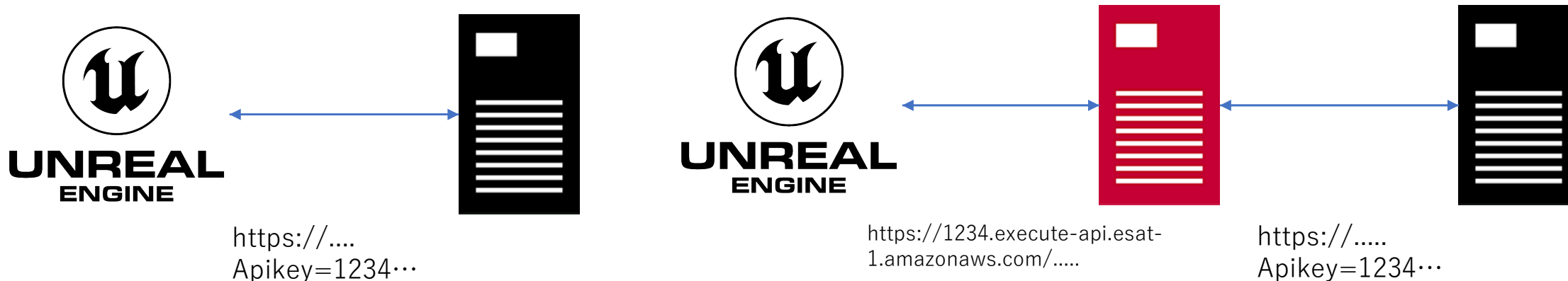
Qiitaとgithub上で解説、配布をした



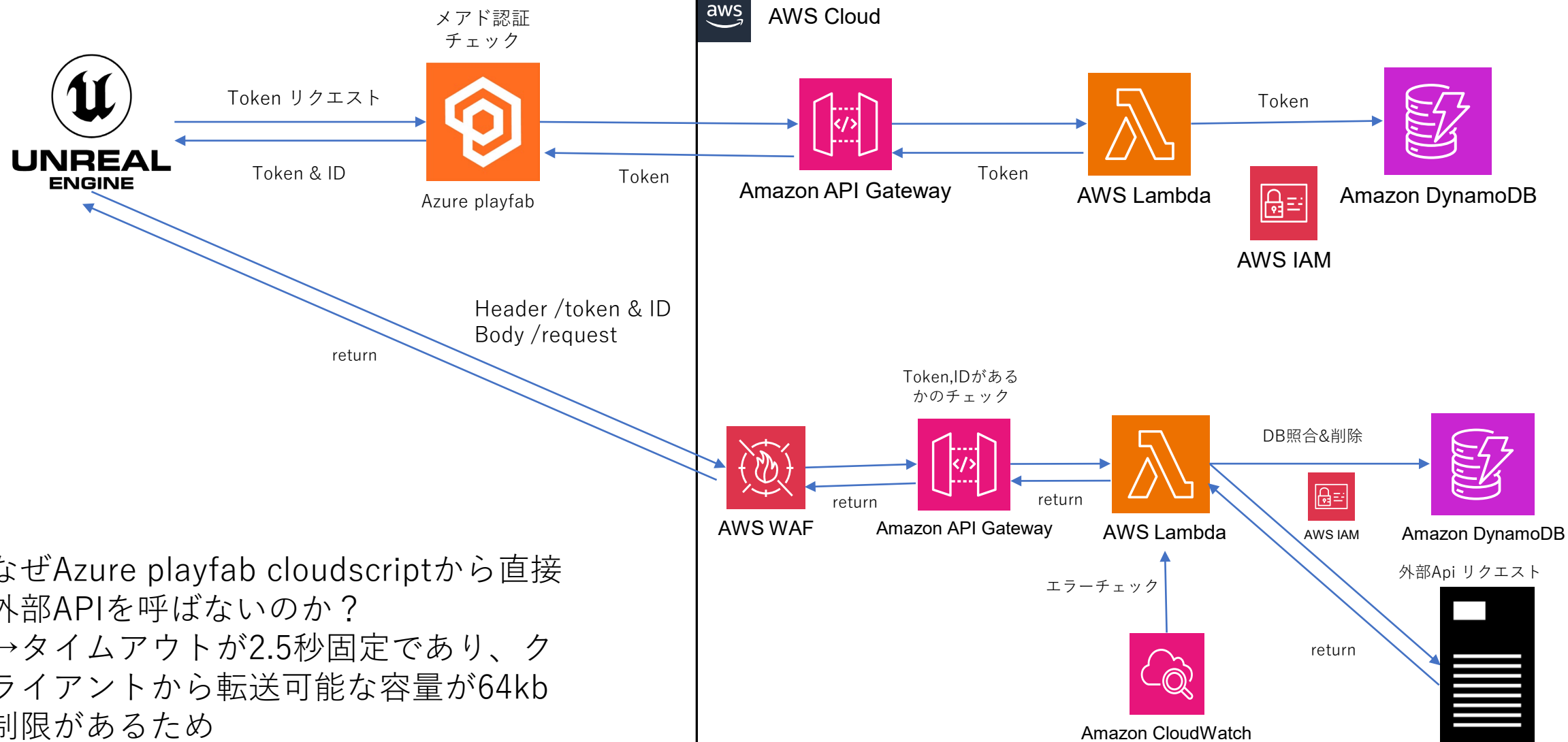
https://github.com/aoharudesu/BP_clear_widgets#bp_clear_widgets

RESTAPIとセキュリティ

今回openAIのGPT、whisperとrinna株式会社のkoemotionのAPIを使用している。
この場合APIkeyの流出、不正利用が最大の問題になる。
BPはデコンパイラーが無い為静的解析による流出は無いがcheatengineやうさみみハリケーン等によりいくらでもメモリ上の値を読み取れる。
よって自分のサーバー側でAPIリクエストを送りそれをクライアントに返す必要がある。



RESTAPIとセキュリティ



なぜAzure playfab cloudscriptから直接外部APIを呼ばないのか？
→タイムアウトが2.5秒固定であり、クライアントから転送可能な容量が64kb制限があるため

