

Concept Connect

ECE1778: Final Report

Apper:

Hyunmin Cheong

Programmers:

GuanLong Li

Sina Rasouli

Due Date:

April 12th 2013

Word count:

Main Report (not including Figures/captions): 1984

Apper Context: 498

List of Contents

Page

1. Introduction	1
2. Overall Design	1
3. Statements of Functionalities	2
4. Lessons Learned	5
5. Contribution by Group Members	6
6. Apper Context	6
7. Future Work	7
8. Disclaimer	8
9. List of Resources Used	8

1. Introduction

The project “ConceptConnect” focused on developing a social gaming app that helps users practice associative thinking. The app proposed the current project based on the premise that associative thinking is central to producing creative ideas and can be trained.

Our group developed an association game to be played between two players. The first player finds similarities between two concepts. For example, between the concepts “subway” and “newspaper”, which could be provided by the app, the first player could find similarities such as “transport medium”, “periodic”, “used by people”. The list of similarities and one of the two concepts are then sent to the second player, whose goal is to guess what the other concept is. The proposed app enables users to play this association game (Figure 1).

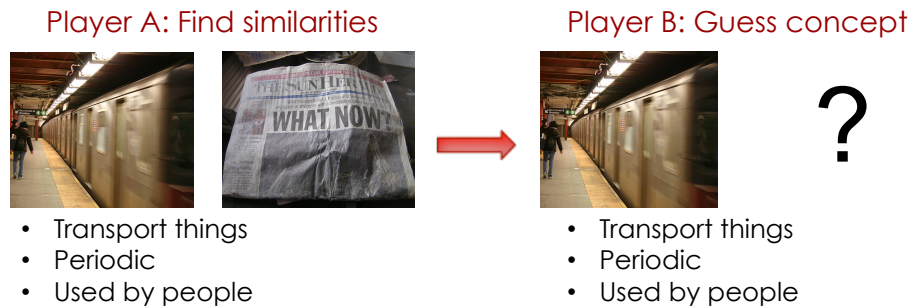


Figure 1: How the proposed game is played between two players.

Besides enabling users practice associative thinking, the gameplay data generated by users could be analyzed to identify patterns in types of associative thinking people use. The analysis would contribute towards understanding the creative thinking process.

2. Overall Design

Figure 2 shows a block diagram depicting the overall app design.

