ZhaoNhan

WORDS MEMORY

committing words to memory

Hong Zhao (3985714)
Quang Nhan (4297428)

CSCI342 - Mobile Application Assignment 4

ZhaoNhan

Table of Contents

ZhaoNhan

Mission Statement

Word Memory allow its users to memorize new words in an interactive way through gameplay offering perception of growth and discovery.

Scenarios

A student has been given a task to memorize key words that is critical to learn a new topic. How would he she go about doing this. Flash card? cover up the word and read out repeatedly? BORING!

A tourist wanting to visit Australia wanting to learn purely new words and commit it to memory. How does she do this? read books? That's fine, but the first step to use it efficiently is to remember it!

Description

As the name suggests, the game is designed to have one primary task - to help the user to remember words. Word Memory is based off of the model of repetition through exposure and users actively seeking for the letters that form the word. It uses short burst of time (50s) for each game to align with the idea of play pause play pause because the mind works better with intervals.

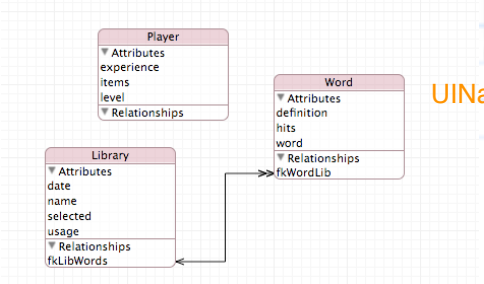
To keep the game interesting, we had introduced the notion of growth, development and discovery through the use of ingredients, power-ups and leveling. As the users actively play the game, there are chances that they may discover ingredients, scrolls and potentially a power-up that lay hidden under the letters. When these ingredients are found, they are added to the user's inventory. Once the game had finished, the user can combine the ingredients and scroll to uncover power-ups.

The power-ups spices the game play up with the ability to be used anytime while the clock is ticking down. Power-ups may freeze time, shuffle the gameboard to give a fresh perspective, or even highlight the words for you. It is envisaged that more power-ups may be added as the game is developed further.

The game also offer a leveling aspect. The idea of leveling was to be able to share with the user's friends his/her level, libraries, and achievement encouraging a little competition amongst his/her peers. Because due to facebook constraints that it require the app to be available on the app store and has an app id therefore we cannot add this features.

Additionally, there are three game difficulty level able to be chosen without restriction. Master 1 produces a 7 x 7 board, the master 2 produces a 9 x 9 board, while the master 3 produces 11x11 board. The idea is the bigger the board the harder it is to pick out the letters, but at the same time the higher the level the higher the probability to discover ingredients and chances of finding scrolls and rarer ingredients.

Storyboard

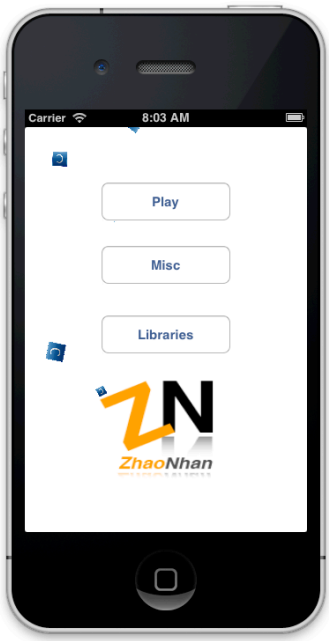


Core Data model

WM uses 3 navigation controller, 3 table view controllers, and 2 tab bar controller to navigate around the app