RESTAPIとセキュリティ

AWS側のDDoS、EDoS対策

図のように複数のlambda関数とapi gatewayを使用しているが一番気にすべきなのはサイバー攻撃である。まず攻撃者はどのような方法を取るか知る必要があり、qiitaで調べたところ”API gateway + lambda + S3でDDoS攻撃を受けて1日あたりで\$3000溶かした話”(https://qiita.com/naka345/items/98582cfb45ab09c3df9f)を見つけた。どうやらアプリからエンドポイントを抜かれる以外にもエンドポイントに対するブルートフォース攻撃もあるようだ。よって以下の対策を取った。

①API Gatewayのステージ名はランダムかつ長く設定する(default,dev,test等はNG)

→ブルートフォース攻撃を食らう可能性を大幅に下げられる

②各APIごとの適切なメソッドスロットリングの設定

→万が一攻撃を食らっても最小限の被害で止められる

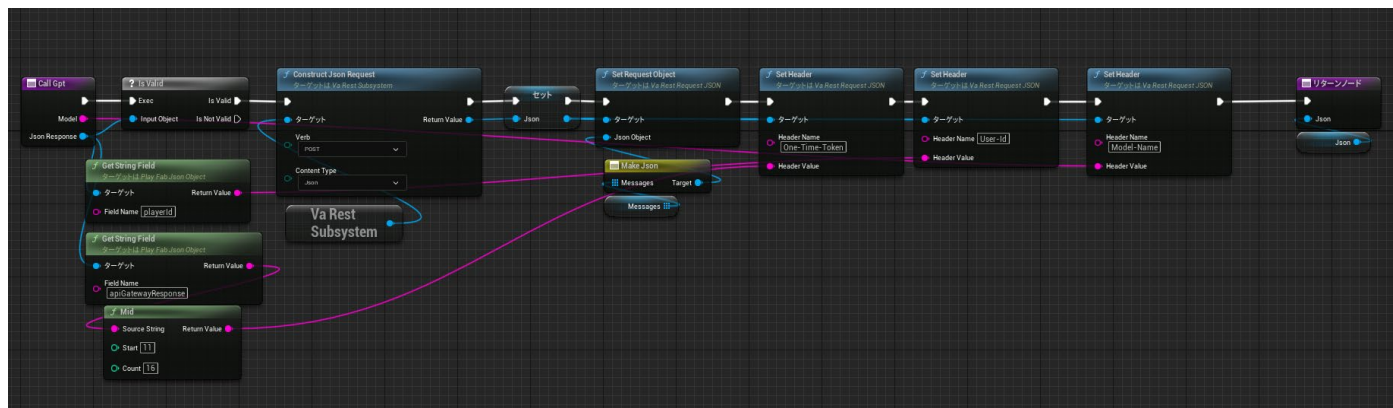
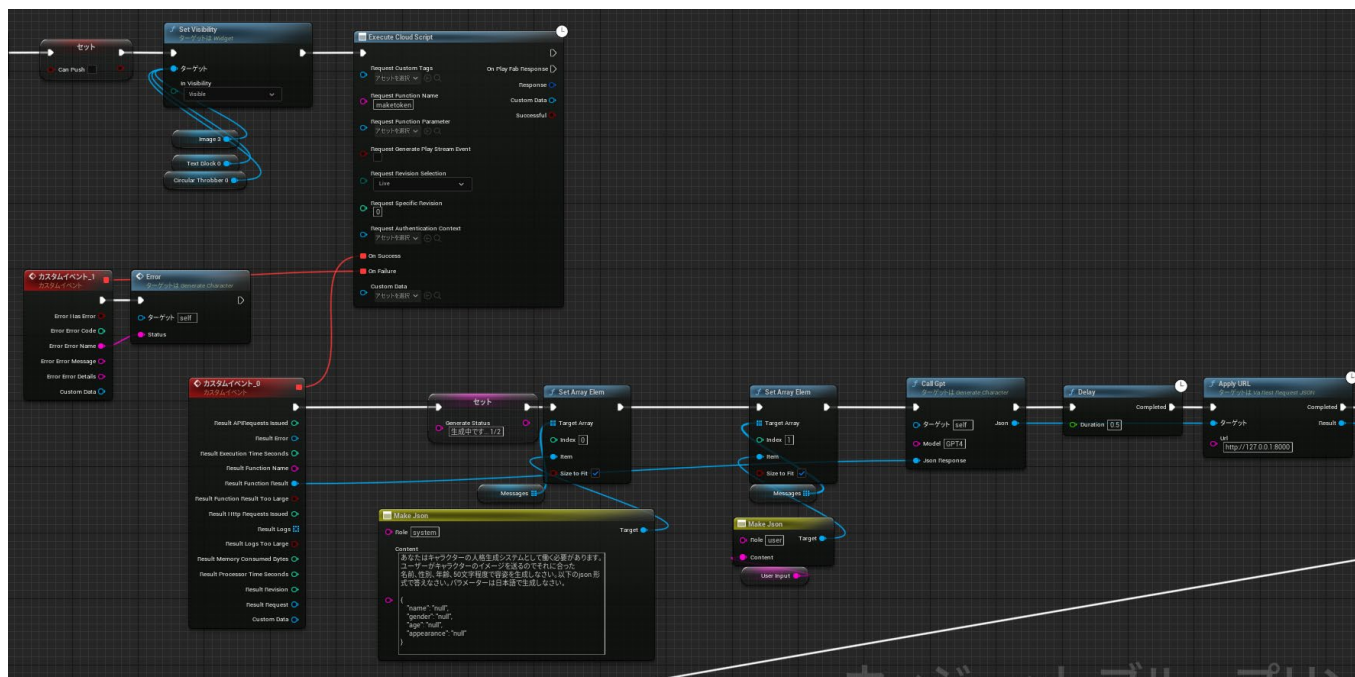
③API Gatewayでのヘッダーの存在の確認

→適切なヘッダーが無い場合lambda関数の起動自体を抑えられる

④Web ACLの適応

→クライアント側に提供されるエンドポイントへの保護(リリース前なので現在停止中)

RESTAPIとセキュリティ



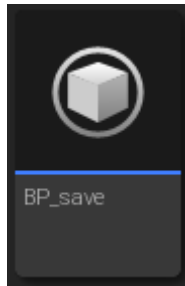
実際のBP。

Execute sloud scriptでcloudscriptを実行し、
戻り値のjsonからID,tokenを取り出しpost
requestを構築しエンドポイントに投げている
下はローカルにpostした結果。(python,flask)

[illegible]