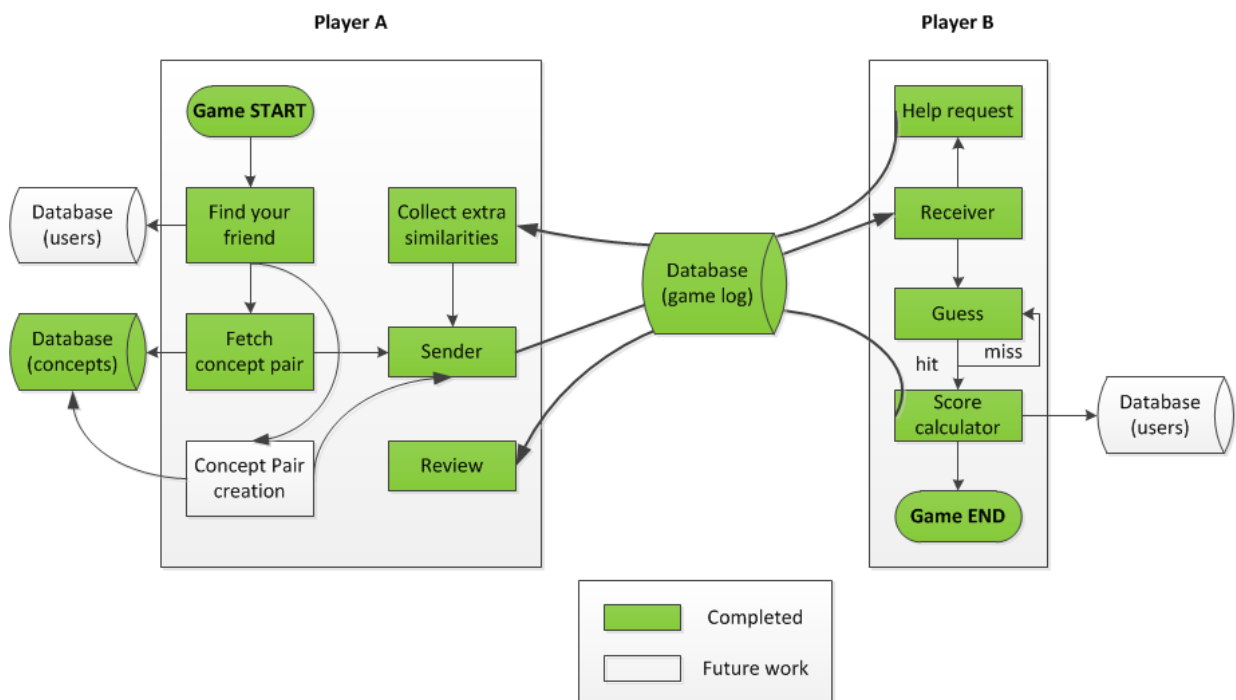


Figure 2: Block diagram illustrating the overall app design.

App functions

- » **Find your friend:** Locates a friend to play the game with from the user database online.
- » **Fetch concept pair:** Randomly fetches one of the concept pairs stored in the concept database online.
- » **Sender:** Sends the challenge created by Player A (concept pair, similarities identified, usernames of players involved) to the game log database online.
- » **Receiver:** Fetches the challenge intended to be solved by Player B from the game log database online.
- » **Guess:** Prompts the guess for the unknown concept and compares it to the answer.
- » **Help request:** Upon receiving help request from Player B, update the status of the current challenge in the game log database as “help”.
- » **Collect extra similarities:** Upon identifying the status of the current challenge changed to “help”, prompts Player A to identify additional similarities.
- » **Score calculator:** Calculates the game points based on the number of guesses Player B made.
- » **Review:** Display current and past challenges from the game log database online.
- » **Concept pair creation:** Allows users to create their own concept pairs. Imports images captured with the device camera and allows users to name the concepts depicted by those images. Stores this information on the concept pair database online.

Databases

- » **(users):** Contains a list of usernames registered
- » **(concepts):** Contains a list of concept pairs identified with concept pair IDs, and corresponding images
- » **(game log):** Contains a history of game logs with the following attributes:
 - Game (challenge) ID
 - Username of the player who sent the challenge
 - Username of the player who receives the challenge
 - Concept pair ID
 - Similarities identified
 - Guesses made
 - Game status: new/help/completed

3. Statements of Functionalities

3.1. Login and main menu

When the app is first launched, it checks whether or not the user has registered for the game. If the user has already registered, it will automatically launch the main menu (Figure 4); if not, it directs the user to the registration page shown in Figure 3:

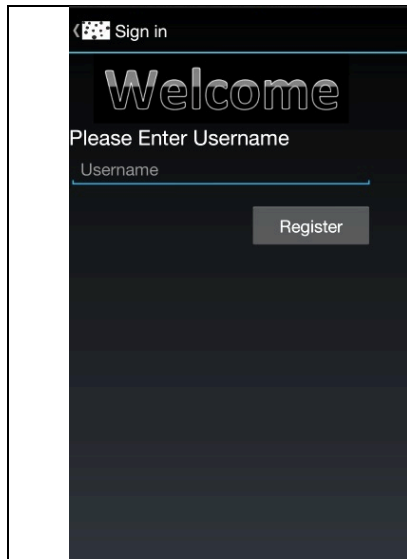


Figure 3: Registration menu

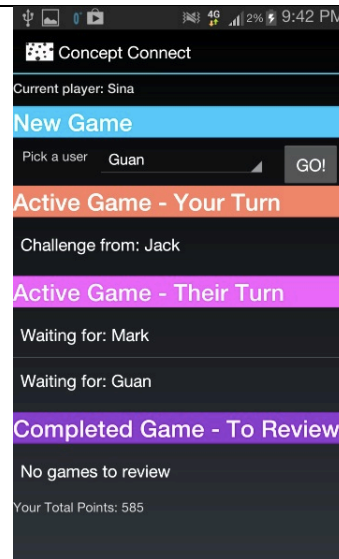


Figure 4: Main menu

3.2. Find your friend

In the main menu, the user can select the drop down menu under 'New Game' where all registered users can be accessed (Figure 5). After selecting a user to play with, the user can initiate a game by pressing the GO button.