Walkthrough & Features

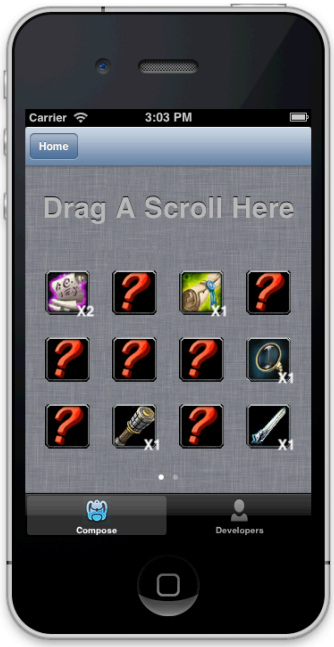
The user is presented with the startup screen with 3 options. To play, to create/view a library, or to explore and combine ingredients discovered whilst playing the game. Pressing the misc button will introduce to the compose view page. Combine ingredients by firstly selecting the scroll, then match the ingredients displayed by the greyscale icon if the user have the available ingredient. Once all of the ingredient had being dragged onto the list, the compose button will appear. Press the compose button will combine the ingredient, a particle animation and associated power-up will be uncovered. press ok to keep the power-up.



Start up screen offering the option to play, misc, and library buttons

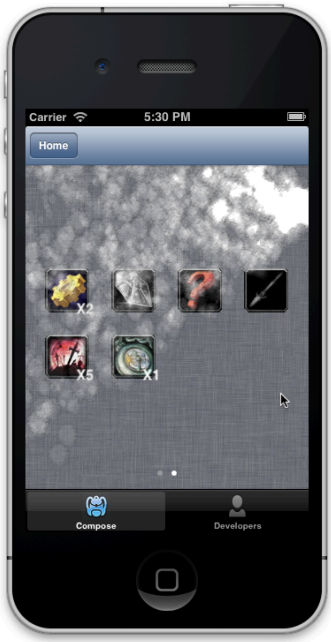
Developer screen

Misc

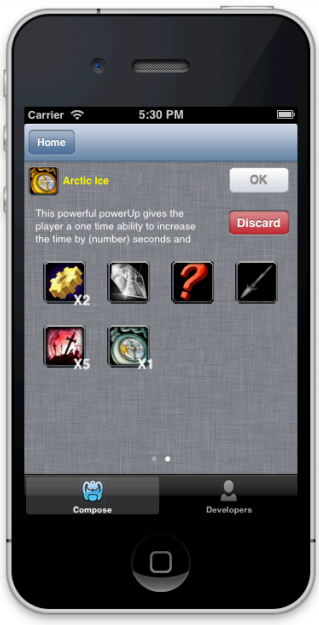


the combine ingredient screen displaying the ingredients and scrolls

after selecting the



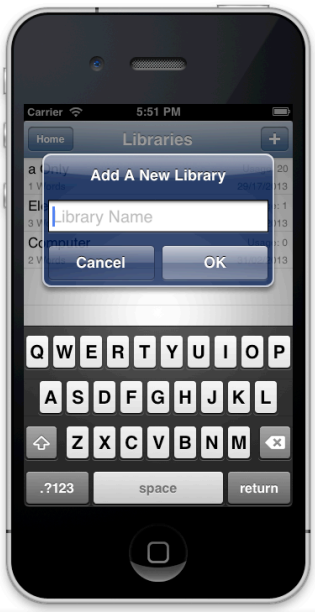
the animation effect when an ingredient is composing using particle effects using the quartz core



the animation effect when an ingredient is composing.

Library

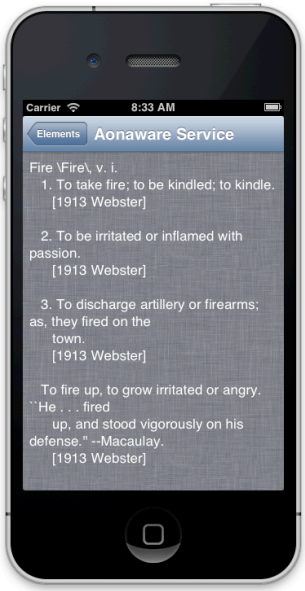
Before the game can be played, the user must build a library and create a list of words in order for the app to integrate it into the game board. Select the libraries button on the start up screen. Select the plus button on the navigational bar. Enter a new library in the alert box. press ok. select the newly created library. Again press the plus button on the navigational control screen and enter the words to be listed in the topic. If the user entered a duplicate words, an alert box will display a message to let the user know that he/she cannot do that. To view a online definition of the word, the user can tap on the word to pull data from Aonaware service. The user at anytime may choose to delete the word or library. Additional to the display of the word, the subtitle also displays the number of times the words had been found during the gameplay.