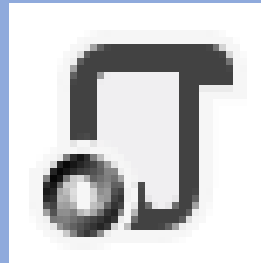
Jsonとindex

キャラクターの作成データはjson形式で保存している。これは将来的にAzure playfabのDBへの保存を考えた設計である。その為にunreal engine独自のセーブクラスにjsonの名前のみを保存しstringの配列型で保持している。整合性の保持のためにinstanceから起動時にsaveとjsonのデータのチェックを行っている。又jsonのフィールドが増えても対応可能なようにUI側でもループ処理で極力変更を加えずにできるようにしてある。



String array

0:kogane
1:example
2:....



```
マ: <スキーマが選択されていません>
1  {
2    "name": "桜井 こがね",
3    "gender": "女",
4    "age": "19",
5    "appearance": "銀髪ポニーテール。スカートをはいている。白い肌をしており女性らしい体つきをしていがそこそこ筋肉がある",
6    "personality": "お姉さん系。面倒見がよく肯定してくれる。少しイケメンな性格を持ち合わせている。",
7    "conversation_sentence_1": "もお仕方ないですね...こんかいだけですよ///...",
8    "conversation_sentence_2": "そうやって君は私を困らせるんだから",
9    "conversation_sentence_3": "私もありがたうって伝えたかった",
10   "speaker_y": "7.3",
11   "speaker_x": "5.5"
12 }
```

プロンプトエンジニアリングと正規化

本アプリケーションの最大の目玉であるLLM(GPT4)であるが、これを使いこなす上でかなりの工夫を要求される。現在世界中で行われており正解などはないが自分なりに最適化を行った。以下が重要視した点である。

①正規化表現ができているか

→AIの返答を元にjsonを構築、field名を元に値を取得する為json形式で返答させることが絶対条件となる

②返答が長くないか

→GPT4はtoken数が増えれば増えるほど曲線的に返答を生成するのに時間がかかるようになる。返答時間が長すぎるとAWS lambdaの同時実行可能数を超えてしまったり、タイムアウトしてしまったり、コストが高くなったりしてしまうためできる限り短い時間で返答を返させる必要がある。

又ChatGPTをはじめとするサービスによりあたかも人間が中にいるような挙動を感じると思うが、結局は続きの単語の予測をしているだけである。その為最も有効なのは事前にフォーマットを送ってしまい、思考の誘導を行うことである。

具体的には事前に割り当てられるroleの”system”,”assistant”,”user”を適切に割り振り、特に”system”で形式を指定し具体例の提示をすることで思考の誘導に成功している。

プロンプトエンジニアリングと正規化

“assistant”, “user”, systemとは何か？以下の役割がある。

assitant→過去のGPT4の返答を保持 / user→ユーザーからのインプットを保持 / system→システム(開発者)の指示

例えば人格生成システムではGPT4に2回リクエストを送ることで人格を生成している。以下がプロンプトである

1回目

system:あなたはキャラクターの人格生成システムとして働く必要があります。ユーザーがキャラクターのイメージを送るのでそれに合った名前、性別、年齢、50文字程度で容姿を生成しなさい。以下のjson形式で答えなさい。パラメーターは日本語で生成しなさい。

```
{
  "name": "null",
  "gender": "null",
  "age": "null",
  "appearance": "null"
}
```

user: //user input here (ex.お姉さんキャラ)

2回目

system:あなたはキャラクターの人格生成システムとして働く必要があります。すでにユーザーによってイメージ、名前、性別、年齢、容姿が決められています。性格、3つの会話文を生成しなさい。以下のjson形式で生成しなさい。パラメーターは日本語で生成しなさい。

```
{
  "personality": "null",
  "conversation_sentence_1": "null",
  "conversation_sentence_2": "null",
  "conversation_sentence_3": "null"
}
```

user: //user input & GPT response here(ex.お姉さんキャラ & “name”:“深月 紫子”, “gender”:“女性”, “age”:“25歳”, “appearance”:“艶やかな黒髪と優雅な笑顔、紫色の瞳。スタイル抜群なスレンダーボディ”)

