人工知能レポート

-人工無能(O-siri)の作成-

提出期限：平成 29 年 1 月 23 日

第一班

氏名　　：Calvin J H

小畠大輝

年藤捺紀

中山颯

百垣太一

1. 人工無能について

製品名：O-Siri

愛称：野獣先輩

性別：男(諸説有り)

年齢：24 歳

職業：学生

性格：根は優しく後輩想い

　　自分勝手になる節がある

2. 人工無能の特徴

「野獣先輩」は、とある動画投稿サイトに投稿された動画に出演した人物であり、ネットでは一躍有名となり現在も世間を騒がせている。この人物について世界中の学者が研究に研究を重ねているものの、未だその全貌は明らかとなっておらず、本名はおろか性別、素顔も解明されていないことから現代の技術を持ってしても何の情報も得られないため巷では実在しない人物だったのではないかと噂されている。

今回、私たちはそんな野獣先輩をモチーフに人工無能を開発した。野獣先輩の性別は不明ではあるものの、その風貌や言動から男性と仮定して話を進めていこうと思う。

彼は 24 歳の学生であり、性格に関しては後輩想いの優しい人物でしばしば「人間の鑑」と呼ばれることもあるほど出来た人間である。しかしながら自分勝手になる節も垣間見えるため、人によりけりといったところか。

発音・滑舌が悪くよくわからない日本語を使うこともあるが、そういった場合は適当に相槌を打っておくと勝手に話を進めていくので遠い目で見ていると良い。

3. 辞書について(年藤,小畠)

本システムにおいて、私たちは４つの辞書を作成し、多様なシナリオを展開できるようにした。

辞書作成の際には言葉選びを慎重に行い、用途を誤らないよう細心の注意を払った。辞書の内容は野獣先輩が動画内で発言したものをベースとし、それに様々なネットスラングを取り込んで構成した。また、弓削島内を舞台にした日常会話を元に作成した辞書も用意しているため、ネットスラングに疎い人でも十分に楽しめる内容になったのではないかと思う。

辞書の内容は、４つのうち２つは、ネットスラングをメインにした辞書。残りの２つは、弓削島内を舞台にした日常会話をメインにした辞書になっている。また、１つ１つの内容は短いものになっているので、手軽に楽しめるようになっている。

会話はまず野獣先輩の発言から始まり、対話形式で話題が展開されていく。辞書内に登録されていない言葉が返ってきた時は、シナリオ展開用の辞書とは別に例外処理用の辞書を呼び出すという方法で例外処理を行っている。例外処理の応答文は野獣先輩の機嫌によってランダムに変化する仕組みになっている。

4. 音声について（百垣）

5. プログラムについて（カルビン）

上記に述べたように，本プロジェクトは 2chan で流行っている“野獣先輩”というネタキャラクターをもととした人口無能であり，多様性を測るために 5 つの辞書を用いた人口無能である．４つの辞書の中身は普段の会話の辞書で，もう 1 つの辞書は入力ミスや前の４つの辞書に登録されていない返答が返ってくるとき（例外処理）のエントリーが含まれている辞書である．我々が目指している人口無能の設定とその流れをこれから記述する．

設定は，ユーザがあるチャットルームで“野獣先輩”と出会って会話を開始するというものである．“野獣先輩”は最初に“あなたは誰？”という挨拶から会話を始める．これに対して，ユーザが自分の名前を入力し、入力した内容がユーザ名となる．そして，事前に用意した先ほどの４つの辞書を用い，ランダムに選択された“野獣先輩”はまず、“ほお、＜ユーザ名＞，＜辞書の最初のエントリ＞”をユーザに送信する．この返事に対し，ユーザは会話を自然に“野獣先輩”が思う通りに継続しなければならない．つまり，辞書に登録されているエントリーの流れに従うということである．もし，ユーザが自然な返事をしなかったら，“野獣先輩”が起こりはじめ，怒りを込めた返答をし始める．自然ではないレスポンスをユーザが継続していくと，徐々に“野獣先輩”の返事が悪化していき，最終的に非常に不良な言葉で返答するようになる．野獣先輩が怒りモードに入る前の“野獣先輩”の発言にきちんと返事すれば（辞書に返事の対応が含まれている場合），会話が進んでいくようになり，“野獣先輩”の怒りゲージが減っていく．“野獣先輩”が飽きたら（辞書の最後のエントリにたどり着く場合），会話が終了し，”リスタート?”というレスポンスが返る。それに対して“Yes”などとの同意の言葉を送信すると会話をもう一度やり直すことができる．会話の途中で，ユーザが“bye”と送信すると，“野獣先輩”が“元気でな。”と挨拶を言い，会話を終了させる．