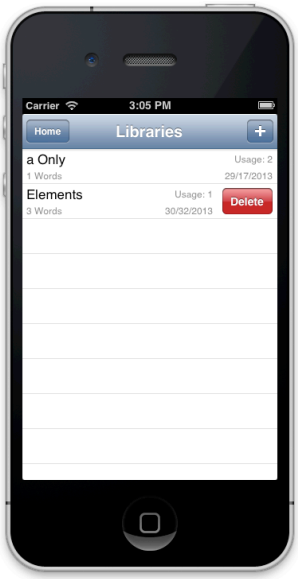
press the + on the navigation bar in the libraries brings up the alert box to add a new library

press the + on the navigation bar in the defined library brings up the alert box to add a new word

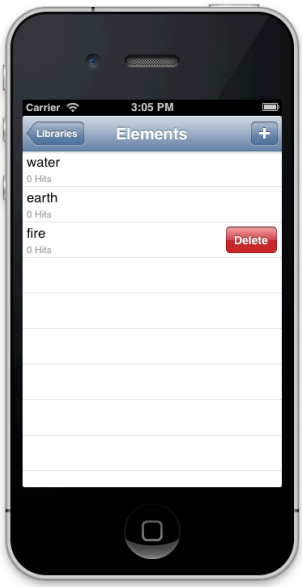
adding duplicate word brings up an alert box



clicking on a word pulls XML structure from Aonaware service



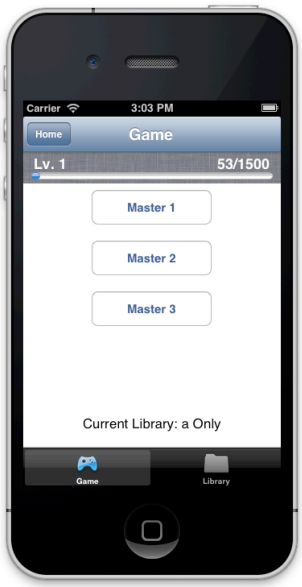
delete a library - swipe right gesture to bring up the delete option



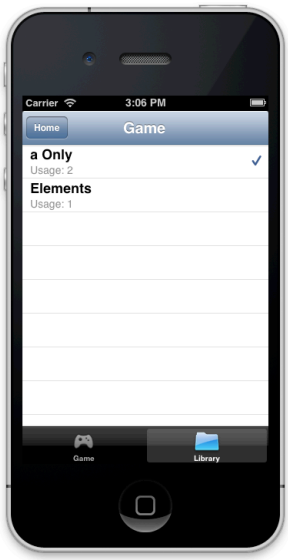
delete a word - swipe right gesture to bring up the delete option

Play

The game play offers three master mode. The master mode as outlined earlier offers different game board sizes. The higher the master level, the more distractive the board becomes but however the chances of finding of ingredients, scrolls, and power-ups increases. It also displays the current level and experience points the user had accumulated. But before the game can be played the user must make sure he/she had selected the library that the generated board-game must be based off of. Select the library tab, tap on the topic, and there should be a tick next to it. click on the game tab and then select a master level. Once the game starts, a countdown of 50 seconds are displayed. During this time the player form the word by tapping the letters in order. As the letters are tapped, it appears on the display view. If the user makes a mistake, he/she can erase the selection by tapping on the display screen this will unselect all the letters (this is the price of making a mistake). If the user wants to use one of his/her power-ups, he/she can do this by sliding the display view on top to the left to reveal the available power-ups. The user then can use it by selecting the power-up and drag it to the game board. Depending on the type of power-up he/she has the result will become visible. Once the time had ended the game will ask if the user want's to play another game. If the user chooses to not to play the screen will return to the master selection screen again and the user will see a short animation of the experience bar filling up.



developers tab bar to bring up the list of developers



Select master level page having 3 master level options as well as select library and game tabs.

Game play screen - master 1

Selecting the letters to form the word - master 3

Used magnifier power up to reveal letters to form the word

interface & HIGS

Word Memory uses iOs6 design guide.