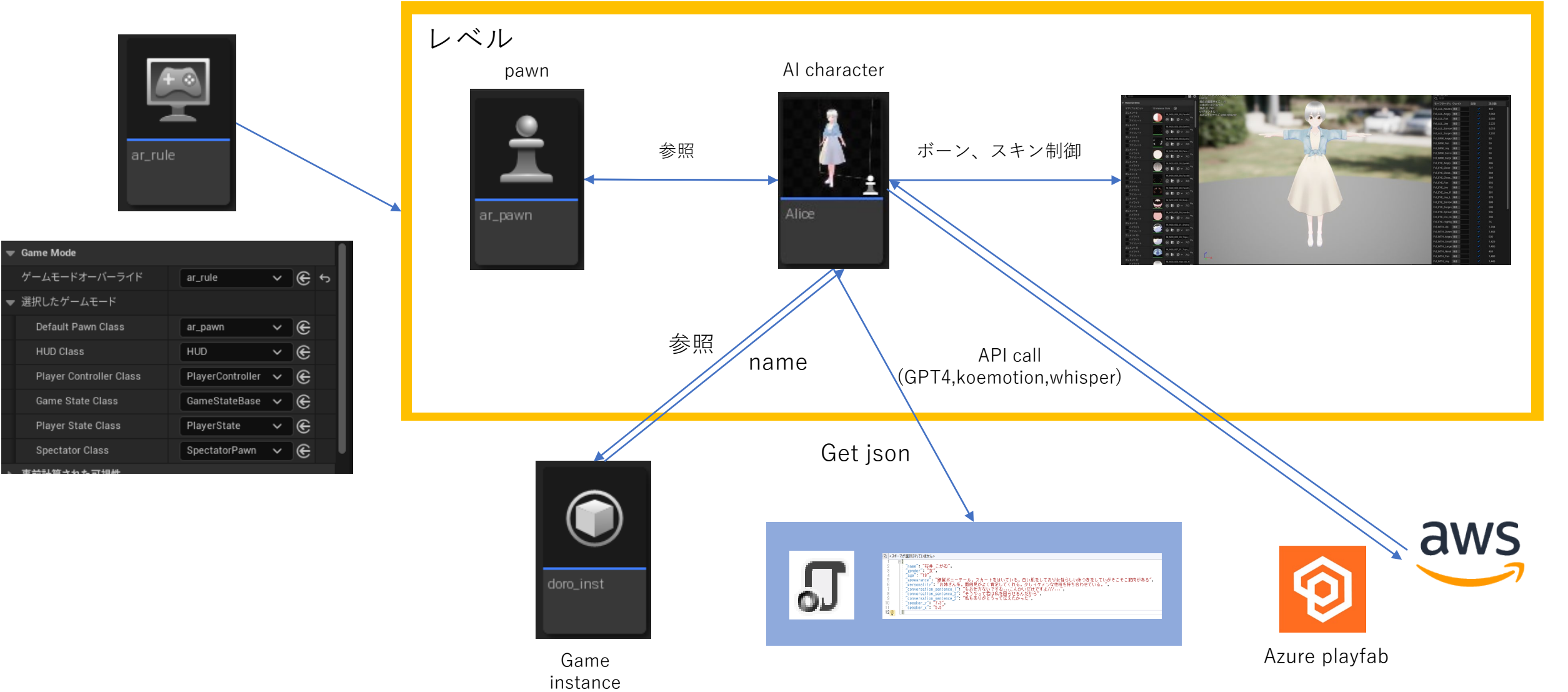
プロンプトエンジニアリングと正規化

以下が実際の返答例である。user:お姉さんキャラ

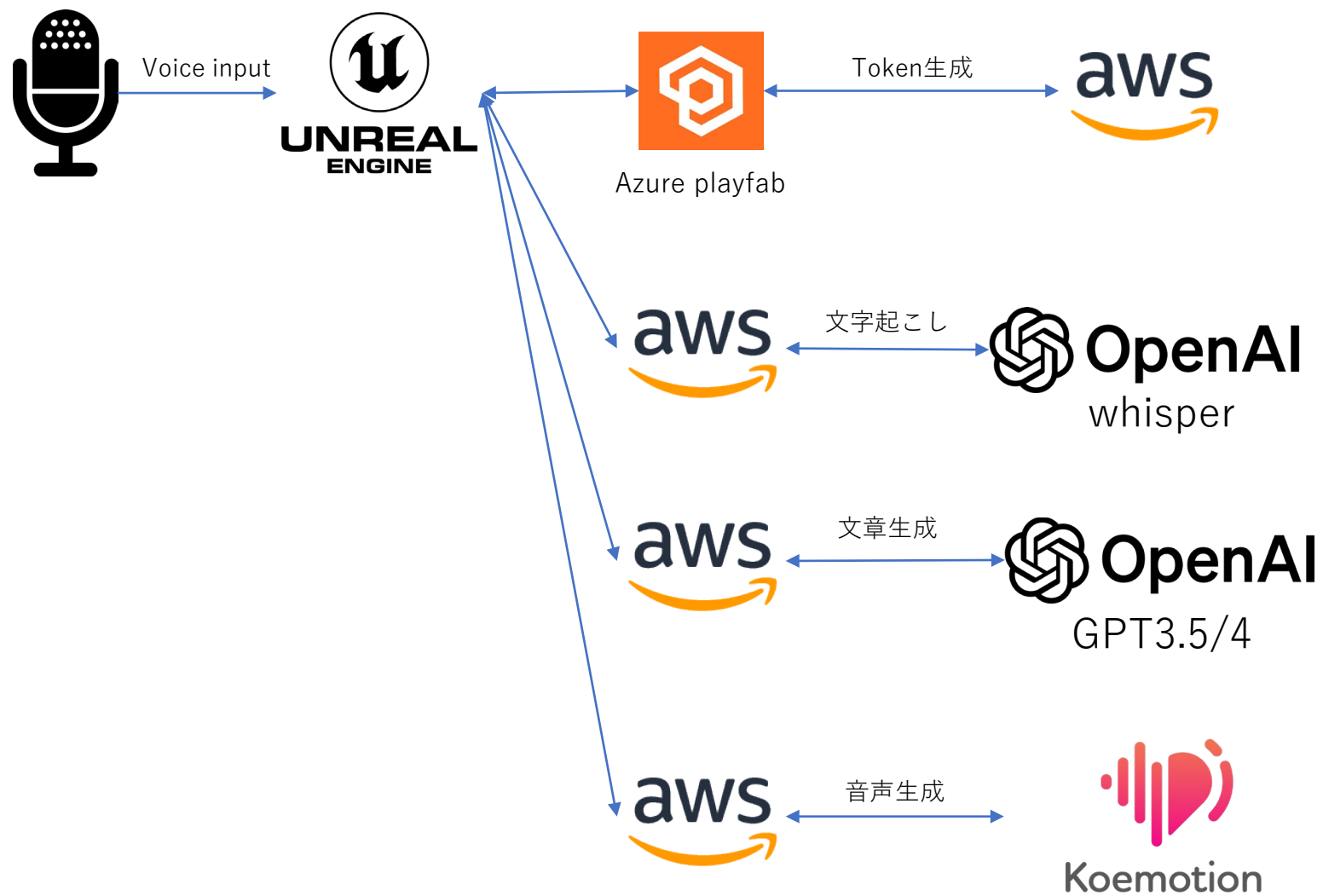
```
1回目
{
  "id": "chatcmpl-7qbYbXvJdlxyA3SUU5XU49tp9pKer",
  "object": "chat.completion",
  "created": 1692771781,
  "model": "gpt-4-0613",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "{¥n  ¥"name¥": ¥"深月 紫子¥",¥n  ¥"gender¥": ¥"女性¥",¥n  ¥"age¥": ¥"25歳¥",¥n  ¥"appearance¥": ¥"艶やかな黒髪と優雅な笑顔、紫色の瞳。スタイル抜群なスレンダーボディ。¥"¥n}"
      },
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 162,
    "completion_tokens": 83,
    "total_tokens": 245
  }
}

2回目
{
  "id": "chatcmpl-7qbYotp7Z5mOvBjXuDRQPqakEZsfr",
  "object": "chat.completion",
  "created": 1692771794,
  "model": "gpt-4-0613",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "{¥n  ¥"personality¥": ¥"面倒見が良く、おっとりとした優しい性格。頼りにされることが多い。¥",¥n  ¥"conversation_sentence_1¥": ¥"あら、大丈夫？何かお困りなら何でも言ってね。¥",¥n  ¥"conversation_sentence_2¥": ¥"今日は何をしたいの？私はおまかせするわ。¥",¥n  ¥"conversation_sentence_3¥": ¥"これを飲んで、リラックスして。疲れているのかしら？¥"¥n}"
      },
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 255,
    "completion_tokens": 135,
    "total_tokens": 390
  }
}
```