この設定を実現するにあたり，我々が作成した人口無能は大きく分けて二つの部分に分類される．一つ目は Kivy を用いた GUI の作成。二つ目は Python による対話処理である．今回のプロジェクトでは，我々は他チームとは全く別の方法でプログラムの作成を行い，使用言語は Java ではなく，Python を使用する．Python を選択した理由はいくつかあるがその中でも 4 つ，以下に示す．

- 他の言語と比較した際，比較的に簡単にプログラムのフィーチャーや機能などを実現することができる．
- すでにあるプログラムに制限されず，開発の際，自分たちの実現したい機能に集中することができるため、自由度が高くなる．

- GUI を作成するつもりであったため，比較的に早く開発ができる Python の GUI ライブラリ（Kivy）を使用したい．
- 自分(Calvin,中山)の Python を用いた開発能力の向上につながる．

GUI を作成するために，本プロジェクトは Kivy という Python の GUI ライブラリを用いる．このライブラリは，マルチプラトフォームで，モバイル型端末にも対応できるという利点がある．

4.1 Kivy GUI について(中山颯)

Kivy とはのマルチタッチアプリケーション開発のためのオープンソースライブラリである。画面構成を図 4.1.1、図 4,1,2 に示す。

4.1.1 Chatroom

4.1.2 Setting Panel

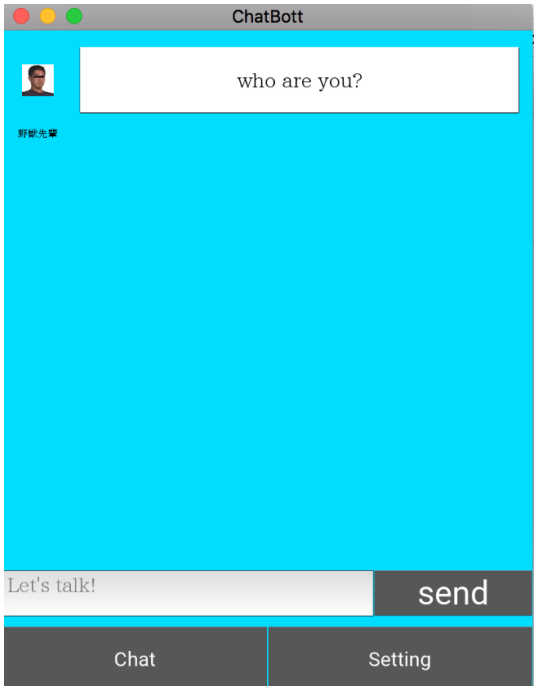