Layout

WM uses consistent appearance throughout the app using Apple's view controllers to maintain external consistencies. Top down approach had been adopted, maintaining hierarchy of importance of information structure. The layout is simplistic and familiar so the user can learn the crust of the gameplay easily. Event-though master3 may prove to be less friendly in terms of the letter box size, it was intentional to keep it as a challenge. Other than that all buttons are easily selectable providing a user friendly experience that is familiar and reduces the learnability burden.

Navigation

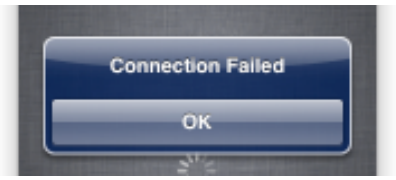
WM uses 3 navigational controller - for the play screen, misc screen, and the libraries screen - to navigate control the depth of the view stack. Additionally 2 tab bars were used on the play screen and the misc screen. This provide a branch view to different sub section of the app. These together with 3 table view controllers and some custom view controllers provide an easy to navigate app that can dip in and out between screens with ease providing a user friendly experience.

Modal

Although definitions can play an important role, we kept that as an optional feature allowing the user to view it when he/she wants to. Another feature of the app is the misc tab, we had not designed the app in such a way that it needs the power-ups to successfully keep up with the game-play, thus the user is not distracted from the more important task of playing the game.

Interactivity & Feedback

All actions in the app has a a form of feedback. When the user enters a new library or word, the cell is displayed with the words/library that had been just added into the system. The user can delete the the library or word by swiping right on the cell of the table, and the power-ups can be accessed by swiping left on the score view of the game play (refer to walkthrough section for screenshots). Each navigational bar has a title, and each tab bar has its unique icon and text to let the user be aware where he/she is. The level/experience visual bar and numerical display allow the user to know how much more is needed to reach the next level. The letters is highlighted to inform that they had been selected. The timer shows the remaining time left before the game ends. The library shows a tick to inform the user that he/she had selected this topic to play with, as well as a text feedback on the game play menu. If there's no connection to draw the word definition, a graceful alert box appear to inform the user of the problem. All of these feedback help aids the user's understanding of the app and provide a friendly experience.



Network failure alert box

a tick to let the user know this is the current library in play

Selected library is displayed on the play screen

Highlighted box to inform the user that these letters had been selected. The display shows the sequence of letters selected as well as the timer and if any ingredients found

visual feedback on the user's current level

Branding

The developer's logo is only displayed at the start up screen minimizing obtrusion on the app whilst maintaining its presence whenever the user wishes to enter the other modes. Additionally developer information can be reached in the misc screen if the user wishes to see more details therefore not affecting the core function of the app.

Color & Typology

Blue colored letters were used to be coherent with the default navigational bar - but at the same time does not intrude with the selected letters. Selected letters and highlighted letters has distinct color used to maintain contrast with the blue unselected cubes to minimize the effect of the colorblindness (tritanopia - those that cannot distinguish different shades of red). all prints are legible using only sans serif types to eliminate clusters of fanciful kicks and turns taking up unnecessary white spaces.

Icons

We support both the regular and retina display providing @2x.png icons. The design of the icon matches the title as well as the symbolic cube used during the gameplay. The letters board are dynamically created at runtime adjusting its size to fill in the views. This functionality allows the app to be displayed on both the 3.5 inch and 4inch screens.

Design

The app had used many default features like navigation bar, tab bar, tables, and alert box to maintain internal and external consistencies. Spell icons are large and clear enough to allow the user to interact with it directly. Apart from the gaming aspect the game also provide an educational value. We kept a balance between gaming and leaning through the use of animations, clear and concise layout that is simple to follow and gets the task done. The tab bar uses icons that metaphorically resembles real life items eg, the game pad to indicate it game screen, a folder to symbolize data - library, and person profile to indicate a people in the developer screen.