3.3. Fetch concept pair

Once the user initiates a game, the app fetches a pair of concepts randomly from the online database and displays them with three text boxes for the user to type in similarities (Figure 6). Each concept pair has a unique ID associated with it, which is used to keep track of which pair the users are playing on. Once the user identifies three similarities, the challenge can be sent and this activity is recorded in the online database. Note that the program prevents users from typing in the same words used to describe the concepts as similarities.

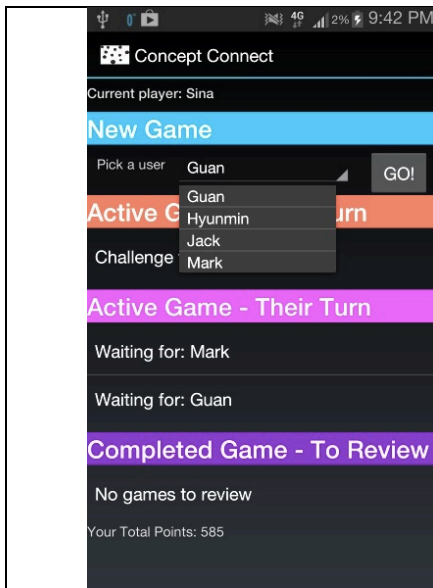


Figure 5: Finding a friend and initiating a game

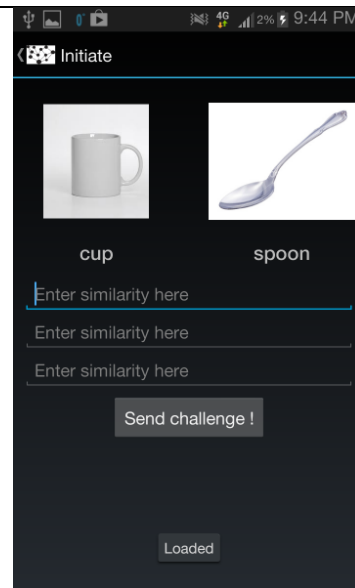


Figure 6: Concept pairs loaded for the gameplay

3.4. Sender/Receiver/Review

The main menu (Figure 4) shows three other sections beside ‘New Game’. ‘Active Game - Your Turn’ shows a list of games created by another user, waiting to be solved by the current user. Clicking on a listed game will open up the challenge (Figure 7). ‘Active Game - Their Turn’ shows a list of games created by the current user that are waiting to be solved by other user. Lastly, clicking on one of the completed games listed in ‘Completed Games – To Review’ allows the current user to see all the guesses made by the other user (Figure 8).

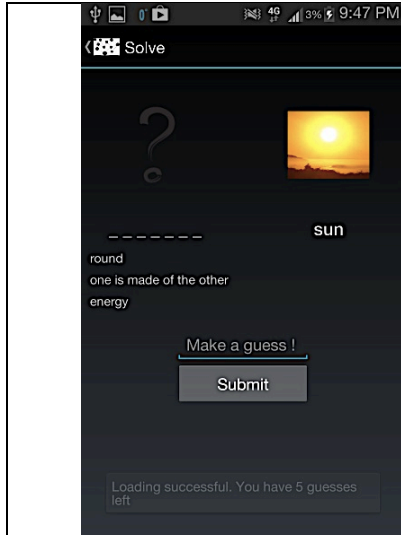


Figure 7: Challenge loaded to be solved

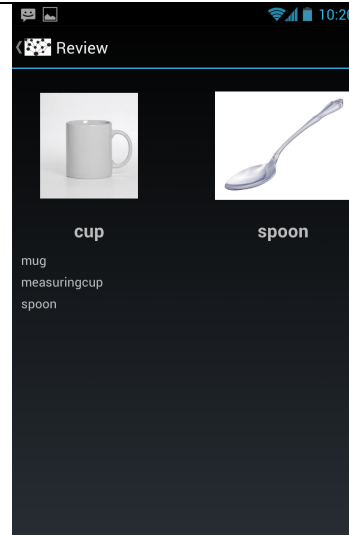


Figure 8: Reviewing the guesses made by another user

3.5. Guess/Help request

Guessing the unknown concept requires the user to input guesses in the “Make a guess!” form. The program checks whether this input is same as the correct name of the unknown concept.