図 4.1.1　Chatroom

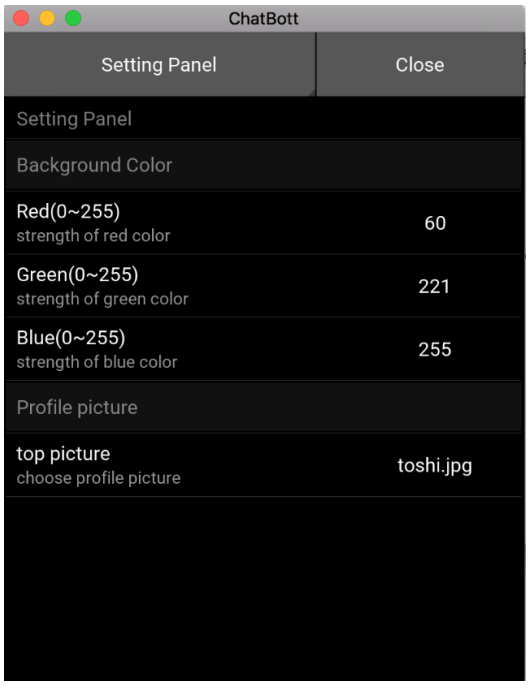


図 4.1.2　Setting Panel

次にそれぞれの画面の機能について説明する。

4.1.1Chatroom

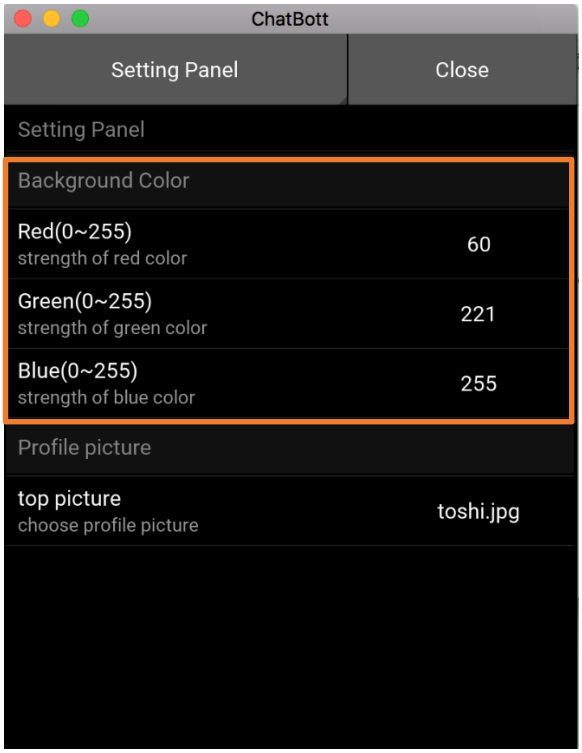
Chatroom では ChatBott（野獣先輩）との会話の内容を表示する。コンソール上の味気ない会話よりもよりリアリティーの高い野獣先輩とのお話を楽しむことができ、まるで野獣先輩とチャットしているように錯覚しかける。

4.1.2 Setting Panel

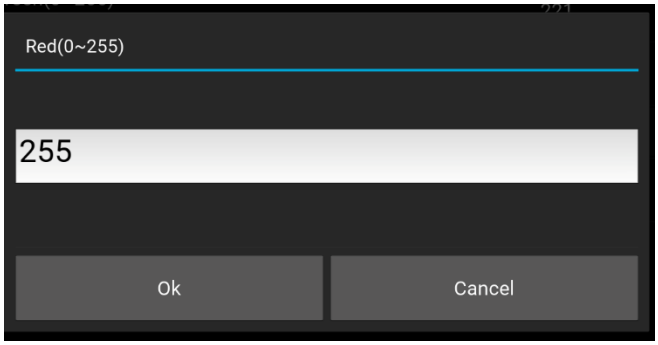
Setting Panel では Chatroom の背景色と Top picture を変更することができる。

1. 背景色の変更

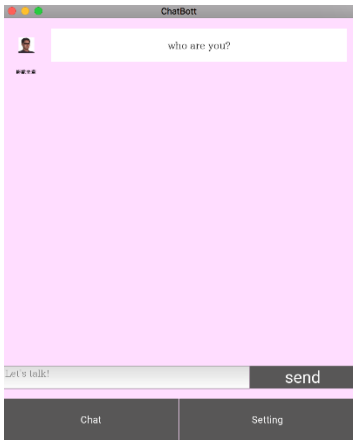
Setting Panel を開き RGB の値を選択することで背景色を変更することができる。まず Setting Panel 画面を表示し、



Setting Panel の background Color のセクションを選択すると

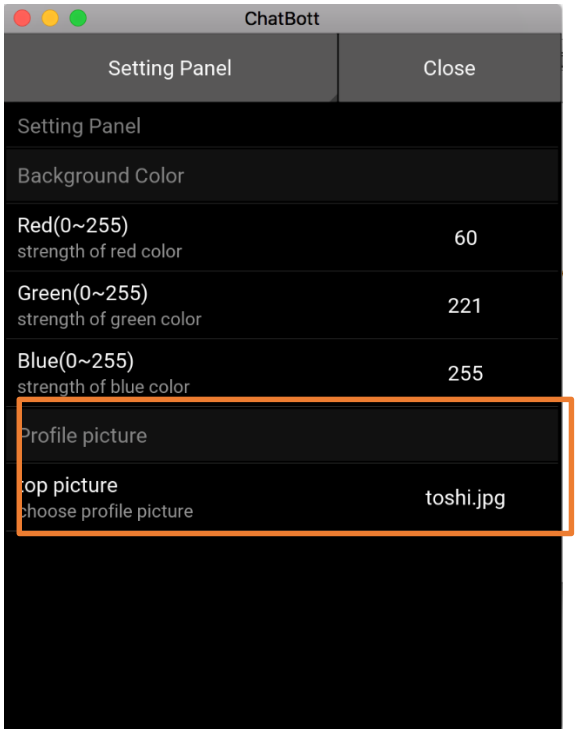


このような画面が表示されるので RGB の値を変更することで背景色を変更することができる。背景色の RGB を 255,221,255 のように変更すると以下のような背景色になる。