Delight

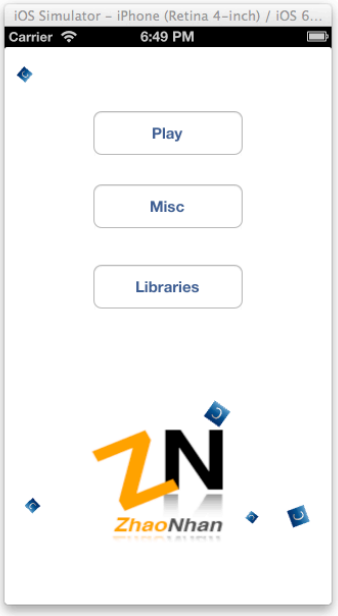
Throughout the app, there are minor elements of surprises, from the animation of falling cubes in the startup screen, the flip animation of the board when a new board uncovers, to the composition of the ingredients, and elastic rebounds from the incorrect ingredient selection to the fact of probability of finding the ingredients itself. This delightful devices help to make the app interesting and engaging.

Technologies & Platform

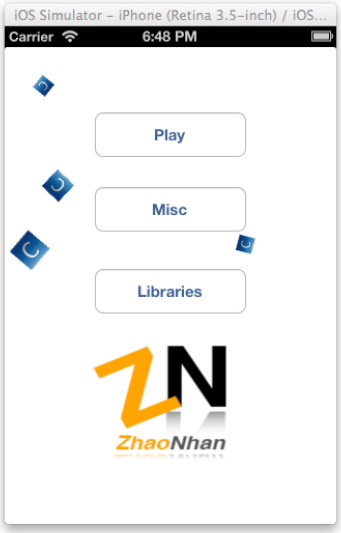
Although somewhat this is a game, but at the end of the day it is purposely designed to achieve a serious goal - to help the user memorize a list of words. Thus the design intent was functionally driven. Get in, interact with the game for 2 -15mins memorize the words and come back time to time to test and refresh yourself. Thus it is perfect to be designed on a mobile device where the user may interact with it at anytime and at anyplace - train trip, during lectures, or while waiting for dinner to be served.

Screen Support

The game support both 3.5 and 4inch iPhone screen in portrait mode.



4 inch load screen



3.5 inch load screen

3.5 inch screen game play

4 inch screen game play

Single Player Mode

Since the intent of the game is personal growth and development, the library is related to each individual. Apart from the intention of sharing library and publishing accomplishments, the game do not support multiplayer aspect due to the nature of each person has a set goal of learning his/her own words.

MVC pattern

The code is designed off of the MVC pattern where there are distinctions between the model classes, the view classes, and the controller classes. It takes full advantages of delegate methods and protocols to pass messages between these classes allowing the controller to inform the view what to display and how the model class using Core Data framework to store and retrieve information. For example the ComposeViewController uses the delgates ComposeCollectionViewDelegate, ItemCellDelegate, ComposeViewDelegate, UIAlertViewDelegate to handle and default and custom events for page scrolling, item dragging, item tapped, item cancelled, discard button tapped, ok was tapped, and alertview handling.

Model	View	Controller
Player Library Word NDTrie ItemDropModel Item Constants	ItemCell ComposeCell ComposeView ComposeCollectionView ItemDescriptionView WordCell PlayView TitleView UsableItemView ExpBar	ComposeViewController MiscTabBarController LibrariesTableViewController WordsTableViewController DefinitionViewController LibraryTableViewController LevelViewController PlayViewController PlayTabBarController ViewController AppDelegate

Graceful Failure Risk Aversion

There are 3 points to mention:
1. The definition of a word is pulled of of http://services.aonaware.com/DictService/DictService.asmx?. In an event if there is no internet connection, an alert box informs the user of this issue. Refer to Walkthrough and feature section - Library - section.

2. During the gameplay mode, if the user decides to put the game in background mode, the timer is paused allowing the user to continue when he/she re-enters the game

3. If the game is exits and deallocated from memory, all the information are saved using core data technology. Upon relaunching the app, the user will not notice any lost of data.

