Artificial Intelligence: Free Word Association

Group Project, CMP2089M: grp 13

A. Parker, D. Smith, G. Onoufriou, J. Trigg, J. Liou, T. Sun,

2016

Contents

[Abstract 2](#_Toc448404022)

[Introduction 2](#_Toc448404023)

[In our proposal 2](#_Toc448404024)

[In the making 2](#_Toc448404025)

[Criteria 3](#_Toc448404026)

[Summary of Criteria Justification 4](#_Toc448404027)

[Design & Reasoning 5](#_Toc448404028)

[architecture 5](#_Toc448404029)

[abstract data type specifications 5](#_Toc448404030)

[User Interface 6](#_Toc448404031)

[Summary of Design 7](#_Toc448404032)

[Tools and Methodologies 9](#_Toc448404033)

[Critical reflection: 11](#_Toc448404034)

[Honest Appraisal 11](#_Toc448404035)

[Teamwork 11](#_Toc448404036)

[Belbin role utilisation 11](#_Toc448404037)

[Group Members 12](#_Toc448404038)

[George Onoufriou 12](#_Toc448404039)

[Jichan Liu (Ken) 12](#_Toc448404040)

[Tao Sun 12](#_Toc448404041)

[Jake Trigg 13](#_Toc448404042)

[Alex Parker 13](#_Toc448404043)

[Damon Smith 13](#_Toc448404044)

[Records 14](#_Toc448404045)

[Bibliography 15](#_Toc448404046)

[Appendix 16](#_Toc448404047)

[One 16](#_Toc448404048)

[Two 17](#_Toc448404049)

# Abstract

Artificial Intelligence.

What is Artificial Intelligence, and how do we distinguish it from traditional, organic, intelligence. This, at heart, is our core topic of research in this project. We, six members, decided to tackle this problem, using an old psychology technique. Free word association. (Jung, 1910)

We were tasked, with passing the Turing test, or a type thereof. How we chose to tackle this problem, and creating its necessary criteria, was left up to us.

In this report, we critically assess our methods, throughout, detailed in each relevant section, summarising them, later.

# Introduction

## In our proposal

We proposed, that, the most applicable way, for us to pass the Turing test, was to create a, free word association, AI. Our reasoning can be summarized in the following:

* Since our AI, represents an entity pretending to be a human, it needs to be able to interact with as many people as possible. Free word association, is a simple concept that most can understand, abbreviated to, answer as fast as possible, the first word that, occurs in thought. (Jung, 1910)
* We have limited experience in creating AIs. And we are tasked with passing the Turing test. We believed we would have greater success, with a simpler version of natural language processing, limited to one word. Free word association however, still allows us to portray distinct thought patterns. It also allows us to, more easily implement personas.
* Personas, were a key feature of our proposal. The University of Reading (2014), had many personified AIs. We chose to follow the example of Eugene Goostman, who passed the Turing test, against 30 experts. According to their documentation, the persona played a large role in the believability of the AI, instead of just being a generic human. The AI had its own, unique quirks. Since we had a limited time period, to implement such advanced features, we would need to, use a more simplified version of the Turing test. Free word association was perfect for this role.
* Free word association, has been used extensively in the past, for patients, to project their thoughts and emotions, in a manner that, can be easily extracted. (Jung, 1910)

We examined the literary evidence to provide us with the base we could then build upon. We then set out a time plan, using Gantt charts, of how long we expected each of the aspects of our AI were going to take.

## In the making

As our project developed, we learnt that some areas, took longer and others, too shorter amounts of time, than expected. We did assess in our risk matrix that this was a likely scenario, as we had at the time of starting, not had experience in creating such systems, so the timelines were best estimates. This will impact us in the future, as we now better understand, the complexities of both AIs and of working in teams, on (relatively), large projects. This means we will be able to, more accurately estimate, time spans, which are required to do certain tasks. Even with this difference of expectation, we created an AI system, which can meet many of our initial, and further developed criteria.

# Criteria

In our project proposal, our initial fulfilment criteria (subsequently goals) were:

1. Model AI composition (Hanheide, 2016)
   1. Knowledge Base
   2. Inference Engine
   3. Interface
2. “Full set” of English words
3. Relations between words
   1. Weighted
4. Persona Capabilities
5. Convincing >30% of users
   1. Believable response
   2. Delays
   3. Mistakes

As our project developed we tweaked the above criteria of success. Primarily the “softer” criterion (2, 4 & 5). The main causes, reasons, and results, are summarised as follows:

* Criterion 2; “Full set” of English words. We discussed in length about getting the information necessary for a “full set” of English words. We felt that to represent every, single, English word that could possibly be written, and its connections to every other word, was both unreasonable and unnecessary. Therefore needing to be changed, from our initial plans.
  + The subsequent adjustments:
    - Getting a smaller, reliable base data set. We reasoned that creating a system that could get any word from the internet, and interpret it, in a meaningful and reliable manner was unreasonable. We decided to go with a set, tested database which we could use as our reliable starting point.
  + The results:
    - What we found as a result astounded not only ourselves, but all manners of people who learned of this amazing resource. The database we discovered had several different facets, appendixes, ranging from advanced statistical testing of word relations, to idiosyncratic responses. (Nelson et al. 1998)
    - We had more than enough data to satisfy our second criterion, this data also advanced our third criterion to success. The relations between words, so explicitly tested that we as a result of this small change to mind set, and a stroke of luck to discover this ancient gem, completed two criterion in one.
* Criterion 4; we reasoned that simply creating a persona capable AI was not representative enough of what we wanted to achieve. We felt that not only does the AI need to be believable in terms of being human but arguably, just as importantly, it had to have a convincing personality. Just like Eugene Goostman. (Reading, 2014)
  + The subsequent adjustments:
    - Added more factors into persona word choices; literacy, daily experience relation chains. We would use our custom database management system, with several calls, to find if any words from what the user said and what the persona would be most likely to relate, overlap. This means we could give higher likelihood to words expected for this persona.
  + The results:
    - This could have been the most inventive, and crowning believability factor adjustment. But at the time of writing this report only a basic version has been implemented due to time constraints. But we look forwards to the final implementation, ready for the demonstration.
    - This basic adjustment, has made a large difference for some words likelihood, impressing many passing users, who have tried our system. The more advanced version will hopefully improve on the amazing work that has already been done to implement this feature. Serving as its groundwork.
* Criterion 5; we found through testing, sometimes, for more recent words that we would occasionally not have the directly relevant data. This hampered the believability of the program, as the best solution to the problem, in the original state, was to use the most generic words in the database. We deemed this to be ineffective, if used repeatedly. We needed to add convincing responses, which change over time, to more successfully convince users.
  + The subsequent adjustments:
    - A clever use of our AI’s memory. If for example the user responds to the AI with a word it has not seen before, it knows that, the user's word is somehow related to its last word. It uses its memory, and instead looks up its own last word. Then uses that last word as its association, while removing the word the user just gave, from its possible responses.
  + The results:
    - This gave us a now complete and bullet proof system. We now had the dynamic where, if the first word the user gave was unknown, that it could use generic terms. But as the game progresses the AI learns from both its past and the user’s responses to give the most, believable responses possible. Speeding up, just as we humans did, as the game is underway. Based off of our relatively small, but in depth data set.

### Summary of Criteria Justification

Throughout the process, of creating our AI system and algorithms, we critically assessed our methods. Improving our system together. Our criteria were not perfect upon proposal. But now with the adjustments of our criteria, and assumptions, we have a better understanding of both, what our aims are, and how we have achieved those aims. We would have liked to have undertaken, fully fledged, user testing. This was however, likely an unrealistic expectation, due to everyone's individual commitments, but would have been plausible, if considered early on in the project.

#### Where we succeeded in meeting our criteria:

* (Criterion: 1) We have undoubtedly, succeeded in creating a model AI system.
* (Criterion: 2) We have a “full set” of English words, taking into consideration that, a full set may not necessarily be, a (full) dictionary of words, but instead, enough words to allow our AI to be, believable. Just as humans do not know every word, and its meaning.
* (Criterion: 3) We have a highly effective, sophisticated, dataset. We have manipulated our base data, and abstracted, the useful information. We have successfully created, a clever weighting system, which combined with our randomness factors, leads to realistic and sometimes quirky responses.
* (Criterion: 4) We have created, a persona based system, that changes according to several distinct factors. Things such as literary ability is taken into consideration.

#### What we would need to improve, to better meet our criteria:

* (Criterion: 5) User studies to better quantify our results, especially with passing 30% success mark.