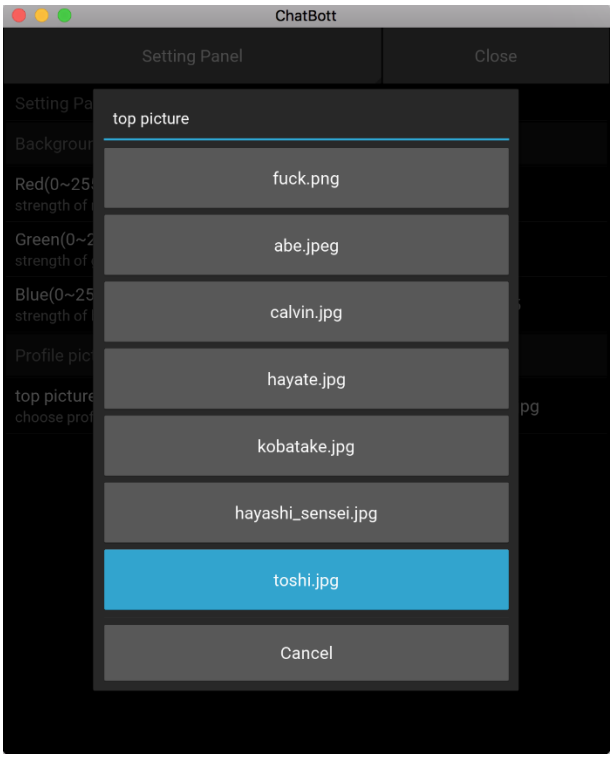


2. Top Picture の変更

Setting Panel から自分の Top Picture を変更することができる。まず Setting Panel を開き、



Setting Panel の Top Picture を選択すると、以下のような画面が表示される。



画像を選択すると、

このように自分の Top Picture を変更することができる。

4.2 対話処理（カルビン）

Python で対話処理を作成するため，あらかじめ用意された Java プログラムを使用することができず，そのプログラムの機能を完全に実装し直す必要があった．Java のプログラムを分析した結果，主に 3 つの大きな処理に分けられると考えた．これらの処理をリストアップすると以下となる．

- 辞書を読み込む処理
- 辞書を参考にし，ユーザの返事に対応する処理
- ユーザに対する返答を出力し，音声を流す処理

最初の 2 つの処理は Python の標準ライブラリで実現できるが，音声を流す部分を実現するために，Python の標準ライブラリの上に以下のライブラリを導入した．

a. pykakasi

このライブラリは，KAKASI という言語フィルタを Python で使用できるようにしたライブラリのことである．KAKASI（Kanji Kana Simple Inverter）は，漢字，ひらがな，カタカナのいずれかをひらがな，カタカナ，ローマ字のいずれかに変換する言語フィルタのことである．このライブラリを使用すると，辞書の中に書かれているエントリーをローマ字（訓令式ローマ字）に変換することができるため，辞書は漢字，ひらがな，カタカナで書いても音声ローマ字で書かれている音声ファイルにも対応ができるようになる．

b. pygame

pygame とは，従来ゲームの開発を支援するライブラリのことである．しかし，そのパッケージの中で，音声ファイルを非同期に流す処理が含まれており，この処理を使用すると，遅延なしで音声ファイルを再生することができる．

次に、本プログラムのフローチャートを以下の図に表す.