Web service integration & Other Challenges

Integrate with Anoaware Service

http://services.aonaware.com/DictService/DictService.asmx?op=DefineInDict

The app integrates with the web service by connecting to the Anoaware Service sending a request with the word

```
NSURL* url = [NSURL URLWithString:[NSString stringWithFormat:@"%s%@", WEBSERVICE, _word]];
NSURLRequest* req = [NSURLRequest requestWithURL:url];
[NSURLConnection connectionWithRequest:req delegate:self];
```

If the site returns an xml data and the app parse the data into an NSString and the user is shown the definition in the word detail screen.

```
- (void)parser:(NSXMLParser *)parser foundCharacters:(NSString *)string
{
    if (_currentElement && [_currentElement isEqualToString:@"WordDefinition"])
    {
        NSString* str = [string stringByReplacingOccurrencesOfString:@"{" withString:@""];
        str = [str stringByReplacingOccurrencesOfString:@"}" withString:@""];
        _definitionTextView.text = str;
    }
}
```

Other Challenges and how we overcame it: in particular Graphics

A considerable amount of time invested had been revolved around the graphic side of the app because of its nature of game. Upon the initial commit of this project, we were planning to use OpenGL ES to do graphics. However, we did not have much time to start it from scratches so we resorted to images and animations. We are not creative arts students so the only source of images is the Internet. The first task is to find images that are relevant to our project, that is, it can represent or symbolize entities in the app. The images of different letters with different colours are easier to find than those of items. The item images have to be consistent themselves, and they also need to be designed in such a manner that users know what they are straight away. Then, in order to fit them into our app, multiple modifications have to be done, such as transparent background, the size of the image and the like. The former is a little complex since it involves Photoshop while the latter can be solved by the photo utility shipped with Mac OS. Another change on the item images is to make a monochromatic version of them, representing their unavailability. This is also done in Photoshop.

Games are really graphics reliant with user interactions. The way to approach these interactions is animations. For example, in the Compose tab inside the Misc screen, the user is allowed to drag items from the collection view to the composition area. One problem is that if they drag an item that should not be dragged, such as an ingredient instead of a scroll on first drag, they should be informed that this item cannot be here now, and that the item will be put back to the collection view. The animation here thus is to move back the item to its original position in the collection view. Given the fact that the item image is built 3 levels down into the collection view, the delegate has to be passed through all these levels. The delegate is then able to tell the view controller which item is dragged. Besides, the coordinates are also necessary. However, the coordinate inside the item is a relative to its own view, so mathematical computation has to be performed to obtain the absolute coordinate in the view of the view controller.

Animations

List of animations used

Particle Effects
* The background of the initial screen
* The smoke of the composition

Fading
* Every time when the user finds a word, the whole word fades out from the title view.
* The “Drag A Scroll Here” message fades in and out when a user drags an item there.
* The buttons in the Misc composition view fade in and out.
* The item description view fades out in 5 seconds if the user taps the item without any further gestures.

Resizing
* The brick resizes itself when a user undoes his or her selections.
* The result of a composition resizes itself after it shows up.
* The item resizes in the Compose tab when the user starts to drag it.

Rotating
* The item dropped from a found word keeps rotating itself while moving to the top left corner of the title view.
* All bricks rotate themselves after the user finds a word.

Elastic Moving
* The view holding all usable items during a gameplay comes out with elasticity.

Other NSTimer-Based Animations
* The user’s experience points keep growing after a gameplay.
* The timer showing the remaining time during the gameplay
* The timer flashes and freezes if the user uses a time-stopping item.

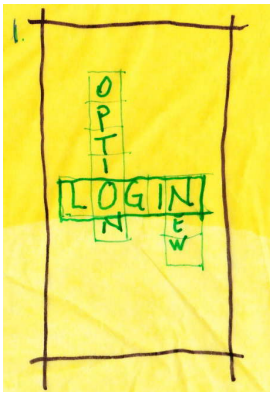
Third Party codes:

https://github.com/nathanday/ndtrie

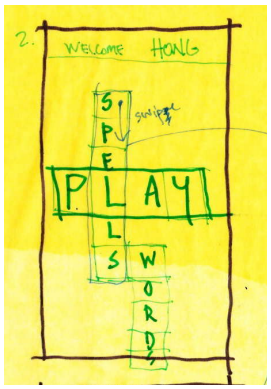
NDTrie written by Nathan Day was developed for text completion used to store and retrieve and match words with Key values in a trie, an ordered tree structure. It was a matter of copying the file and utilizing it.