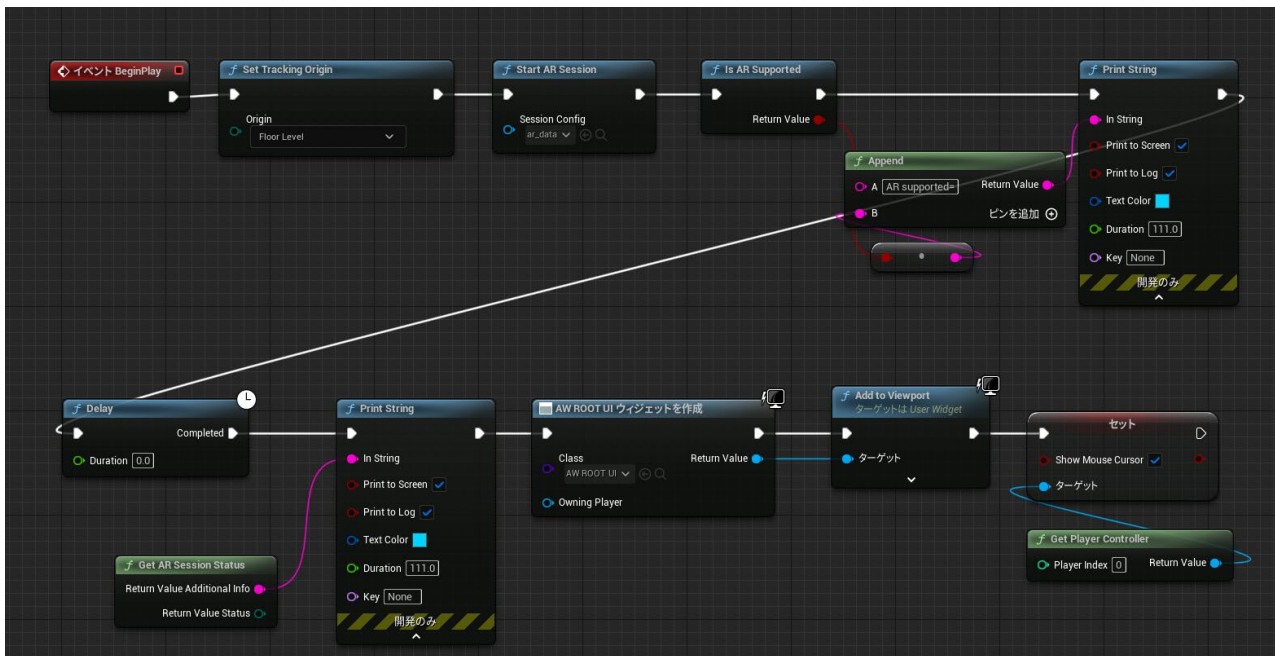
Alice_world



Alice_world



UnrealEngineとGoogle AR Core



google公式が出している Google AR core sdkと Google AR core Service sdkを有効化するだけで AR sessionを開始できる。

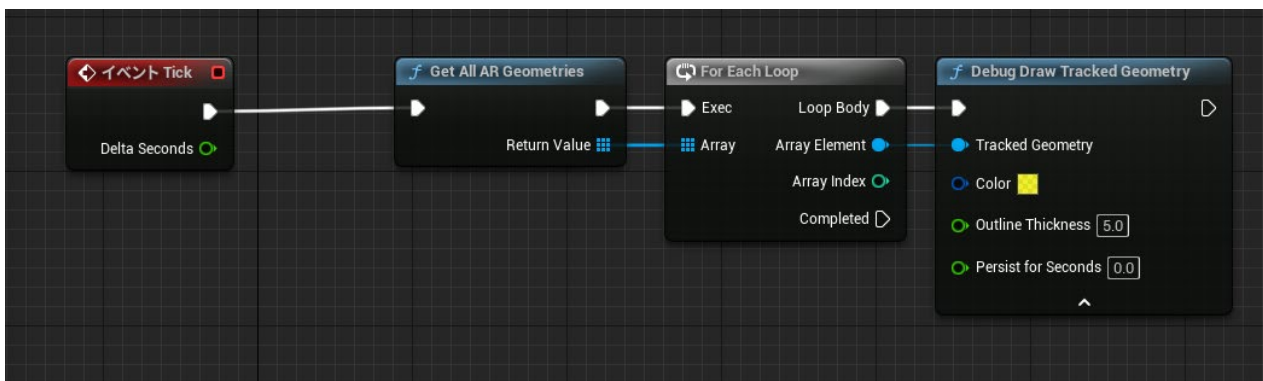
図の処理の流れは

Beginplay(スタート時に呼び出される)→セッション開始→ステータスチェック→widget構築、追加→カーソル表示

Tick(1fごとに呼び出される)→すべてのジオメトリーを取得→for eachで各ジオメトリーを描画

である。

又特徴としてジオメトリーの場所が確定することで初めてワールドのorigin座標が確定することになる



[illegible]

LLMによる感情表現と3Dキャラクターの制御



笑顔

モーフターゲット	ウェイト	自動	頂点数
PolALL_Neutral	0.0	✓	469
PolALL_Angry	0.0	✓	1,068
PolALL_Fun	0.0	✓	2,082
PolALL_Joy	0.0	✓	2,222
PolALL_Sorrow	0.0	✓	2,018
PolALL_Surprise	0.0	✓	2,202
PolBRW_Angry	0.0	✓	60
PolBRW_Fun	0.0	✓	60
PolBRW_Joy	0.0	✓	60
PolBRW_Sorrow	0.0	✓	60
PolBRW_Surprise	0.0	✓	60
PolLEY_Angry	0.0	✓	386
PolLEY_Close	0.0	✓	727
PolLEY_Close	0.0	✓	384
PolLEY_Close	0.0	✓	384
PolLEY_Fun	0.0	✓	556
PolLEY_Joy	0.0	✓	731
PolLEY_Joy_R	0.0	✓	381
PolLEY_Joy_L	0.0	✓	378
PolLEY_Sorrow	0.0	✓	588
PolLEY_Surprise	0.0	✓	688
PolLEY_Speak	0.0	✓	596
PolLEY_Hiss_Hi	0.0	✓	288
PolLEY_Highlig	0.0	✓	76
PolMTH_Up	0.0	✓	1,394
PolMTH_Down	0.0	✓	1,443
PolMTH_Angry	0.0	✓	636
PolMTH_Small	0.0	✓	1,429
PolMTH_Large	0.0	✓	1,486
PolMTH_Neutral	0.0	✓	469
PolMTH_Fun	0.0	✓	1,480
PolMTH_Joy	0.0	✓	1,445