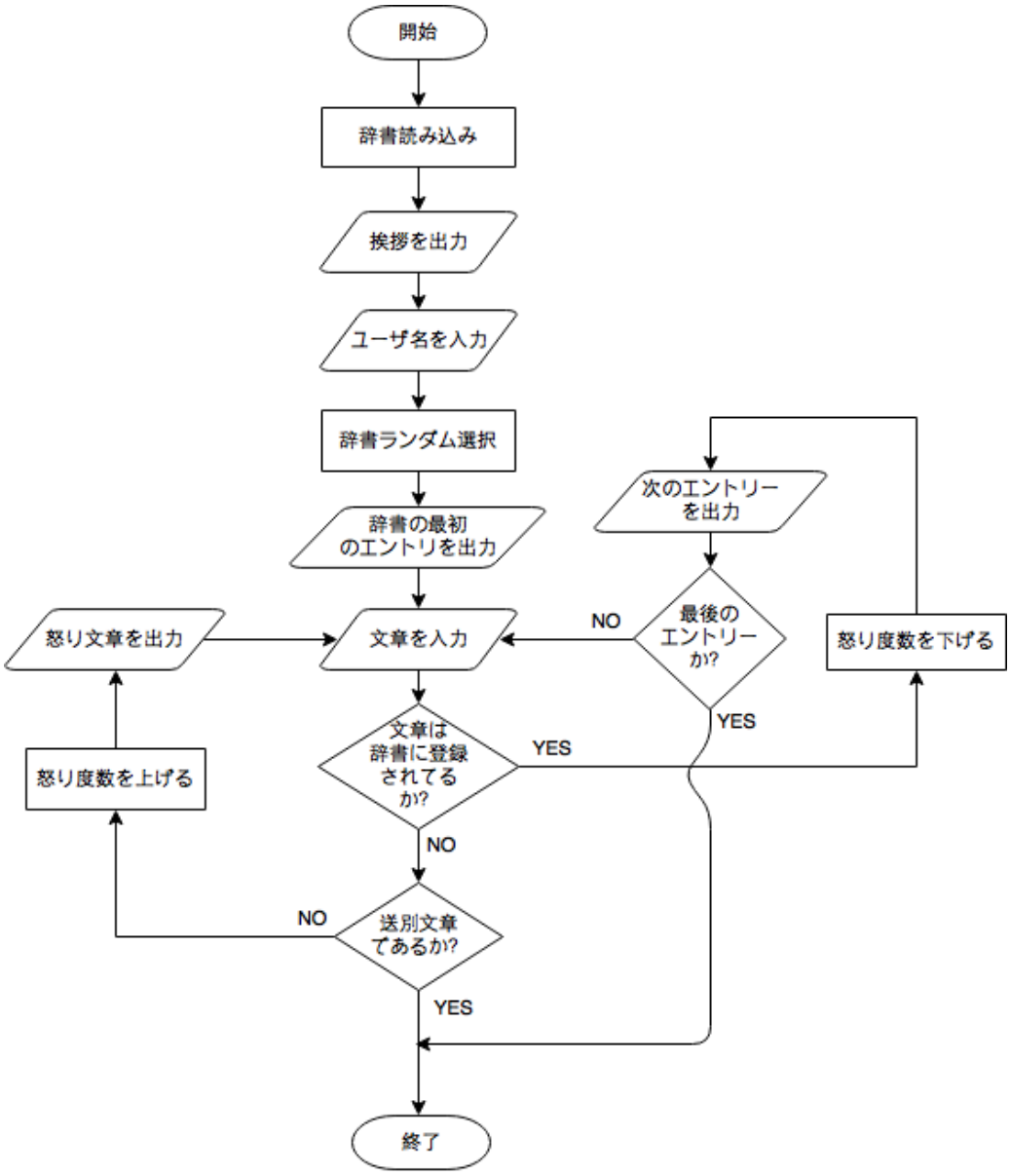


図 4.2.1 プログラムのフローチャート

最初に，用意された辞書を読み込み，各エントリをリストに入れる．辞書の一つのエントリでは，現在の ID，次の ID，出力文章，入力文章の４つの要素が入っており，その間をタブで分割する．辞書を読み込んだ後，挨拶を出力し，名前を問う．ユーザは名前を入力し，その名前を今回のセッションのユーザ名となる．

次に，リストからランダムに一つの辞書を選択し，それ以降はその辞書にしたがって

対話を進める．そして，その辞書の最初の出力文章を出力し，現在の連鎖番号をその ID とする．ユーザがこの出力文章に対し文章を入力し，現在の連鎖番号の中の入力文章の可能性の中に含まれていたら，怒り度数を下げながら，次のエントリーの ID 番号に変換し，その番号の出力文章を出力する．そして会話が継続できるように，この処理を繰り返す．もし，入力文章はその可能性に含まれていないのであれば，怒り度数を上げ，怒り度数に合う文章を出力する．現在の連鎖番号の中にある可能性のある入力文章が答えられるまでこの処理を繰り返す．