Appendix

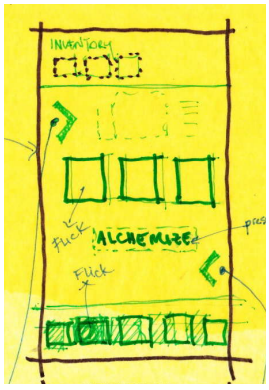
Initial Sketch Design



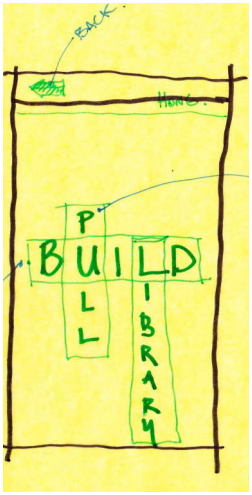
Startup screen showing the log in and option pane



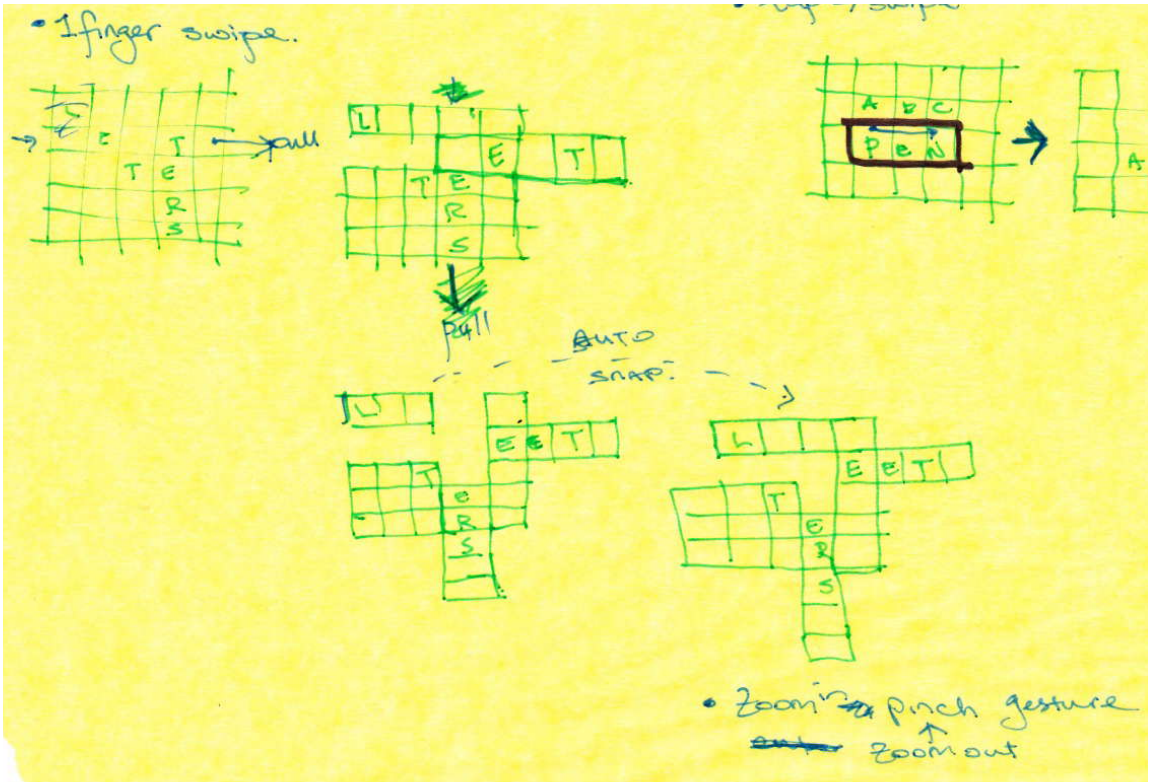
The pre-game screen showing the option to play, brew spells, and word library



