

自動会話プログラム (Chat Bot)

の制作

香川 怜 河上 優志
東 凌河

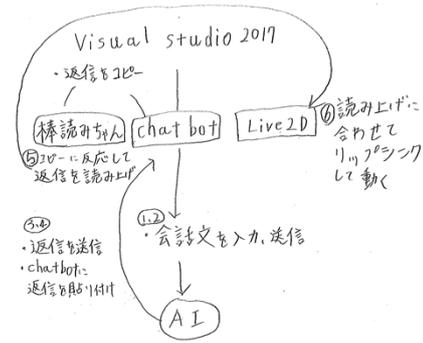


図 2 Chat bot の構成

1. まえがき

私たちは、課題研究で Visual Studio などを用いて Chat bot の作成をした。Chat bot とは、「チャット」と「ボット」を組み合わせた言葉で、人工知能を活用した「自動会話プログラム」のことである。

2. 原 理

この Chat bot の原理は、図 1 の画面に表示されるテキストボックスに会話文を入力すると、インターネット上の AI がその返事をテキストで返すのでそれを合成音声で読み上げて言葉に合わせてキャラクターが動くものである。

Chatbot

完全自動の日常会話を可能にするAIです。会話を学習していくことでより賢く・より自然な会話を実現します。

他人の名称・メールアドレス・住所・電話番号など個人を特定しうる情報はアップロードしないでください



図 1 Chat bot の入力画面

今回、インターネット上の AI はリクルート社の A3RT の Talk API を使用させていただいた。

3. 研究内容

まず、Chat bot の大まかな構成をペイントツール描いてみた(図 2)。

(1)Chat bot の動作の流れ

- ①テキストボックスに会話文の入力、インターネット上の AI に送信
- ②AI の応答をテキストベースで受信して表示
- ③応答テキストをペーストバッファにコピー
- ④応答テキストを合成音声ソフトで読み上げる
- ⑤キャラクターが合成音声ソフトに合わせて動く

(2)テキストボックスに会話文の入力、送信

最初にこの Chat bot の AI と会話をするためのユーザー側の会話文を入力して送信するためのプログラムを作成した。

```
protected void TextBox1_TextChanged(object sender, EventArgs e)
{
    TextBox2.Text = TextBox1.Text + Environment.NewLine + Environment.NewLine + TextBox2.Text;

    //文字コードを指定する
    System.Text.Encoding enc = System.Text.Encoding.GetEncoding("UTF-8");

    //POST送信する文字列を作成
    string postData =
        "apikey=Q51kBlzVZC3vtagDv00TJmD2J99HjCn&query=" +
        System.Web.HttpUtility.UrlEncode(TextBox1.Text, enc);
    //バイト型配列に変換
    byte[] postDataBytes = System.Text.Encoding.ASCII.GetBytes(postData);

    //WebRequestの作成
    System.Net.WebRequest req =
        System.Net.WebRequest.Create("https://api.a3rt.recruit-tech.co.jp/talk/v1/smalltalk");
    //メソッドにPOSTを指定
    req.Method = "POST";
    //ContentTypeを"application/x-www-form-urlencoded"にする
    req.ContentType = "application/x-www-form-urlencoded";
    //POST送信するデータの長さを指定
    req.ContentLength = postDataBytes.Length;

    //データをPOST送信するためのStreamを取得
    System.IO.Stream reqStream = req.GetRequestStream();
    //送信するデータを書き込む
    reqStream.Write(postDataBytes, 0, postDataBytes.Length);
    reqStream.Close();
}
```

図 3 入力・送信についてのプログラム

このプログラムは、textbox1 に送信したデータに、AI の API キーを貼り付けてエンコードし、データとして送信するための形に変換する。そして変換後のデータを API に送信というものである。