会話を終了させる方法が二つ存在する．一つ目は，ユーザが辞書の最後のエントリーまでたどり着くときである．二つ目は，ユーザが終了コマンドを送信するときである．この時，“元気でな。”という文章を出力する．

本プログラムは，文章を出力すると同時に，文章に合わせた音声ファイルを流す．しかし，音声ファイルが存在しない場合（この場合では怒りの文章），そのエントリーの文章を上記に述べた pykakasi というライブラリを用い，ローマ字に変換する．変換した後，そのローマ字の文章をあらかじめ録音した訓令ローマ字の音素ごとに合わせ，単純にその音素の音声ファイルを連続的に再生していく．上記にも述べたように，音声ファイルは pygame のオーディオ処理機能で再生する．

6. 動作例(カルビン)

ここでは，いくつかの動作例を挙げる．最初に，普通に会話が流れるときのスクリーンショットを以下の図に示す．

図 4.2.2 普段の会話（前半）

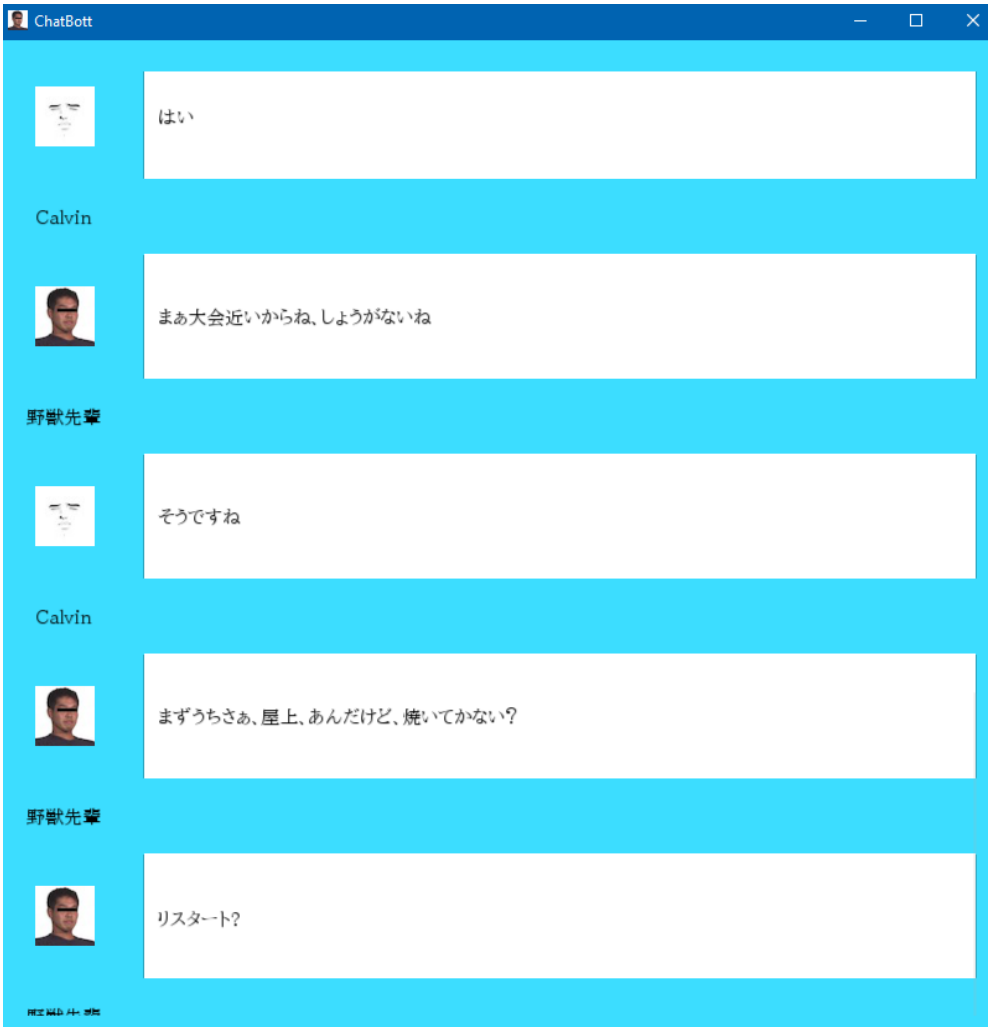


図 4.2.3 普段の会話（後半）

これらの図に示す会話の流れは，１番目辞書（Dictionary 1）が選択され，辞書に登録されている返事をした場合である．会話が終了するときに，通常に“リスタート？”が返答されることがわかる．これに“yes”と返事すると，再開される．この状態を以下の図 4.2.4 に示す．