The spell screen where you can combine ingredients to create power-ups



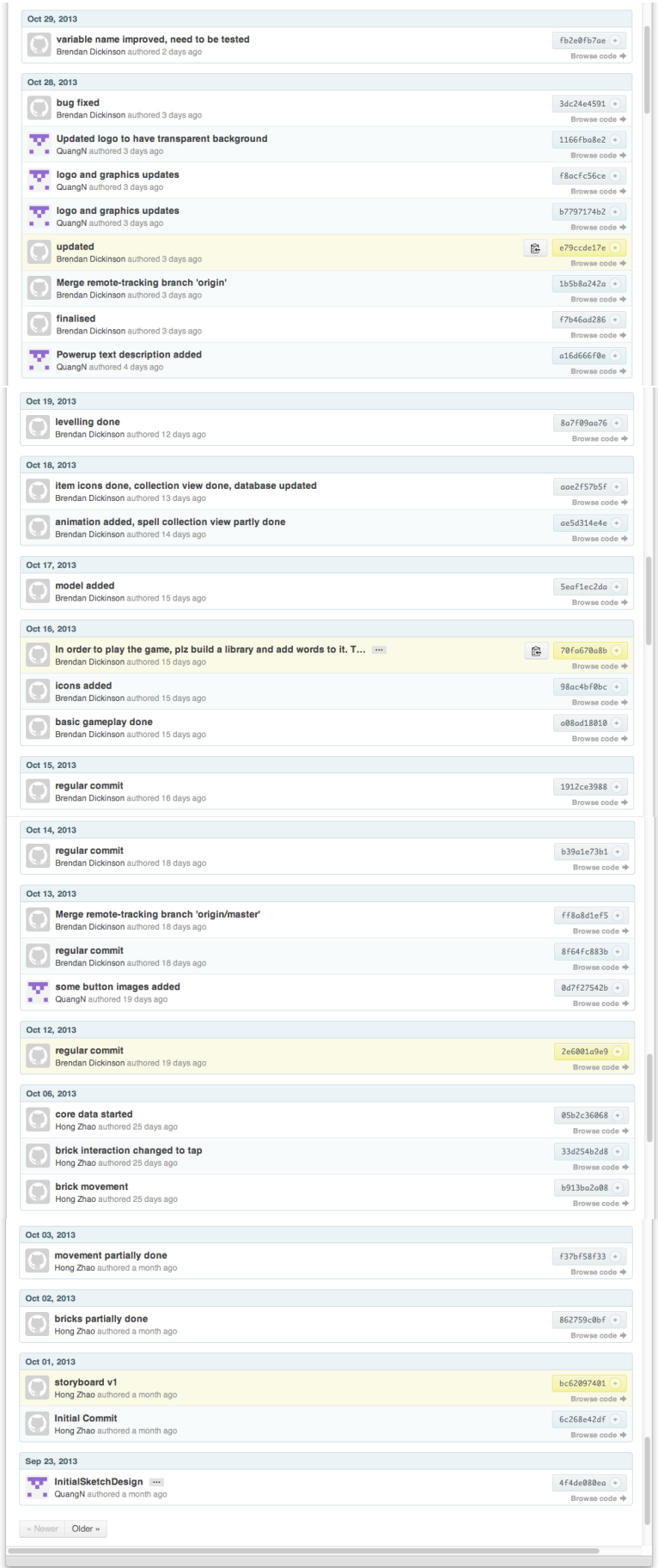
Word library screen, showing the options to build library, pull from internet and and view existing library



original gameplay board where it was based off of the pull and push model. Conceptually it would had been exciting but it would take too much development time and the complexity that will limit the time spent in the other area of the project.

GitHub

https://github.com/krauzezhao/CSCI342



Note:
Brendan Dickenson and krauzezhao are both secret identities of Hong Zhao

Timelog

These timelog are based on approximate hours combined spent for meetings, sketch design, research, coding, photoshopping, correspondence, and report.

Hong Zhao

40 hours/week = 165hrs over 7 weeks

Quang Nhan

13 hours/week = 90hrs over 7 weeks