# Design & Reasoning

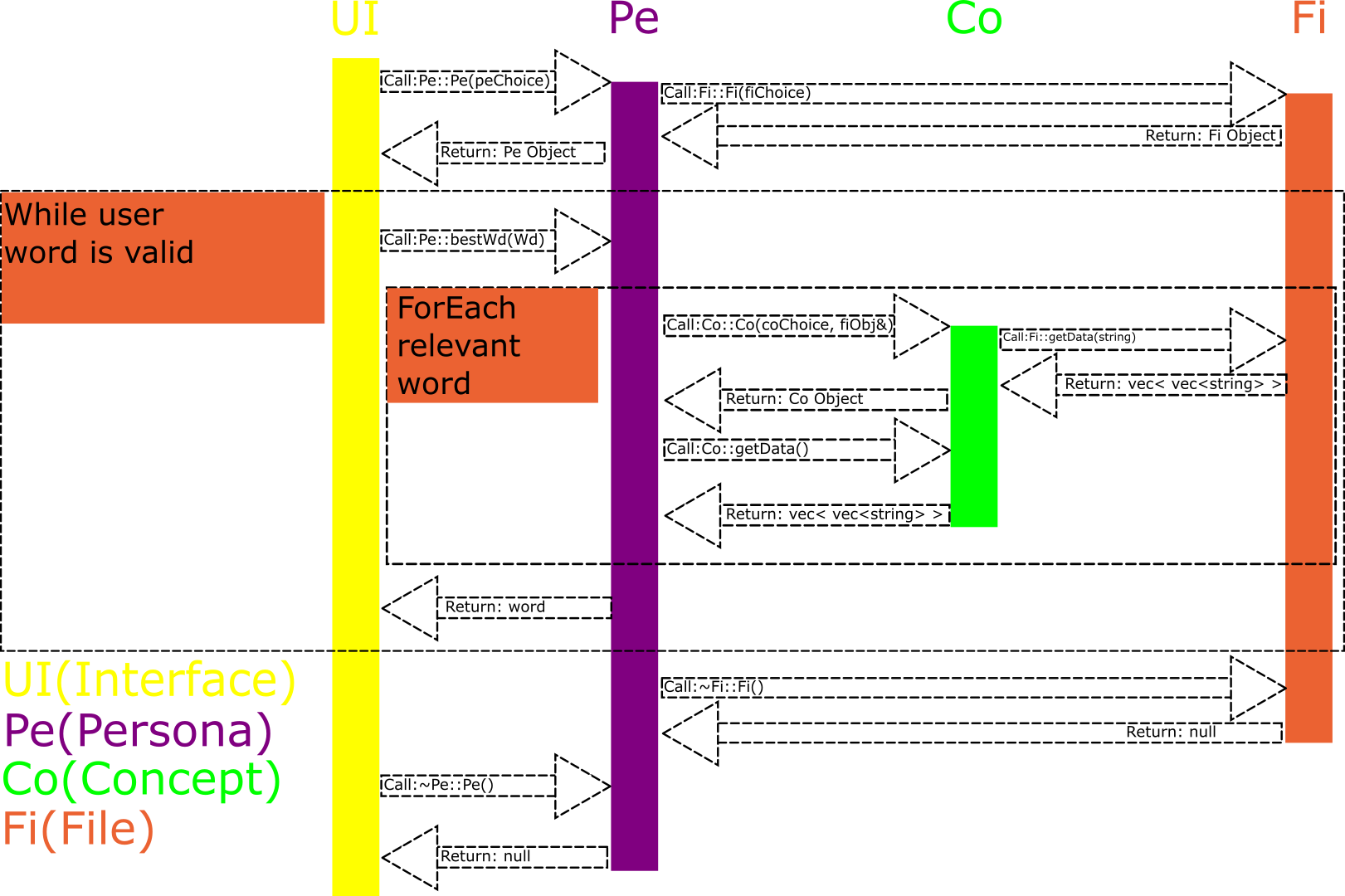


Figure 1: Initial Database Architecture

### architecture

Since, the group decided to use Object Orientated Programming (OOP) languages, we abstracted the system into objects. Through abstraction we found the main components of the system needed to be:

* User Interface - Needed interface for the user, to better assess the AI
* Persona (human factors) - Needed to personify responses and chain (as proposal)
* Concept (individual concepts) - Needed to frame individual ideas for use (OOP)
* File (files) - Needed to manage file data to be more accessible

Based, off of the above necessary components, we linked them together in a manner, which would not require processor intensive, reconstruction of the objects, regularly. The exception is Concept, as multiple instances of Concept would be necessary. Examples of multiple Concepts are, when given a user word that a second word/ concept is pulled to compare. This includes comparing against Persona relations; Bob, is a persona. Bob is a builder. The database is searched for “builder” the concept. The results are compared against the results from the user's word.

### abstract data type specifications

We initially intended to use ADTs, but we found problems in that, not all members were familiar with them. We did begin solving this problem, by helping members understand ADTs, but instead we opted for class diagrams, as they were something the group was already comfortable with. This did also allow for the individual programmers to be creative, and subsequently be more enthusiastic about each section, as the specifics of function semantics is left up to the programmers. This lack of a formal definition, did have its own drawbacks.

#### The main drawbacks of not using ADTs are:

* Others are less familiar with the intricacies of each individual function, and what each one is meant to do. This can lead to some confusion.
* More difficult to debug and test, as you do not have the specification to refer to, if something seems out of place.
* More difficult to enforce, good programming practices, such as minimal, global scoping, since there were less restrictions, without formal definitions.

#### The main advantages we gained from using Diagrams:

* Not as temporarily expensive, getting members to define ADTs, and ensuring their definitions are correct.
* Feeling of freedom, and achievement of creating something, rather than implementing someone else's code.
* No need to spend time ensuring, members understand ADT specifications.
* Quickly, go into the productive stage, to produce the artefact that we have a restricted time to make, and write about.

### User Interface

#### Plans

We initially, planned in an abstract manner. Which we developed into a fit for purpose, user interface.

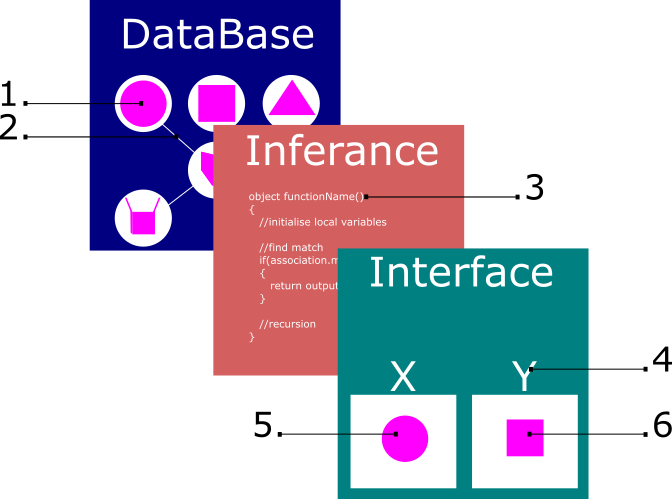


Figure 2: Initial abstract plans, of the AI system.

Figure 2 shows how we abstracted the components. 4 being the two personas X (Player), and Y (AI). 5 & 6 represent responses, such as square when given circle. This functioned as the base, which we then developed.

Since we had already planned in Figure 1, to link the Interface using set functions, we had no need to factor this in to this preliminary design of the interface.

#### Current





Figure 3: Current User Interface.

The team developed the initial abstract design, into Figure 3 above. Please note that, this is it in the current state. But we do have plans to develop this further before the demonstration date. Especially in regards to the second persona displaying, and adding certain game conditions for draws.

Figure 3 shows the prominent persona layout, to be at the very head, so as to be instantly seen. We have a cascading reply system, so it zig zags between responses. Flowing better on the page.