怒った顔



3Dモデルは主にboneとmesh、materialから構成される。Vtuberをはじめとする人型の3Dモデルの規格にvrm(3Dモデルの人型アバター専用のファイルフォーマット)があり、今回はこの規格に沿ったものを使用している。vrmではboneに事前にモーフターゲットというものが存在しており、パラメーターを変えることで各boneを動かすことができる。今回はLLMに対しzero shot(few shotとも言う)をインジェクションし、感情パラメーターのjson fieldを返答させ、この値を元に表情と声を制御している。又レスポンスにひらがなで返答させるようにさせることでLip Synch(口の制御)を行っている。詳しくは次の資料で解説する。

LLMによる感情表現と3Dキャラクターの制御

以下がデフォルトキャラの”kogane”へのプロンプトと返答である。

```
system:

あなたは与えられた人格に基づいてそのキャラになりきる必要があります。アシスタントのような振る舞いはしてはいけません。
人格は以下の通りです。
{
  "name": "桜井 こがね",
  "gender": "女",
  "age": "19",
  "appearance": "銀髪ポニーテール。スカートをはいている。白い肌をしており女性らしい体つきをしていがそこそこ筋肉がある",
  "personality": "お姉さん系。面倒見がよく肯定してくれる。少しイケメンな性格を持ち合わせている。",
  "conversation_sentence_1": "もお仕方ないですね...こんかいだけですよ///...",
  "conversation_sentence_2": "そうやって君は私を困らせるんだから",
  "conversation_sentence_3": "私もありがとって伝えたかった",
  "speaker_y": "7.3",
  "speaker_x": "5.5"
}
```

anger、joy、pleasure、sadness、surpriseの5つのパラメーターを返す必要があります。各パラメータは0～0.7で評価されます。5つの内1つ又は2つの値を返す必要があります。60文字以下でないといけません。
返答はひらがなにもする必要があります。
感情スタイルtalk、happy、sad、angry、fear、surprised、whisper、をパラメーターを返す必要があります。
以下がフォーマットです。json形式で答えなさい

```
{
  "response": "your_response",
  "response_in_hiragana": "your_response_in_hiragana",
  "anger": " your_parm ",
  "surprise": "your_parm",
  "style": "your_style"
}
```

assistantには"response"の内容しか乗せませんが、きちんとjson形式を保ってください。

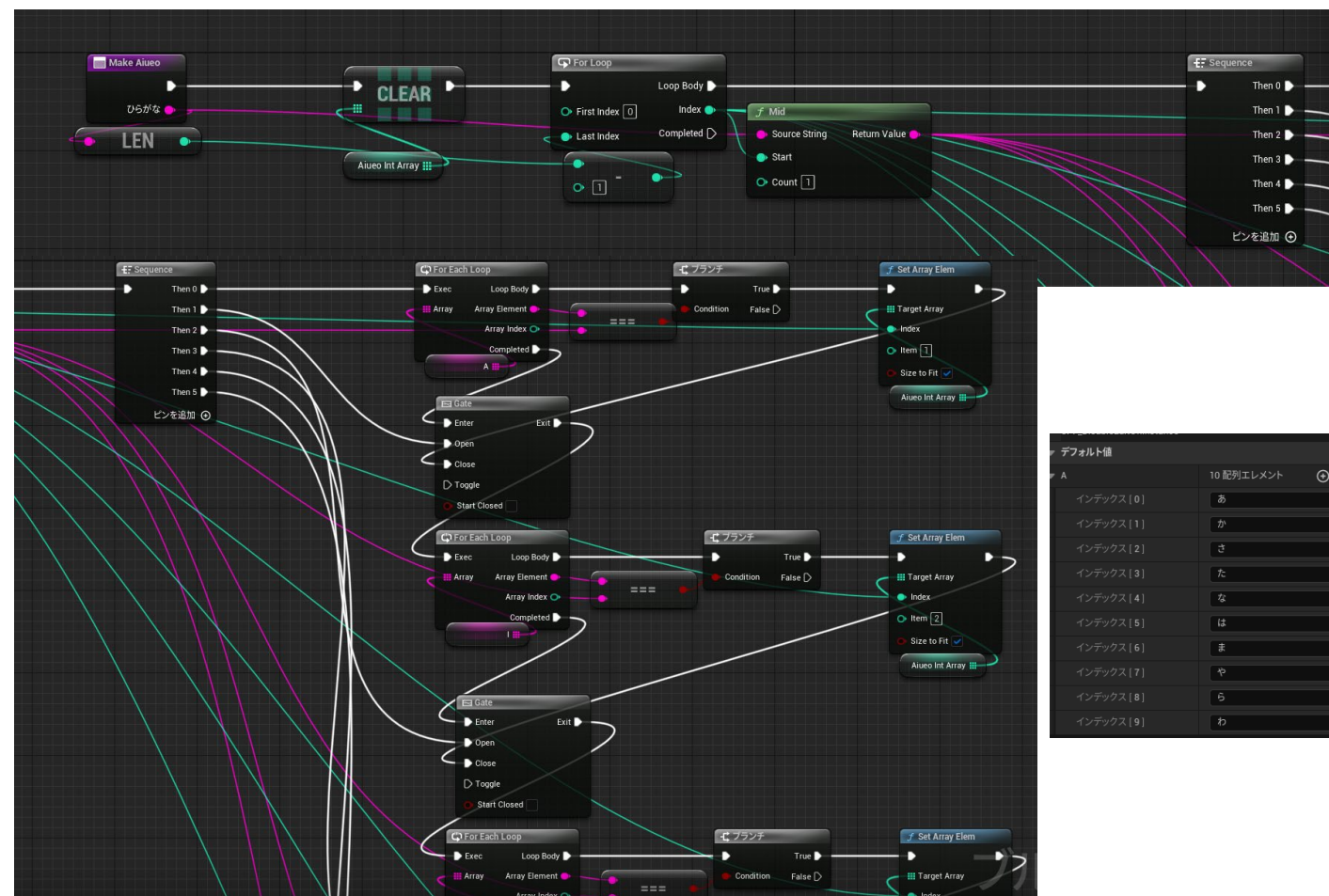
```
user:

//user input here

GPT4の返答

"choices": [
  {
    "index": 0,
    "message": {
      "role": "assistant",
      "content": "{\n  ¥"response¥": ¥"最近気になるニュースとか、何かありましたか？世の中の動きって、私たちの生活に直結していることも多いですから。¥",¥n  ¥"response_in_hiragana¥": ¥"さいきん きになる にゅーすとか、なにか ありましたか？よのなかの うごきって、わたしたちの せいかつに ちょくせつ していることも おおいですから。¥",¥n  ¥"anger¥": ¥"0¥",¥n  ¥"surprise¥": ¥"0.2¥",¥n  ¥"style¥": ¥"talk¥"¥n}"
    },
    "finish_reason": "stop"
  }
]
```