Some features were added in to reduce the difficulty of the game. First, the number of characters of the word describing the hidden concept is shown. Another was the ‘Help Request’ feature. If the user is unable to guess the hidden concept from the first three similarities, the user can request for more similarities from the other user who created the challenge. Figure 9 shows how help can be initiated. The game is then sent back to the initiator and an additional similarity is requested from that user (Figure 10).

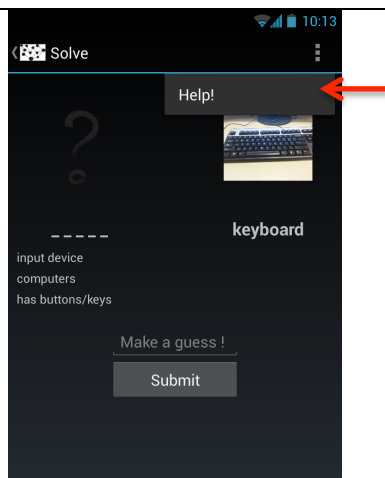


Figure 9: Initiating a help request

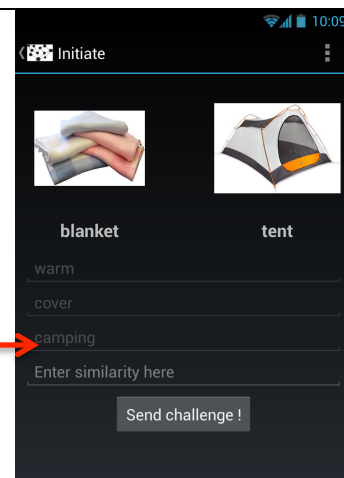


Figure 10: Prompting to identify an additional similarity

3.6. Score calculator

The game points are based on how many times the user requested for help, and how many guesses the user took to get it right. If the user gets the correct answer in the first try with no help, the both challenger and guesser will be awarded full points; whereas if they guess wrong after five tries, neither of them will receive any points.

3.7. Concept pair creation:

The users could also decide to create their own concept pairs. Figure 11 shows how this can be initiated. The users could type in a concept name and take a picture with their phone camera (Figure 12). Clicking on the dotted symbol will load up the camera.

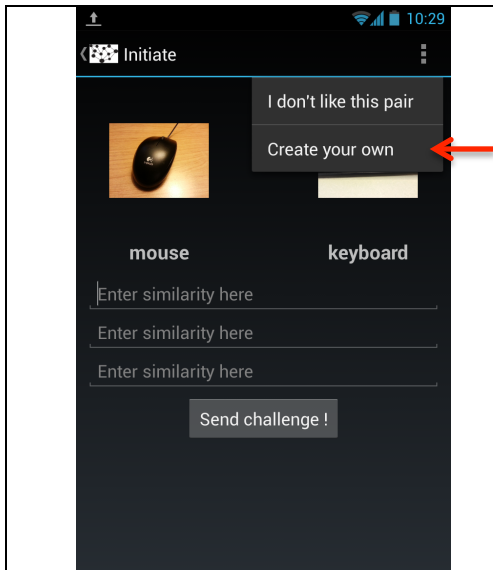


Figure 11: Initiating to create concept pairs

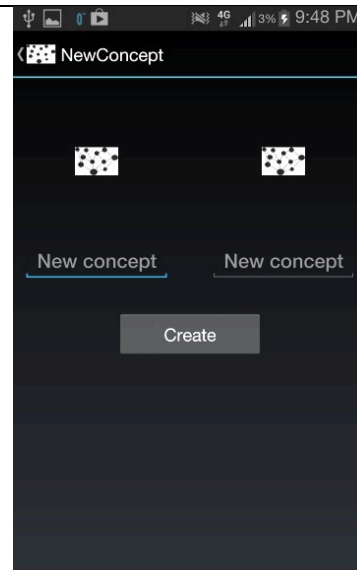


Figure 12: Creating concept pairs

4. Lessons Learned

4.1. Server data communication and management

All game data are stored in tab-delimited text files. These text files are modified accordingly each time the app calls the PHP script. One issue with this method is that in order for the app to be able to write data onto the site from any device, read/write permissions had to be given to these files. Hence they are vulnerable and open for anyone to add, modify or even delete. We should have set up a more secure database with a type of authentication such that the files stored online would not be exposed to such vulnerabilities.