図 4.2.4 会話を再開する

そして，会話の途中で辞書に登録されていない返答をする場合，怒り度数を増加させ，それに合わせたレスポンスを返す例を挙げる．この動作例を以下の図 4.2.5,図 4.2.6 に表す．また，デバッグ用スクリーンを示す図？？の“angry level”と“angry percentage”に注目すると，怒り度数が通常に上昇していることが分かる．

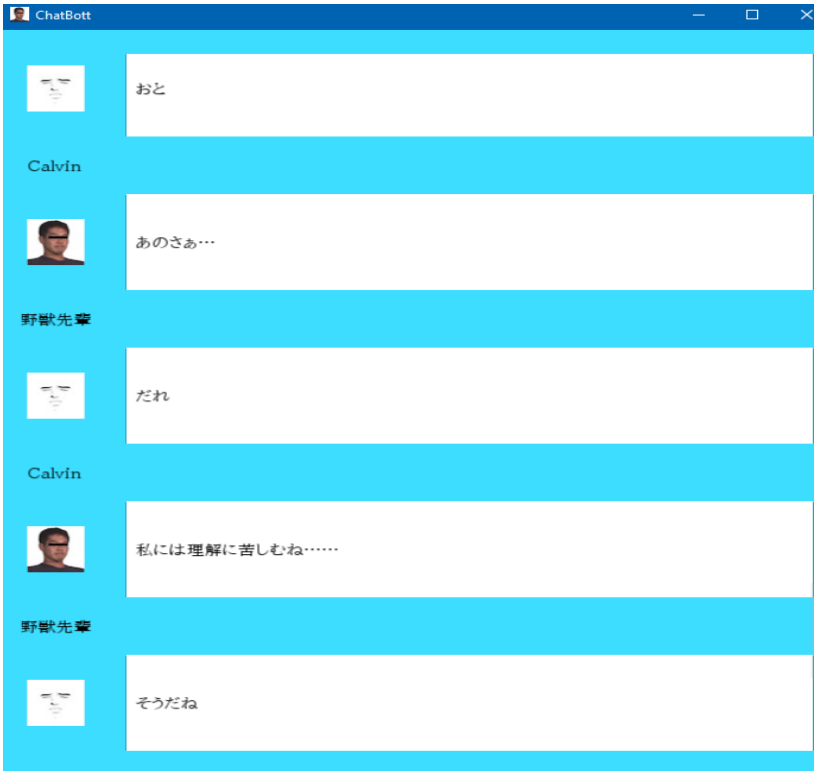


図 4.2.5 怒っている様子

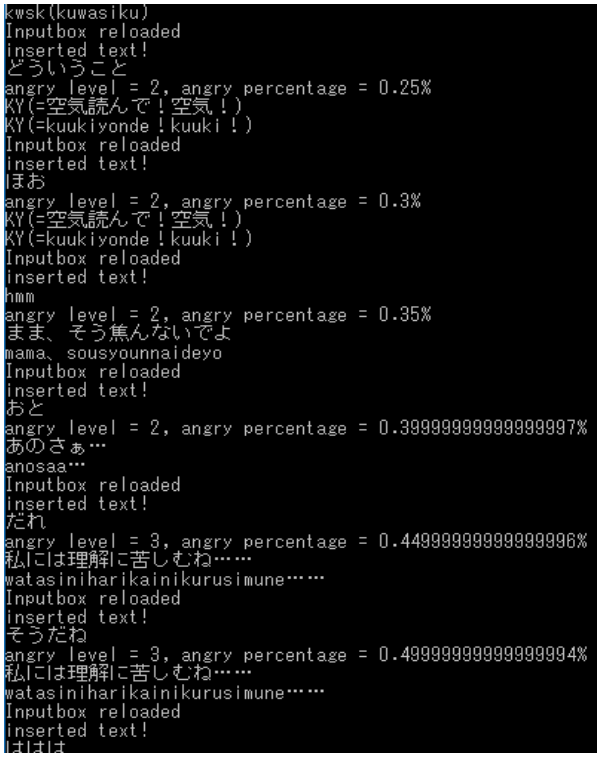


図 4.26 怒っている様子（デバッグ）

最後に，会話の途中で，“bye”などのような送別言葉を入力すると，会話が終了し，“元気でな。”という言葉が返ってくる．この様子を次の図 4.2.7 に示す．

図 4.2.7 会話を強制的に終了させる場合

7. 考察

7.1.自分たちの作品について（中山颯）

本人口無能の開発にあたって一番大切にしたことは自分たちが本当に面白いと思うものを作ることであった。その題材として野獣先輩は最適なものであったのではないかと思う。また、開発メンバーも田房研究室と長尾研究室の開発メンバーがそろっていた。そのため今までの Java で構築されていた人工知能システムを Python で再現し Kivy を用いた GUI 開発、さらに windows 上で実行するための exe ファイルまで作成を行うという他の班とは一線を画すものとなった。

また、昨年度までの作品と比較しても人工知能として野獣先輩を使うという着眼点、野獣先輩のネタの引き出しの多さ、システムの完成度いずれにしても負けているものではないと自負している。特に野獣先輩の求めている発言がない場合や会話が続かない場

合に野獣先輩の怒りポイントが上昇していくシステムはいい発想だと思う。
しかし、システムの特徴として会話内容を把握していないと会話が続かないのでその点を改善する必要がある。

7.2.ほかのグループの作品について

8. 今後の課題

（カルビン、中山颯）

本プロジェクトでは，マルコフ連鎖を利用したかったが，時間の都合上や辞書のスケールの都合上応用できなかった．今後の課題としては，形態素解析を用い，ユーザが入力した文章をいくつかの部分に分け，それらの部分からユーザが言いたいことを分析する．分析した文章から，辞書から意味を推測し，マルコフ連鎖で確率的に返事を返すという機能を実現していきたい．

また，ユーザの入力文章が辞書に登録されていない場合，動的に対応できるように辞書の自動登録を行いたい．登録されていない文章が入力されたら，人口無能はそれに対し質問をし，意味をユーザに問う．これを行うことにより，意味と一緒にキーワードを辞書に登録でき、さらに効率のいい人工知能システムになると推測される．