#### Successes

* The user interface, contains all the necessary information, to begin to pressure users. The timer, functions to keep people replying, as fast as possible. Improving the overall game, as less time is given to input esoteric terminology.
* The user interface, sanitises user input, especially on the log in page, preventing important details from being missing, and as correct as possible. This ensures that the user can compare, their attributes, to those of the persona they are playing against, after the final UI changes before the demonstration. Although please note Figure 3, does not how opponents persona currently.
* Lists are adjustable, so as to allow the game to carry on, in particularly quick situations.
* Errorless UI system, which will always elegantly exit the application.
* Relative response delays, to make the responses believable, harder responses taking longer.

#### Improvements

* Finding ways to make the page less, cognitively expensive. To emphasize the important details, such as time remaining, to further user pressure, and speed.
* Backdoor to access free word association game immediately, during development.
* Pressing enter on log in page, should be bound to clicking the start menu button for user ease.

### Data

We began with a very convoluted and inconsistent data format. The data that was within was invaluable to us, but the formatting and consistency of it was a draw back. To solve this problem we developed our own data set, which was far more concise and applicable for our needs.

Figure 4: Original Data Set. (Nelson Et Al, 1998)

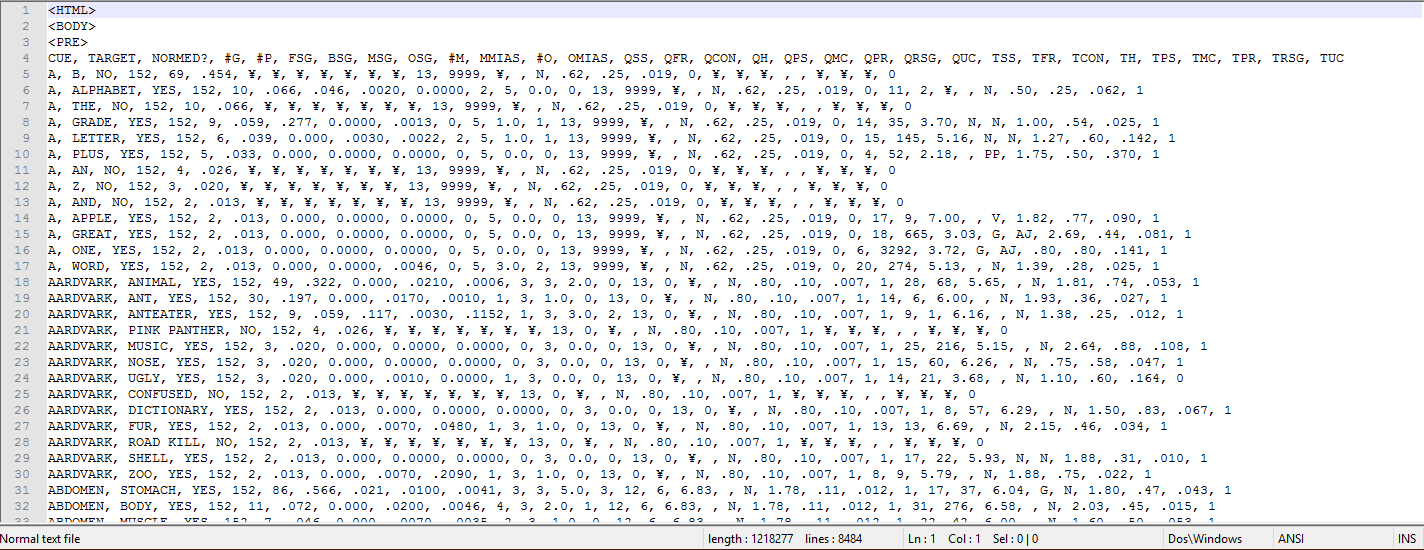


Figure 4 shows how the initial data set, had peculiarities, and anomalies that would later need to be removed, to ensure the data remained useful to us.