(3)AI の返答データを Chat bot に送信

AI の応答データを、Chat bot に送信するプログラムを作成した。

```
//サーバーからの応答を受信するためのWebResponseを取得
System.Net.WebResponse res = req.GetResponse();
//応答データを受信するためのStreamを取得
System.IO.Stream resStream = res.GetResponseStream();
//受信して表示
System.IO.StreamReader sr = new System.IO.StreamReader(resStream, enc);
String tmp;
tmp = HttpUtility.UrlDecode(sr.ReadToEnd());
//TextBox1.Text = HttpUtility.UrlDecode(sr.ReadToEnd()) + Environment.NewLine + TextBox1.Text;
TextBox2.Text = tmp + Environment.NewLine + TextBox2.Text;
//TextBox1.Text = tmp;
//Literal1.Text = "<script type='text/javascript'> window.onload=execCopy(\"\" + tmp + "\");</script>";
//Literal1.RegisterStartupScript(this.GetType(), "key", "<script type='text/javascript'> window.onload=execCopy(\"\" + tmp + "\");</script>");
sr.Close();

TextBox1.Text = "";
TextBox1.Focus();
```

図4 API の応答を送信・表示するプログラム

このプログラムは、API のサーバーからの応答を受け取るための `WebResponse` と、応答データを受け取るための `Stream` を取得し、Chat bot の `textbox2` に応答のデータを貼り付けるというものである。

(4)返答を合成音声ソフトで読み上げる

私たちは(3)で受信して表示させた AI の返信テキストを読み上げさせたいと思った。そのために棒読みちゃんというソフトを使用した。

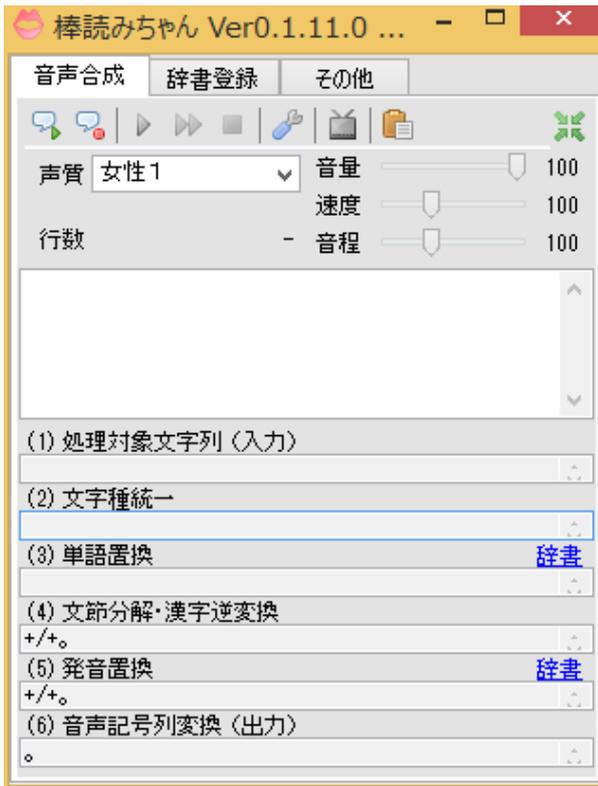


図5 合成音声ソフト「棒読みちゃん」

このソフトは画像中央の枠に入力したテキストを読み上げるソフトである。画像の上部にある「その他」からクリップボード監視を選ぶことでコピーされたテキストを読み上げさせるこ