LLMによる感情表現と3Dキャラクターの制御



Lip Synchだが、先ほど用意した
“your_response_in_hiragana”fieldからstringの
文字列を取得し、文字数分ループを実施→母音
がaiueoの各stringの配列と同じかを比較し
A(あかきたなはまやらわ)→1、
I→2、U→3、E→4、O→5、それ以外→0とし、
int型の配列を作る。例えば”こんにちは”なら
55221となる。

koemotionAPIで帰ってきた音声の秒数を
文字数で割り、1文字当たりの秒数を求め
Lip Synchを行っている。

デフォルト値	
A	10 配列エレメント
インデックス [0]	あ
インデックス [1]	か
インデックス [2]	さ
インデックス [3]	た
インデックス [4]	な
インデックス [5]	は
インデックス [6]	ま
インデックス [7]	や
インデックス [8]	ら
インデックス [9]	わ

トライ&エラー&エラー&エラー.....

よくunityと比べられるunrealengineだが、明確に負けている要素がある。それはコミュニティの広さと情報量の差である。そして何よりも環境依存のバグが多い。以下が開発においてつまづいた点である。

- ・ androidでの書き出しができない

→ドキュメント通りにやってもできない。海外の英語の動画みてもエラー。救ってくれたのは個人ブログでした。30時間ほどの検証の末にVS code 2022+SDK31で安定した。Qiitaに備忘録付けといた。

- ・ C++を追加するとプロジェクトが確率で文鎮化する。

→原因不明。多分エディターの気分次第。最初にこうなったときはあきらめたが、ファイルを20時間ほど弄繰り回したところ復活方法が判明。

- ・ プラグインでなぞのエラー。コンパイルに失敗する。

→ソースコードを直接ビルドすることで解決。この手のエラーに累計100時間以上取られている。

- ・ EPIC公式のリポジトリからクローンしたのになぜかアクセス拒否

→github認証などをしてみたが無理だった。解決策見つからず(ただ問題はない)

- ・ audio compomentにアクセスするとエラーになる。

→変数に昇格させてもエラーになるので悩んでいたら処理順が違った。

トライ&エラー&エラー&エラー.....

- ・ playfabでのプレイヤーステータスの取得方法が分からない
→unityで解説されているブログを読み、雰囲気ですべてを読み何とか実装
- ・ playfabでメアドステータスがどうしても取れない
→playfab communityでunityのC#での回答を発見し解説、実装。フラグを入れたjsonを送る必要があった。
- ・ playfab cloudscriptのドキュメントが消されており詰む
→メインのドキュメントは消えていたが、 community内の掲示板のURLから飛ぶと断片的に情報が！
- ・ ドキュメントを読んだところで分からない
→GPT4とbingAIに100回以上質問した末に何とか動かすことができた
- ・ ue4.27から5.11への移行で数えれないほどのエラー
→twitterでの情報を元に修正
- ・ 5.11へ移行したことで従来のアニメーションリターゲットが使用不可に
→ue4.27で変換したものを5.11で無理やり合わせることで解決
- ・ そもそもAWSってなに？
→qiitaの記事を読み続けた(100記事ほど)

トライ&エラー&エラー&エラー.....

- ・ハッキング、不正の手段が分からないから防ぎようがない

→youtubeでハッキング系の動画をあさり、CTF勢の方に直接twitterでDMをした。Zenn、qiitaでまとめられているものを片っ端から見た。サイバーエージェントとLINE、MIXIが出してるスライドが参考になった。

- ・AWS S3でOACに関する話が最新すぎてない

→対照実験と勘で対応

- ・APIkeyに関するリスクについてセキュリティ上客観的な話が知りたい

→qiitaで質問

- ・そもそも多重ネスト構造のjsonの扱いが分からない

→UE5独自の話のためローカルホストにポストを繰り返し、対照実験で把握

- ・API Gatewayで送ると502エラーをlambda関数が返す

→jsonのデコードの問題だった。Cloud watchでログを解析し判明。

- ・GPT3.5だとpostに成功するのにGPT4だと成功しない

→lambda関数のデフォルトタイムアウトの設定の問題だった。最初jsonを上手く構築できていないのではないかと疑ったが、長文を送るとタイムアウトになることに気が付きタイムアウトの存在を認識。

トライ&エラー&エラー&エラー.....

- ・ ARcoreが起動しない

→googleのドキュメントはopenGLをvulkanどちらにも対応とあったが、ueではopenGLにしか対応していなかったのが原因。Communityフォーラムをあさって発見。Qiitaを書いておいた。

- ・ 5.20でARセッションを開始しても何も表示されない

→ Communityでも話になっており、修正待ち

- ・ htmlwidgetがモバイルだと表示されない

→どこにもドキュメントが無かったが、対照実験でvulkanでないと映らないことが判明。Twitterで発信

- ・ androidAPI33で起動できないバグ

→ Communityで起動時のスプラッシュを無効化することで直るとの書き込みを発見。 Twitterで発信

- ・ どうやっても書き出しできない

→プロジェクト名に特定の文字列を含むとエンジン側がエラーを起こすとブログで発見。

- ・ その他無数の小さなバグたち

→勘で対処

サイト自体の技術スタック (一部は本アプリケーションとは関係ありません)

使用言語

html/javascript/css/python

クラウド

AWS

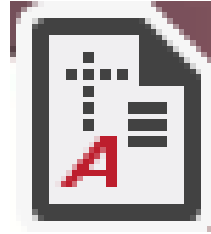
lambda/API Gateway/S3/CloudFront
/Route53/ACM/IAM