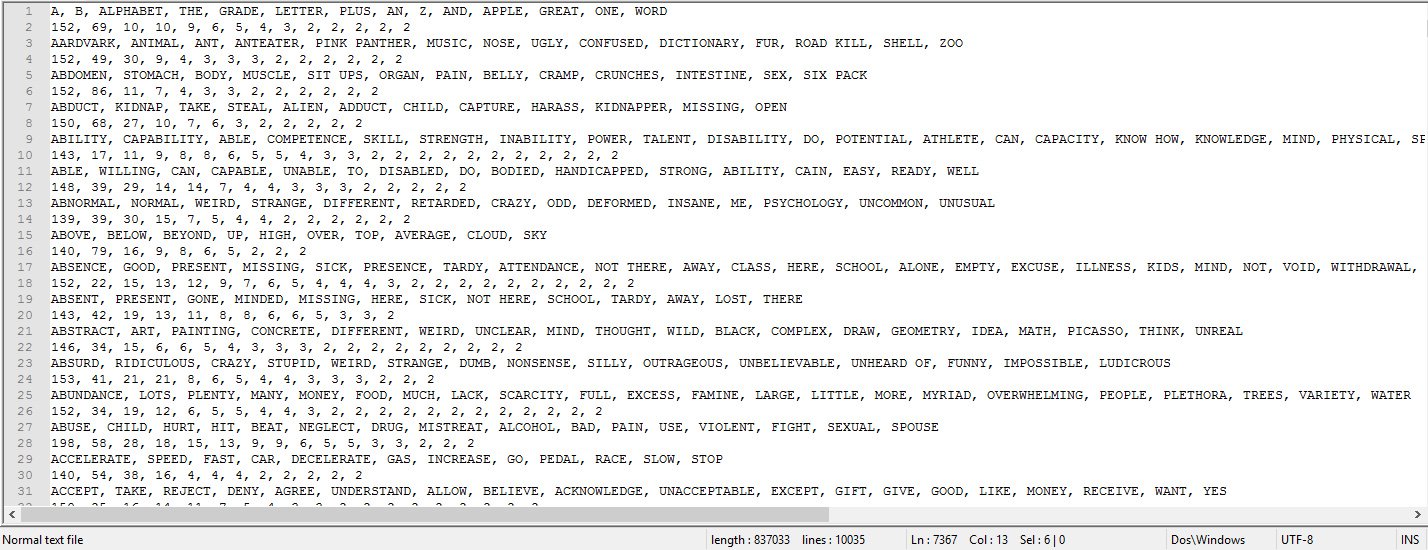


Figure 5: Manipulated Data Set.

Figure 5 shows the resultant data set, modelled after the Google hash code challenge data files. This format is far more consistent and concise than the original dataset. We cut down the data from all 7 files or roughly 1 million+ lines each, into a single file of 837,033 lines. This directly decreased the amount of memory required to hold the same amount of data, and it lead to a more efficient read/ write process. Putting less strain on whatever machine is running the application.

### Summary of Design

Critically looking back, it would have been preferential to have used ADTs. They allow for more coherent design, and easier testing for any bugs that are created.

In future, assuming that most members are comfortable with ADT specifications, we would use them. In our circumstances, due to the lack of familiarity with ADTs at the time, the two choices were either, sit down as a group and go through, making sure everyone’s knowledge is up to date. Or we could use class and other such diagrams which the group is already comfortable with. We did chose the latter, but where possible, ADTs are preferable.

The program overall is efficient. It completes every operation in a timely manner and with no errors. It could have been optimised in certain places, including multi-threading, to make the operation smoother and faster. But, for the most part, there were no flaws in the design that needed, amendment.

The User interface functions effectively, portraying all the information that is necessary, in a useful manner. Always exiting elegantly and dynamic enough to handle both large words, and long lists.

The group worked well, with a well-orchestrated plan to link all the individual sections together using a predefined interface (Figure 1). There were some issues with certain empty returns, for say words that do not exist in the database, but these were dealt with early on. This was solved primarily because of an initially limited G Appendix (main data file), which forced the program to run clean, with empty sequences.

# Tools and Methodologies

From the first meeting, the need for communication was identified as key for a successful project. We decided on the use of the University Email system, as our main line of official communication. With the Emails, the team could easily communicate with each other, as either a whole, or as individuals, enabling the sharing of important updates and issues. This was particularly helpful during the Easter break, with the lack of group meetings due to members of the team returning home; The Email system allowed for clarifications and deadlines for smaller targets rather than relying on a large target set for the end of Easter. Furthermore, these deadlines could easily be altered for any changing circumstances, and unforeseen issues that developed over the break.

The need for a less formal route of communication was also called for; Where Emails would be used to arrange meetings and share key information, we also used WhatsApp for short term reminders and brief updates. This allowed group members to communicate information too brief for an Email with WhatsApp, and too in depth for a message with Email. The format of WhatsApp allowed for a more back-and-forth approach to communication, speeding up smaller discussions like questioning which room a meeting was in, which can be quickly answered through the App.

The development of the Artefact itself began with the creation of a GitHub repository. We decided that using Git for version control would be very beneficial, enabling us to make changes to code, without the fear of creating bugs and risking the long term stability of the project. Any damaging change could be reverted back, and the use of GitHub to store our code enabled the whole group to work on the project, without interfering with the others. Furthermore, the recording of individual commits let the group see exactly what changes had been made, and when features had been added. GitHub notably helped during the development of the Artefact over Easter, enabling the team to view other members’ up to date code, and troubleshoot issues with how the classes interacted. Once the classes were mostly complete, GitHub let the team make changes to a file simultaneously, speeding up debugging the code, and the interactions between classes.

GitHub also aided communication, allowing the team to leave comments in each other's code, as well as providing a Wiki for the project. At the beginning, the Wiki was filled with the basic outline for each section of the project, with a specification for each class. As the database was added, the wiki was updated to detail the format of the Appendix Files, and information was added to help the team understand the other members’ code.

The Code itself was developed in C++, using Visual Studio 2015. C++ was picked as several members of the team were familiar with the language, as well as being commonly used for projects involving databases. C++ is also an efficient language, making it a good choice for projects dealing with large data sets (Baldwin, 2011). Originally, the UI was being developed in C#, due to the UI team being more familiar with the language. Once they were more this changed to C++ once the team members felt more confident in C++ to avoid needing to link multiple languages. Visual Studio natively supports C++, as well as being available to the whole team in the labs and at home through DreamSpark. VS is also the IDE most familiar with the team, making it ideal for everyone to use for the project.

One issue with using Visual Studio was the amount of generated binary files present in the folders being version controlled by Git. This resulted in some issues at the start with pushing commits to the remote repository on GitHub. As the group became more comfortable with git, we created a git-ignore, removing the frustration of binaries.

Google Drive was also used by the team in order to collaborate writing the report, enabling everyone to work on their sections in the same manner that GitHub allowed simultaneous coding. Drive let everyone access the most up to date version of the report, and also let everyone work on their sections individually, while removing the need to manually compile sections, and enabled everyone to add relevant information.

