

# CS 4720 - S18 - Final Project Proposal & Documentation

Device Name: Qwilfish

Platform: Android

Name: Tyler Handley

Computing ID: tmh6de

Name: Hunter Murphy

Computing ID: bhm5zr

App Name: Charades Reloaded

## Project Description:

- Heads Up! is a popular app on both Android and iOS where users play a game very similar to Headbanz or Charades, where a player is given a word on their phone screen that they cannot see. The player must then guess what word he/she was given using the other players' hints or gestures. However, in Heads Up!, all the possible sets of cards that you can play with are made by the developers. Additionally, the more advanced and interesting sets require payment to play. This lacks customizability and flexibility, is where Charades Reloaded comes in. This app will allow for much more customizability in nearly every aspect. The length of each game, the feedback for finishing card, and the card sets themselves are all customizable. How each game works is that the player moves the phone to their forehead with the screen facing out and try to guess as many of the words that come up before they run out of time. After the game is complete successfully guess cards are highlighted in green and incorrect/passed are shown as red.

What we propose to do is create an app that will do the following:

- The system shall allow a user to set game hyperparameters such as game length, allowed number of passes, etc..
- The system shall allow a user to create and edit sets of cards, storing them locally
- The system shall allow a user to tilt the phone forwards to say they have answered correctly and tilt back to say they are passing (i.e. giving up on the current card).
- The system shall record players through the phone's video camera and microphone during a game if the hyperparameter is set and then play it back at the end of a game.

We plan to incorporate the following features:

- Accelerometer - Used to detect tilts to see if a card was passed or answered. 15
- Data Storage using SQLite - Used to store the card sets themselves. 20
- Data Storage using SharedPreferences - Used for the hyperparameters that the user can set. 10

- Data Storage using file read/write - Users can choose to share a set using the share button on the home screen. This writes the set to a .txt file, which another user can then import into their version of the app.

## CS 4720 - S18 -Final Project Documentation

**Platform Justification:** If we were to release this app to the play store we would use the Android style of pricing where there is a free version with limited features and a premium feature with no restrictions. These apps are usually more common and successful on the Play Store, while the App Store uses more of the pay-to-download model. Also one of the developers does not have a mac so it is more convenient to code for android.

### Features

- **Major Features**

- Play: This is the screen where users can actually play game, using the sets they have personally created. This screen makes use of the Accelerometer, Data Storage using SharedPreferences and SQLite Optional Features.
- Sharing: This is the screen where users can create a text file of their card set and email or upload the sets to google drive. This screen makes use of the DataStorage using file read/write and DataStorage using SQLite.
- Managing Sets: This feature is broken into two screens, Manage Card Sets and Manage Cards. In Manage Card Sets the user can view all created sets as well as create new sets. Once they are clicked on the app brings the user to the Manage Cards screen where they can edit the title of the set or delete the entire set. Cards can be created, deleted, or edited here. This feature makes use of DataStorage using SQLite.
- Preferences: This feature is accessible from the Main screen and allows the user to set six different settings, 2 of which are dependent upon others. These settings are Game Time, Show Game Timer (toggle), Limit Cards Per Game (toggle), Cards Per Game (dependent upon Limit Cards Per Game), Correct Cards Limit (toggle), Cards to Win (dependent upon Correct Cards Limit). This feature makes use of Data Storage using Shared Preferences.

- **Optional Features**

- Accelerometer: We recognize the user's tilt movements in our "Play" screen. To test this, click on "Play!" from the main screen. Select a card set with cards in it and wait for the game to begin (essentially, start a game). Using one of the Nexus tablets provided to us by the university, hold the device perpendicular to the ground with the bottom of the screen towards the bottom of the ground. Tilt the top of the device down towards the ground until you hear a "ding." This means you got the card correct. Tilting the phone's bottom up away from the floor

until you hear a “negative ding” that will indicate that you got the card incorrect. (15 points)

- Data Storage Using File Read/Write: We allow users to share their sets by generating .txt files of their sets. To use this, click on the share icon on the main activity and click on “send a set.” Then select the set you’d like to share. Then select a share method that supports .txt files (email or Google Drive). Whoever you share the set with can then import the set by clicking on the share button from the main menu and then clicking on “import a set.” This will import the set if a set with the same name does not exist on that device yet. (15 points)
- Data Storage Using Shared Preferences: We allow users to save game preferences using shared preferences. To see these in action, first go to the gear icon on the main activity. Then select a preference, and if applicable, choose a value. Now go back to the main activity and click on “Play!” and proceed to play a game. Your preference will be reflected in gameplay (for example, if you choose to limit the play time, the amount of time remaining for the game will reflect that). (10 points)
- Data Storage Using SQLite Database: User card storage is done through SQLite. To test this, either click on “Create Your Own Card Set” from the main menu or click on “Your Card Sets” and then click on the floating action button in the bottom right. This will pop up a dialog to name your set. Entering text and accepting this will take you to an edit screen for the card set. Clicking the floating action button will create a new card for the set, which you can fill in with the text of your choice. Click the check mark in the top right to save the cards you’ve added to your set. You can also delete cards with the red x or the entire card set for the box for the Card Set Title. When you navigate back to this screen or the Play screen, your changes will be saved. (20 points)
- All of these optional features add up to 60 points.

## Testing Methodologies

We tested the app repeatedly while coding by making new sets, deleting and creating cards in the sets, sending and receiving sets, and playing the app. We would intentionally add and delete cards in odd ways (such as a user simply swapping the position in the list of two cards, a user deleting a card and then adding the same exact card, etc) to see if any non ideal behavior would arise. Performing these odd user behaviors led to a majority of our bug fixes, most of which fixed the way in which we added and deleted cards. The play testing was done with a group of friends to find key usability issues, which helped us to tune our tilt sensing during the game. Additionally, this allowed us to know whether our user preferences actually translated to gameplay, and that any user preferences entered in error (an empty string entered instead of a number) did not crash the application.

## **Usage**

One important note is that you must press the save button (the check mark) to save changes made to a set, if the user wishes to discard changes they merely need to go back to the previous page.

## **Lessons Learned**

Doing the core skills app was a great way to prepare because it gave us a existing code base and experience in using the various sensors and data storage options. Our app relies heavily off of data storage so both of us became familiar with all the data storage techniques, with the majority of our focus on SQLite. Starting with no code for a feature is very difficult and it's very helpful to have a pre-existing code base. All around we have become more familiar with RecyclerViews and their adapters to achieve all the listing functions across the different pages. We've also learned that accurately detecting tilt actions is very difficult to make very precise. We also learned that dialogue boxes are a much more elegant solution than separate screens for actions that would require a screen with very little content.