とが出来た。

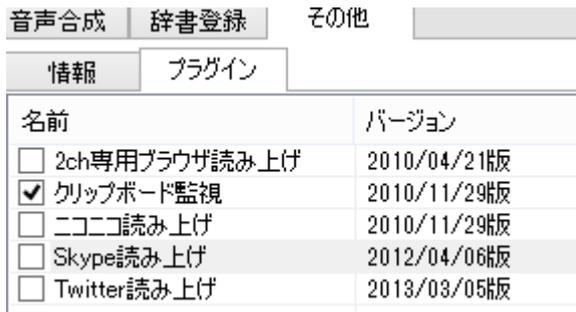


図6 クリップボード監視

(6)キャラクターが音声読み上げに合わせて動く

データを読み上げる際に `Live2D Viewer` というソフトを使用してキャラクターを音声に合わせて口を動かすようにした。

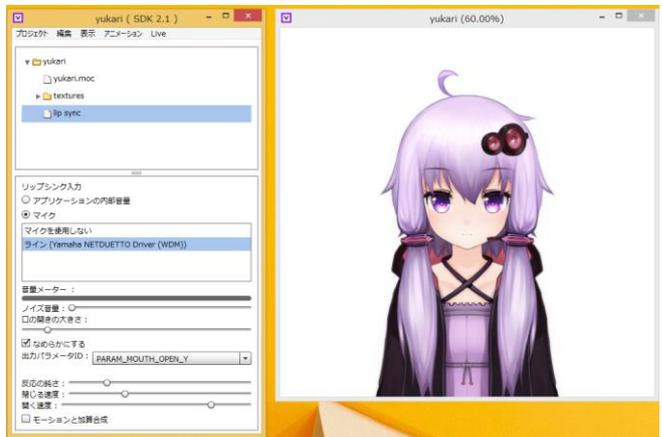


図7 Live2D の設定画面(左)とキャラクター(右)

Live2D でキャラクターを喋らせるにあたって、一つの問題が生じた。棒読みちゃんと Live2D がつながっておらず、棒読みちゃんが喋った時に、Live2D が口を動かさなかったのである。原因は Live2D のリップシンク機能がマイク等で読み取れる外部音声しか反応せず、内部音声に反応しないからである。そこで、どうにか Live2D と棒読みちゃんを繋げる為に NET DUETTO を用いた。

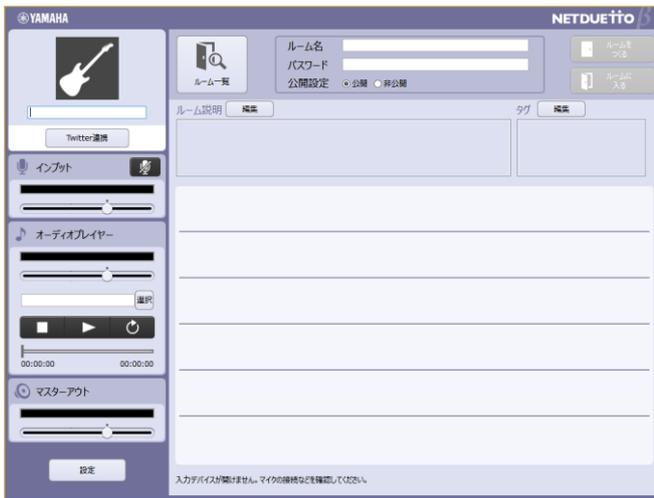


図 8 NETDUETTO の操作画面

このソフトを起動し録音デバイスを「ライン」に設定する。すると Live2D が「ライン」をマイクのような外部音声入力デバイスとして認識してしまう為、リップシンク入力(音声に合わせて口を動かす機能)に設定が出来るようになる。そして棒読みちゃんにも同じ設定を施した。これにより棒読みちゃんと Live2D を「ライン」で繋ぐことが出来、音声に合わせて口を動かせるようになった。

(5) 応答テキストの自動コピー

棒読みちゃんと Live2D を繋ぐことはでき棒読みちゃんのクリップボード監視によってコピーをすると喋るようにはなった。しかし肝心の自動でコピーする事が出来なかった。

```
System.IO.StreamReader sr = new System.IO.StreamReader(resStream, enc);
String tmp;
tmp = Regex.Unescape(getdata(sr.ReadToEnd()));
//TextBox2.Text = Regex.Unescape(getdata(sr.ReadToEnd())) + Environment.NewLine + TextBox2.Text;
TextBox2.Text = tmp + Environment.NewLine + TextBox2.Text;
//TextBox3.Text = tmp;
//Literal1.Text = "<script type='text/javascript'>window.onload=execCopy(\"" + tmp + "\");</script>";
ClientScript.RegisterStartupScript(this.GetType(),
    "key", "<script type='text/javascript'>window.onload=execCopy(\"" + tmp + "\");</script>");
sr.Close();

TextBox1.Text = "";
TextBox1.Focus();
```