* Google drive (word source control)
* Visual Studio (IDE)
* Git (source control)
* GitHub (online repository/ remote)
* Microsoft Office Word (word processor)
* 365 Email (main and official emails)
* WhatsApp (quick communication)

# Critical reflection:

## Honest Appraisal

### Teamwork

#### What we did right

* Team wide, consistent input.
* Effective communication arrangements.
* Good utilisation of team members’ strengths.
* A set schedule/ routine, which members could rely on.
* Non-restrictive system, that allowed everyone to volunteer for sections, that they preferred.

#### What we could have done better

## Belbin role utilisation

Belbin Results:

Damon - Coordinator, Shaper

Jake - Plant, Team Worker

George - Plant, Specialist

Alex - Monitor Evaluator, Plant

Jichan- Implementer, Team Worker

Tao- Monitor Evaluator, Implementer

# Group Members

## George Onoufriou

Roles:

* Database management Design (Appendix G, File.h & File.cpp)
* Report (Design, Architecture, Criteria & report formatting)
* Programming (File.h & File.cpp)
* Data Gathering (Appendix A-H)
* Co-ordinator

Description:

George, functioned as the plant and co-ordinator for sessions and general organiser of the group. This included making sure that everyone contributed. He was also the architect of the overall system.

Grade: 100% (A), Attendance: 100%. StudentID: ONO14475496 Belbin: Plant, Specialist

## Jichan Liu (Ken)

Roles:

* Programmed UI (C#)
* Finding relevant resources
* Research (Linking C# to C++)
* Code UI in C++(Gaming interface)

Description:

Ken, functioned as one of the two implementers of the team. Critically assessing situations, such as the hard decision to swap to a C++ interface, for easier implementation. He primarily worked on the backwards and forwards, game interface.

Grade: 100% (A), Attendance: 100%. StudentID: LIU13490935 Belbin: Imp, T.Worker

## Tao Sun

Roles:

* Design of UI interface
* Programmed UI (C#)
* Research (Linking C# to C++)
* Code UI in C++(Home page)

Description:

Tao, functioned as one of the two implementers of the team. Communicating and pair programming with Ken, to effectively create two interfaces. He primarily worked on the home pages of the UI’s and error prevention.

Grade: 100% (A), Attendance: 100%. StudentID: SUN13473462 Belbin: M.Eval, Imp

## Jake Trigg

Roles:

* Abstracting concepts (Concept.h & Concept.cpp)
* Programming (Concept.h & concept.cpp)
* Report (Tools and Methodologies)
* Updating Wiki (Concept)
* Debugging (concept)

Description:

Jake, functioned as the monitor evaluator, making good judgements for the team, and especially for the program, like changes to make responses more human. Helping all the members of the team make good decisions while completing his own roles.

Grade: 100% (A), Attendance: 90%, StudentID: TRI14468975 Belbin: Plant, T.Worker

## Alex Parker

Roles:

* Abstracting personas (Persona.h & Persona.cpp)
* Programming (Persona.h & Persona.cpp)
* Report (proof-reading & )
* Debugging (Persona)

Description:

Alex, functioned as the shaper, working well under pressure. Overcoming all the problems thrown at him. He went out of his way to learn the new skills necessary to tackle his initial difficulties with C++.

Grade: 100% (A), Attendance: 60%, StudentID: SMI13458446 Belbin: M.Eval, Plant

## Damon Smith

Roles:

* Report (Drafting; Honest appraisal, Passing the Turing test & Critical Reflection)
* Updating Design of UI interface
* Updating Wiki for UI

Description:

Damon, functioned as the team worker, helping the team to gel. He was especially critical to writing, draft sections of the report. He was also key to getting the foreign students to mix with the team, initially.

Grade: 100% (A), Attendance: 60%, StudentID: SMI13458446. Belbin: Co-ord, Shaper

# Records



Figure 4 (above): Attendance record, post instructor meeting.