The current use of tab-delimited format necessitates parsing the files to extract the information the app requires. This increased the complexity of the app and the PHP script. In order to reduce complexity and enhance visibility of the code, a markup language such as xml could be used or an SQL database to properly create the different fields and tags of all types, such as string, integer, etc., to manage the data.

4.2. Text data manipulation

The gameplay mainly involves handling text data generated by users, e.g., similarities identified and guesses made for the unknown concept. The main challenge involved handling synonyms that users could use to describe the same concept. For example, in guessing the unknown concept “subway”, a user might input “metro”. A simple string match would identify this input as incorrect, although the user means the same concept.

Initially, our group looked into using external synonym databases to check different possible correct answers for each concept. However, we decided that any synonym database is unlikely to be robust enough to handle all possible cases. Instead, we implemented a feature that tells the user exactly how many letters are in the correct answer. This feature at least limits the number of answers that the user could use.

Strategies used in other word-based games could have been adopted. For example, a user could receive a mix of random letters, some of which contains the letters of the correct guess. Another approach is to reveal the letters of the correct answer one at a time as a hint. These strategies further add gamification elements to the game and could make the gameplay more interesting.

4.3. Creating concept pairs

The Apper explored different ways of automatically identifying a sufficient amount of appropriate concept pairs. In particular, the Apper tried using WordNet, which is a lexical database that defines relations between sets of words, combined with the WordNet::Similarity module developed by Ted Pedersen. The module can calculate semantic similarity scores between a pair of words. Therefore, the Apper could examine the degree of semantic similarity between two selected words and determine their appropriateness for the game.

However, for this game, relying on semantic similarity measures was found to be inappropriate. For example, “newspaper” and “magazine” showed a relatively high semantic similarity score of 0.57 (scale: 0-1). More interesting pairs suitable for the game such as “newspaper” and “subway” only showed the similarity of 0.05. In fact, ideal concept pairs probably are associated non-semantically.

More robust solution could have been letting users identify concept pairs on their own. Although the App now lets users import pictures from their phones to create concepts on their own, more diverse methods of concept pair creation could have been planned from the beginning of App development.

5. Contribution by Group Members

5.1. Hyunmin Cheong (Apper)

Hyunmin proposed the idea behind the App, initially developed the gameplay, and designed the interface. He was responsible for finding and creating concept pairs with an appropriate degree of similarity. He also provided the access to a private server necessary for communicating game data.

5.2. GuanLong Li (Programmer)

GuanLong was mainly responsible for writing Android scripts to implement all the functionalities associated with the gameplay occurring **within** the client (app). He also architected the overall software design, including creating the block diagrams.

5.3. Sina Rasouli (Programmer)

Sina was mainly responsible for implementing functionalities associated with communicating gameplay data **between** the device and the web server. This includes writing PHP scripts on the server side and Android scripts to call the PHP scripts on the client side (app). He also worked on implementing the interface design.

6. Apper Context

Design research has mainly focused on developing tools to enhance creativity during design sessions, e.g., revised brainstorming techniques. However, developing a tool that is universally effective with different designers is challenging. For example, designers who are introverted or good at criticizing ideas may not benefit much from brainstorming. The apper believes that more effective approach is to train skills that are inherent in

creative thinking, such as associative thinking. The app developed enables potential users to practice associative thinking in a fun and interactive manner.

To evaluate the effectiveness of the app developed, further research should be conducted to examine the longitudinal effect of playing the game on the users' creativity. Measuring creativity is an ongoing research topic in the field of design theory and methodology. Often, researchers ask designers to generate ideas for a particular design problem, and design experts then rate how creative those ideas are. This evaluation process is intrinsically subjective.

The proposed app could also be used to evaluate creativity of its users. For instance, the types of associative thinking users use to identify similarities (superficial, functional, contextual, emotional, etc.) may indicate how creative a particular user is. Another simple measure would be the fluency of identifying similarities. More creative users would likely identify a greater variety and number of similarities. The user data collected from the app could be analyzed over a period of time. If a particular user produces a greater variety and number of similarities over time, this could be an indication that the user's creativity is being enhanced.