図 9 自動コピー

このプログラムは AI の返答データを自動でコピーするプログラムである。

window.onload=execCopy(青線)により、そのあとの(¥""+ tmp + "¥"); (tmp には返答データの内容が入っている)をコピーする。このプログラムを追加したことにより会話文を AI に送り AI から帰ってきた返答データを画面に表示しそのテキストをコピー、そのコピーに反応して棒読み

ちゃんがテキストを読み上げ、読み上げた音声に合わせて Live2D のキャラクターが動くという一連の動作が完成した。

4. まとめ

Chat bot の制作に至るまでの過程がとても苦戦した。まず、自分たちの力で Chat bot の核であるプログラムまで作ろうとしていたが、それには現役高校生にはとても難しいことだったので Chat bot の核自体は A3RT の Talk API を使用し、それを繋ぐプログラムを作成した。何度か難しい課題もあったが、先生の助言で解決する事が出来た。Chat bot を制作していてプログラムのひとつひとつの意味を考えていくととても難しく C#のプログラムについてもっと勉強してプログラムの内容を理解していきたいと思った。今回、プログラムの中にペーストバッファにコピーするプログラムを入れて棒読みちゃんのクリップボード監視を利用したものがとても良い案だと思った。今回の Chat bot の製作により私たちの知識がかなり深まったと思う。

5. あとがき

プログラム自体は先生の助言に頼り切りだったので C#のプログラムについてもっと理解しておきたかった。棒読みちゃんや Live2D Viewer、NETDUETTO などのアプリがとても使いやすく今回の Chat bot の製作においてとても重要なものだった。NETDUETTO と棒読みちゃんと Live2D Viewer の接続の紹介をしていたサイトに良い情報が色々あったのでとても助かった。今回リクルート社の A3RT の Talk API を使った Chat bot の製作をしたが、A3RT のサイトにあった他の API も良いものがあったので機会があれば使ってみてみたいと思った。課題研究は今年で終わりなので機会があれば AI の制作を個人でしてみたいと思った。(河上)

まず、この課題研究をするにあたり、自分たちで 1 から Chat bot を制作するには、あまりにも知識不足だという事を思い知らされた。初めは、LINE で会話ができる Chat bot を作るため

に奮闘した。が、会話のキーワードに反応し解答をする膨大なデータベースの作成や、動作を確認する為のソフトの用意など、限られた時間の中で作るのはとても難しいものだった。最終的にはインターネット上にある AI を利用する形で Chat bot を完成させることが出来た。課題研究は今年で最後になるが、もっと知識を得て、個人的にこういったものを作っていきたい。(香川)

私は今年の課題研究でプログラムの方ではなく Live2D の作業をした。自分で Live2D を使う提案をしたのはいいが使う方法も分からない状態でのスタートだった。インターネットで調べて Live2DViewer と棒読みちゃんを使用する方法を見つけたが、それを Chat bot につなげる方法が分からずこの方法は間違っていると思った時もあった。しかし先生の的確なアドバイスや友達が教えてくれた事をする、最終的に喋って動く AI を作り上げる事が出来た。今回の実習は私の今後についてとてもいい経験になり、私の高校生活の中でとてもいい思い出になった。進学や就職した後も今回のようなものづくりをする機会があると思うのでその時は今回の課題研究で学んだことを活かして作業をしていきたい。(東)

6. 参考文献

- ・生放送等のコメント読み上げに合わせて Live2D モデルに口パクをさせる方法 (複数モデル対応)
<http://hatyati.jp/live2d-live/2/>
- ・A3RT Talk API
<https://a3rt.recruit-tech.co.jp/product/talkAPI/>
- ・棒読みちゃんのダウンロードページ
<http://chi.usamimi.info/Program/Application/Bo uyomiChan/>
- ・NETDUEETO β2 のダウンロードページ
<https://www.netduetto.net/download/>
- ・Live2D Viewer のダウンロードページ
<http://sites.cybernoids.jp/cubism2/tools/live2d-viewer>

・結月ゆかりの Live2D モデルのダウンロードページ(2018年10月31日サービス終了)

<http://library.live2d.com/Libraries/view/000149/>

・チャットボットとは-意味の解説 | IT トレンドの IT 用語集

<https://it-trend.jp/words/chatbot>