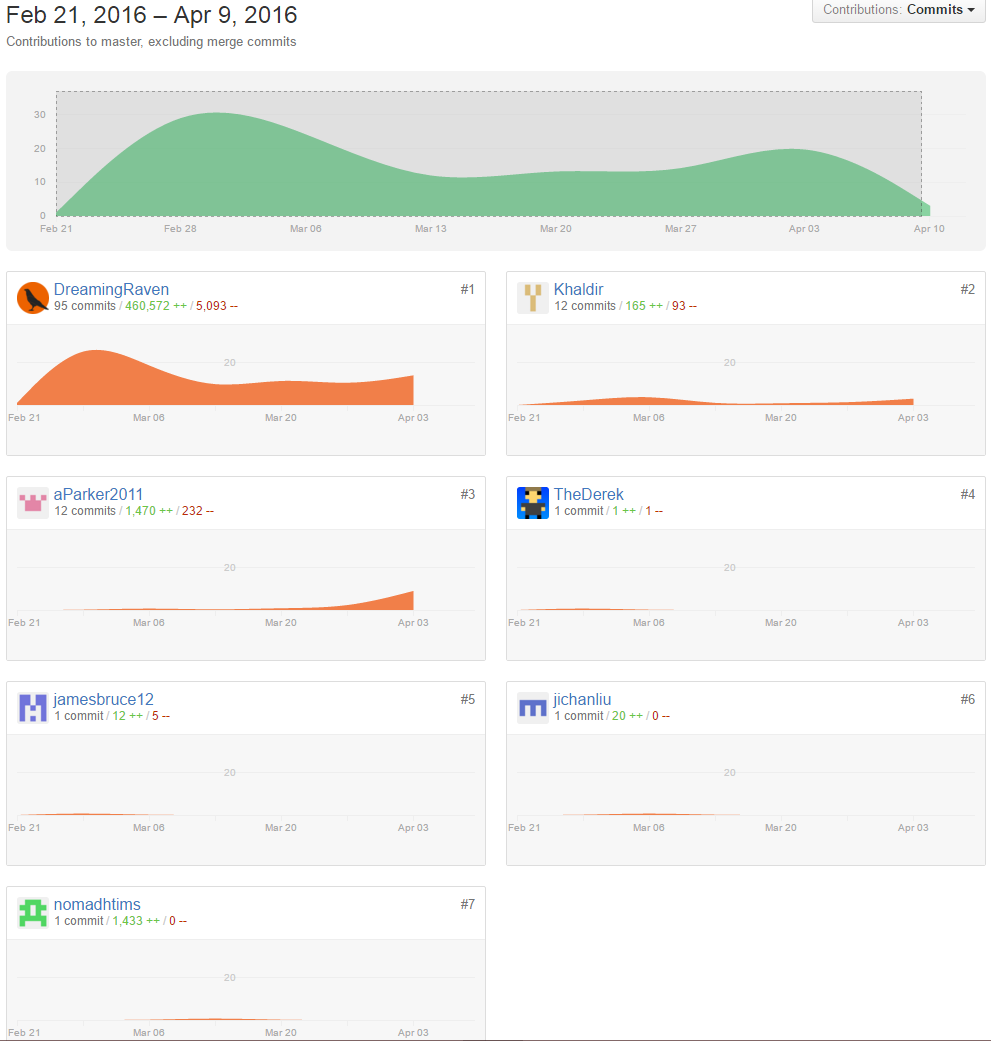


Figure 5 (above): Commit frequency record. (GitHub 2016)

# Bibliography

* GitHub (2016) *Word-Association.* Available from <https://github.com/DreamingRaven/Word-Association> [Accessed 13 April 2016].
* Hanheide, M. (2016) *Knowledge Representation* [lecture]. Artificial Intelligence CMP2020M, University of Lincoln, 29 January. Available from <https://blackboard.lincoln.ac.uk/bbcswebdav/pid-1045624-dt-content-rid-2313639_2/courses/CMP2020M-1516/CMP2020M%2002%20Knowledge%20Representation%281%29.pdf> [accessed 13 February 2016].
* Nelson, D. L., McEvoy, C. L., and Schreiber, T. A. (1998) The University of South Florida word association, rhyme, and word fragment norms. Available from <http://w3.usf.edu/FreeAssociation/> [Accessed 1 February 2016].
* Baldwin, B. (2011) *Why is C++ so fast?* Available from <https://lingpipe-blog.com/2011/07/01/why-is-c-so-fast/> [Accessed 13 April 2016].
* University of Reading (2014) *Turing Test Success Marks Milestone in Computing History*. Available from <http://www.reading.ac.uk/news-and-events/releases/PR583836.aspx> [Accessed 14 February 2016].

\*\*Cover page

\*\* abstract; short paragraph of what we did

\*\* introduction; summarise proposal

\*\* move criteria up

\*\* reflections - components - aims - design implementation

# Appendix

## One

Within the group project you were assigned groups based on a Belbin SPI test to judge how you would work in a group, and what processes you would work best at within the group environment. As we had a majority of the group that didn’t fully complete the Belbin test we had no idea what the strength and weaknesses of each member would be. So for the initial meeting we had to work out what the strengths of each member as well as the weaknesses were. To do this we decided to throw out ideas for the concept of the project. Doing this we realised certain members were more creative and others were more suited to expanding from the ideas. We had 2 initial ideas that we came up with, and working as a group debating the ideas which were in turn very different we listed the strengths and weaknesses of the ideas, as then collectively decided on the AI word association game. The reasoning behind this was the group were all motivated and although other members were passionate about their own chosen ideas, we put all our effort to the idea straight away without any conflict or issues and everyone become passionate about how we would develop our idea into a working artefact.

After we decided on the idea we had to expand it into a working progress in something we all had confidence in it passing the Turing test. Doing this we specified what we wanted our game to be and how someone would interact with the game. So we planned as a group on the objectives we wanted to achieve. But ran into the problem of how we would actually build it. So as a group we asked what the majority of people were comfortable with and this turned out to be C++ which would be the language we would develop the system within. After this we had to find a way we could all work simultaneously on the sections of work we had been assigned. So we decided to use GitHub. As a few members had used this previously, but the majority had never. This caused some friction as work was being handed in slower as some members had trouble uploading or pushing through improvements they had made. So other members were having to wait for other members to finish their work before they could then start on their next section. This was not a major problem so we worked as a group to get all the work that was completed in the same area. Doing this we managed to get all the systems working together which were the Persona, Word Database, and the UI interface. After we had the game working, we sat down in our next meeting and worked through any errors we found and some of the group worked on debugging the program, and others worked on the report but also helped if anything was needed or a fresh pair of eyes were needed to look at it in a different way.