現在では，音声ファイルが存在しない場合は音声合成を用い自動的にスピーチ音声を発生させるが，音素を連続的に再生する際，不自然な音声になってしまう。さらに自然に音声合成を行うために，様々なアクセントやイントネーションで発言した音素を録音し，文章に合わせたアクセントでスピーチを発生させたい．これと同時に，pykakasi ライブラリが漢字を音素に変換するときに正しく変換されない可能性がある（例：入って→いつつて）ため，今後の課題とする．そして，現在 O-siri は日本語にしか対応していないため，今後ローマ字スペリングに対応していきたいと思っている．

また、GUI についても日本語入力の際 Enter を押すまで入力文字が表示されないバグの修正、背景色だけでなく背景画像の選択など、デザインを変更する機能の追加や会話履歴の作成などまだまだ改善点がある。

9. 感想

ネットスラングを使った会話を作成するということだったので、辞書を作成する際に言葉選びに苦労した。しかし、人工無能がどういうものなのかがわかり、人工無能に興味が持てたことが良かったと思う。さらに、このプロジェクトによって、チームワークの大切さやグループワークの良さについて学ぶことができたので、非常に良かったと思う。（小畠）

初めてこのような人工知能技術を用いたプログラムを作成し，前知らなかった様々な知識が知るようになり，非常にたくさんのことを勉強できたと思う．最初に開発を始めたときに，自分の中でいろいろなアイディアを実現したかったが，一から開発するということで困難であることが分かった．また，スケジュール管理があまりできていないことも一つの原因といえる．そのため，今回のプロジェクトを参考にし、次のプロジェクトのスケジュール管理をもっと効率的にやっていきたいと思っている．しかし，よかったと思ったことが一つで，自分が持っている Python の知識を実現できたので，よかったと思う．最後に，このプロジェクトにわたって，ノウホウやみんなとのチームワークの大切を学ぶことができたと思うので，非常に良かったと思う．（カルビン）

まず、チームリーダとしてこの人工知能のプロジェクトを何とか形にできたことを非常にうれしく思っている。昨年までの人工知能プロジェクトと違い、Python での開発および GUI 化という莫大なタスクを短い期間で達成しなければならなかった。そのためにチームが一丸となり、支えあいながらこのプロジェクトを完成させたことは非常に良い経験となった。今後の開発人生にもこの経験を活かすことができるようこれからも精進していきたいと思う。（中山颯）