Given the current project's timeline, the app did not have the chance to ask multiple users to play the game and evaluate their gameplay data. However, the app intends to perform a pilot experiment. For instance, the app could use the Amazon Mechanical Turk (MTurk) to ask its workers to complete tasks that resemble the gameplay of the app. MTurk is a crowdsourcing online marketplace; requesters post simple tasks that can be performed online and workers can complete these tasks for small monetary payments. This study would at least reveal what types of associative thinking people tend to use to find similarities between concepts.

Although no formal evaluation could be conducted, the team's personal experiences of playing the game provided some interesting insights. For instance, without any guideline of what types of associative thinking could be used, e.g., superficial, functional, etc., the programmers initially had difficulty in coming up with similarities. However, upon playing few games and receiving suggestions from the Apper on different ways the similarities can be found, the programmers became more fluent at identifying similarities. This suggests that designers would greatly benefit from learning different types of associative thinking that they can use to inspire new ideas. For instance, Six Thinking Hats developed by Edward de Bono essentially encourages designers to plan six different thinking processes to generate ideas. The Apper could also develop a similar guideline specific for associative thinking based on further analysis of the gameplay data.

7. Future Work

The current version of the App allows users to import images from their phones and create their own concept pairs. In the future, other representation modes such as videos, songs, or other user created contents could be supported for creating concepts. These enable users more freedom to describe different concepts and also play around with abstract and sophisticated knowledge specific to a particular representation mode, e.g., similarities in themes in movies, similarities in beats in songs, etc.

To increase user engagement, more gamification features could be incorporated. In the future, the App should provide a ranking of registered users based on the points earned, to encourage friendly competition amongst players. In addition, the points could be used to purchase items that help solve challenges (e.g., revealing some of the letters of the correct answer). Finally, the App could provide increasingly challenging concept pairs for the challenge to users as they accumulate more points. This requires organizing concept pairs in different difficulty tiers.

We could also publish the game on social networks such as Facebook and enable users to log into the App with their Facebook accounts. This could obtain a greater amount of users to use the App and the users could easily

find friends to play with. In addition, a variety of social interactions can be incorporated, e.g., sharing challenges solved, posting a leaderboard of points earned, and gifting users with items that facilitate solving challenges.

8. Disclaimer

Our group does not wish to have this app used as a project for another course and the source code available online.

9. List of Resources Used

Generic tutorial site:

<http://www.vogella.com/android.html>

<http://www.mkymong.com/tutorials/android-tutorial/>

Java-PHP: image transfer

<http://stackoverflow.com/questions/9920967/android-post-base64-string-to-php>

<http://stackoverflow.com/questions/9987343/android-upload-image-to-server-using-base64>

Java: convert bitmap from/to string

<http://androidtrainingcenter.blogspot.ca/2012/03/how-to-convert-string-to-bitmap-and.html>

Threading: with *AsyncTask*

<http://developer.android.com/reference/android/os/AsyncTask.html>

<http://stackoverflow.com/questions/9671546/asynctask-android-example>

<http://samir-mangroliya.blogspot.ca/p/android-asynctask-example.html>

<http://www.vogella.com/articles/AndroidBackgroundProcessing/article.html>

Android: screen rotation lock

<http://stackoverflow.com/questions/4675750/lock-screen-orientation-android>

Java-PHP: communication with *HttpPost*

<http://stackoverflow.com/questions/6028981/using-httpclient-and-http-post-in-android-with-post-parameters>

<http://www.androidsnippets.com/executing-a-http-post-request-with-httpclient>

<http://stackoverflow.com/questions/2323617/android-http-post-how-to-get-the-result>

<http://stackoverflow.com/questions/4470936/how-to-do-a-http-post-in-android>

Android: read text file

<http://stackoverflow.com/questions/3344551/how-to-read-text-file-in-android>

<http://stackoverflow.com/questions/12421814/how-to-read-text-file-in-android>

Android: *ListArray* adapter

<http://www.ezzylearning.com/tutorial.aspx?tid=1763429>

<http://www.vogella.com/articles/AndroidListView/article.html>

Android: Progress bar:

<http://eagle.phys.utk.edu/guidry/android/progressBarExample.html>

<http://www.mkymong.com/android/android-progress-bar-example/>