Evaluating the group project we found that working in a group environment with a full range of personalities and backgrounds can be extremely difficult, especially when people have different commitments and time constraints can be a real problem. At first our group worked well together with the first assignment we got it all completed without too many issues. After this we set out a realistic plan of what we wanted to achieve and assigned out different sections of work to groups of people. As we had six people within our group we split up into two separate groups and divided up the work equally, this way we could achieve our goal and when both sections were completed we could gather back and work together to link them both. Doing this caused some problems as programming skills varied throughout our group causing some delays and problems especially during the two week break we had. Group tensions flared as some members had completed their sections of work and was waiting for others to add their sections. This did cause some friction but working together we managed to pull through and towards the end of the artefact we worked together to finish. Having the issues at first seemed bad, but helped us in the long run with working towards deadlines and also cleared the air within the group which ended up leaving everyone happy as all the work got done and to a better standard. Within our group we did have some friction and with other people having different commitments and also technical difficulties caused some delays but ultimately everything worked itself out and left us completing our artefact to a high standard and also left everyone motivated and pleased with what we all created, as working together we managed to stick to the original objectives we planned out. Which left us with a functional game that we hoped could pass the Turing test.

## Two

The task that was presented was to produce an artifact that is capable of passing a simple form of the turing test. Which is a way of determining whether or not you're interacting with a machine or a computer by the question ‘Can Machines Think?’. This means one of the major factors of the turing test is attempting to fool a person into believing that they are in interacting with a human - or in our case competing with a human player. This caused many tasks to become more technical as we had to provide the AI system with human traits such as making errors, acting outside the norm, and also being able to adapt to different circumstances.

Our group was confident that we would be up to the task in making our AI system seem as human as possible, which in turn would provide the necessary traits to pass the simple form of the turing test. We decided our AI would be one that would play the “Word Association Game”; wherein two players take it in turns to respond with a word related to the word the other play just said, where a player will lose if they repeat a word. To help decide how to go about designing an AI capable of playing this game in a human-esque manner, we first played this game among ourselves to determine what we did differently from each other, as well as what we had in common. We quickly realised that timing, and thinking in different ways such to create different relations between words were the two main topics that we should focus on. So we would be need to be able to create an AI system that could take varied lengths of time, but also when ran multiple times would throwback varied answers.

This was the part that was tricky, as we had found ourselves a database of words and certain associations with different words such as ‘Animal’ to ‘Dog’ and ‘Fruit’ to ‘Apple’, these were the simple associations that we all could think up. But then we thought we could upgrade our system to have multiple ways of thinking, such as through the guise of a builder, who would think differently when prompted with the word ‘sand’ compared to a 20 year old student’s association. We then decided that this would be the key factor to help us pass the turing test; getting the AI to adopt different personas vs different people. We decided to develop different personas and try to make a wide set of associations that we would think that person would make. One of the personas we created was Bob; a 30 year old Builder. Once we had the groundwork of his age and occupation, which were the biggest enumerable differences that we could see that would change a way a person would think. We then adapted our word database with associations specifically linked to Bob.

Doing some research around different AI systems that had passed the turing test, we had noticed our idea turned out to be similar in context to passing the test with another AI system that was developed. This was a chatbot developed by Vladimir Veleslov, which was an AI system that was based around a persona of a 13 year old Ukrainian boy. The criteria for him to pass the turing test was to fool 30% of the 30 judges that were present, in which he achieved 33%, which meant he passed the turing test. Reading this gives us a massive amount of confidence that if we were to test our final artifact we would have a valid chance in actually passing the turing test due to similarities in our methods.

----

Common social conduct often leads people to ask certain questions when they meet new people, and have certain factors that they use to form opinions and impressions of others from. From this we decided that the personas we used would have 3 specific factors displayed; name, age, and occupation, so by giving our AI system these attributes similar to how Vladimir did. It would be a great start for us to start building our system that is capable of passing the AI system by tricking the user.

Through the production of the artifact there was always constant testing to ensure that our system stuck to our rules that we set out in order for it to appear human. One of the main issues we came up was selecting a suitable word for the AI to throw back to the user, for example if the user used ‘Umbrella’ the AI would throw back ‘Rain’, however despite this being a valid response, it would flag suspiciously if given every time, so we made sure the AI had multiple words that associate with ‘Umbrella’ such as ‘protection’ and ‘weather’ that it would choose from randomly, this way if the game was played more than once there would always be valid responses that would vary but also had the chance of being repeated.

Doing this also created another problem as with the game we had developed a set of rules that both the user and AI would have to follow. This was that once a word had been used it couldn’t be repeated. So we had to ensure that once they had both used a word that they were stored somewhere that if it was used again the other player would win.

When the Artifact was completed we tried to stick to the objectives that we set, which were create a different persona that varied its responses from a set criteria that we laid out, which was ‘Bob the 30 year old builder’. After this we adapted the persona to think if he was actually playing the game what would he do, such as errors and varied response time, which in the end product was all implemented. Our final product stuck to all the objectives we set out, and gives us great hope of actually passing the turing test.

For the future we will developing our AI system further providing it with a more smooth feel, as currently there are still a few bugs and conditions we have to refine to be fully confident of our systems capability in passing the turing test. We also asked ourselves the conditions that would define the winner of the game. These were a user taking too long, a loose association with a word which would be based on the user's responses, as well as repeating a word that had previously been used. Currently we have the adopted persona in which they will try to stick to a specific set of words that we have defined.

Once we have implemented the changes the testing phase begins where we will actually be putting the system against unfamiliar users and trying to see how they respond to our game and whether or not we can trick them into believing they are playing against another human. To do this we will have to test a group of users individually by them actually playing the game and then testing them afterwards to see if they believed who they were playing against or if they believed it was a different persona/person. Or if they were dead set against it being a human and they were sure they was playing against a computer