GCP

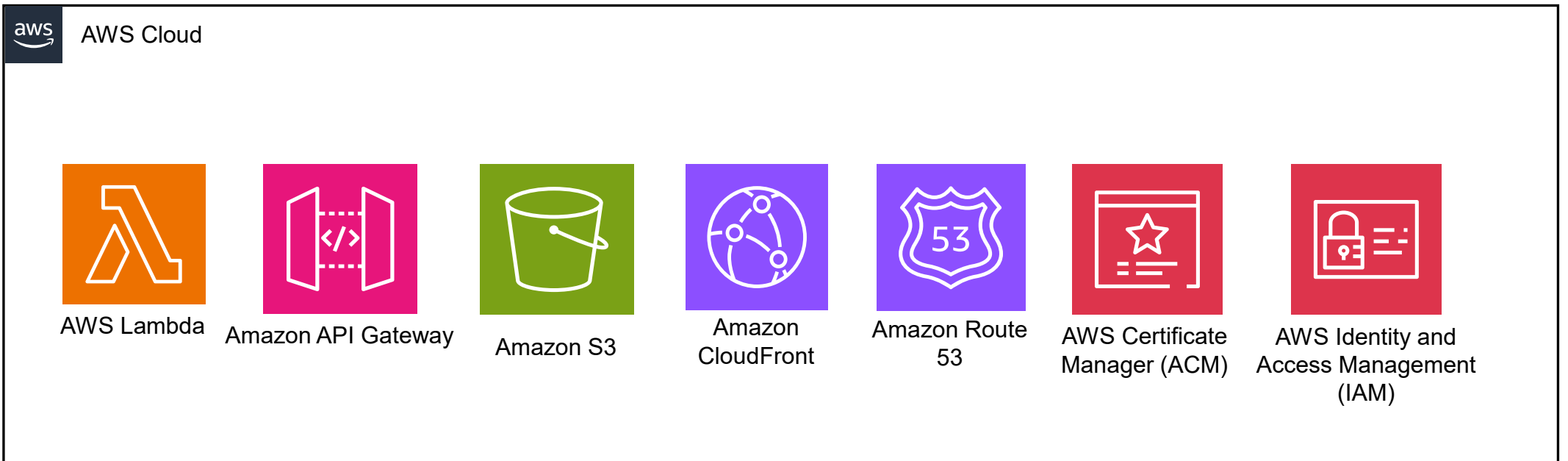
reCAPTCHA



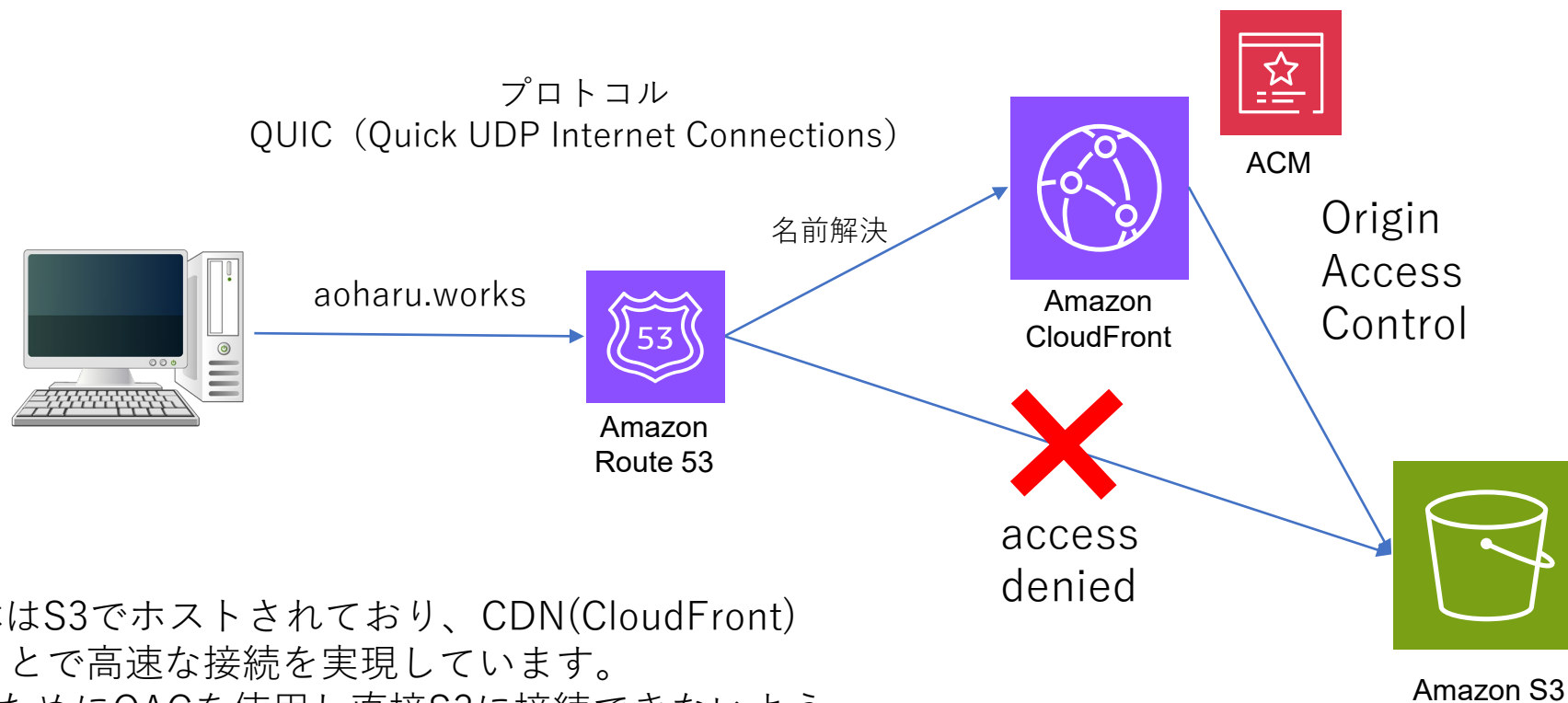
css



javascript



高速なアクセスの実現



まずweb自体はS3でホストされており、CDN(CloudFront)を活用することで高速な接続を実現しています。DDoS対策のためにOACを使用し直接S3に接続できないようになっています。httpでアクセスがあった場合httpsに強制リダイレクトさせています。通信プロトコルはTCPを使うTLS1.3ではなくUDPを使用するQUICを採用しました。これによりハンドシェイク数を減らし高速に接続ができます。