How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: “**Capstone_Stage1**”
3. Replace the text in green

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”

GitHub Username: bwshockley

Heck of a Score

Description

Heck of a Score can simplify your “Heck of a Game” scores. Users keep track of bids, scores, and can even review past games and winners. Select the number of players from your existing player list, or create a new player for that game. Enter and keep track of bids for each round of play and optionally save the results of the game, adding to your player stats - totals, best game, average game scores. The streamlined user interface flows through the game play to minimize app interaction and maximize play time with friends and family.

Intended User

The intended user is families and friends who like to play “Heck of a Game” and want an easy to use score keeping application as well as some bragging rights by reviewing player stats and past games.

Features

Create/Edit/Remove players from the game. Easily add with just a name - keep track of player stats. Start a game by choosing players and placing your first bids.

Game play is saved so that if you exit the app mid game, you can continue where you you were in the game.

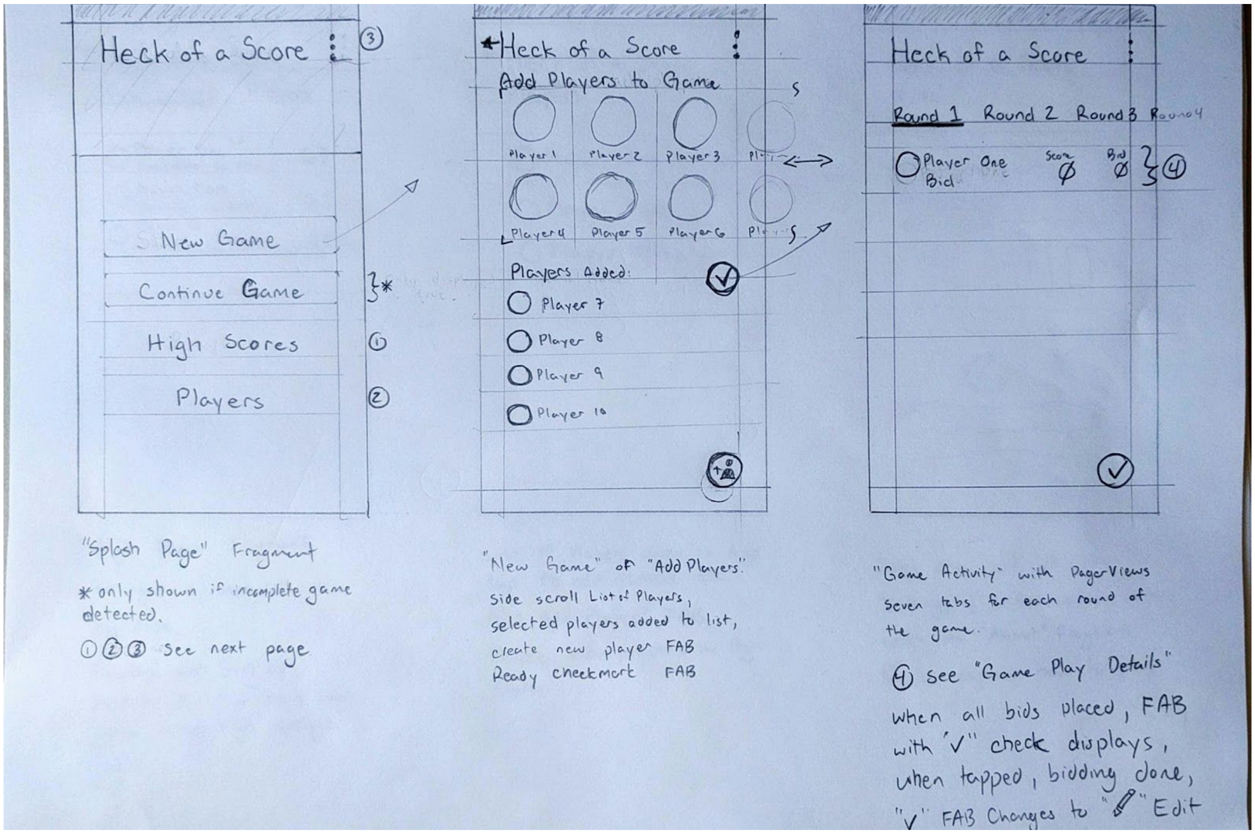
Check past games.

Review the rules of a “Heck of a Game”.

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screens 1-3



"Splash Page" Fragment

* only shown if incomplete game detected.

①②③ See next page

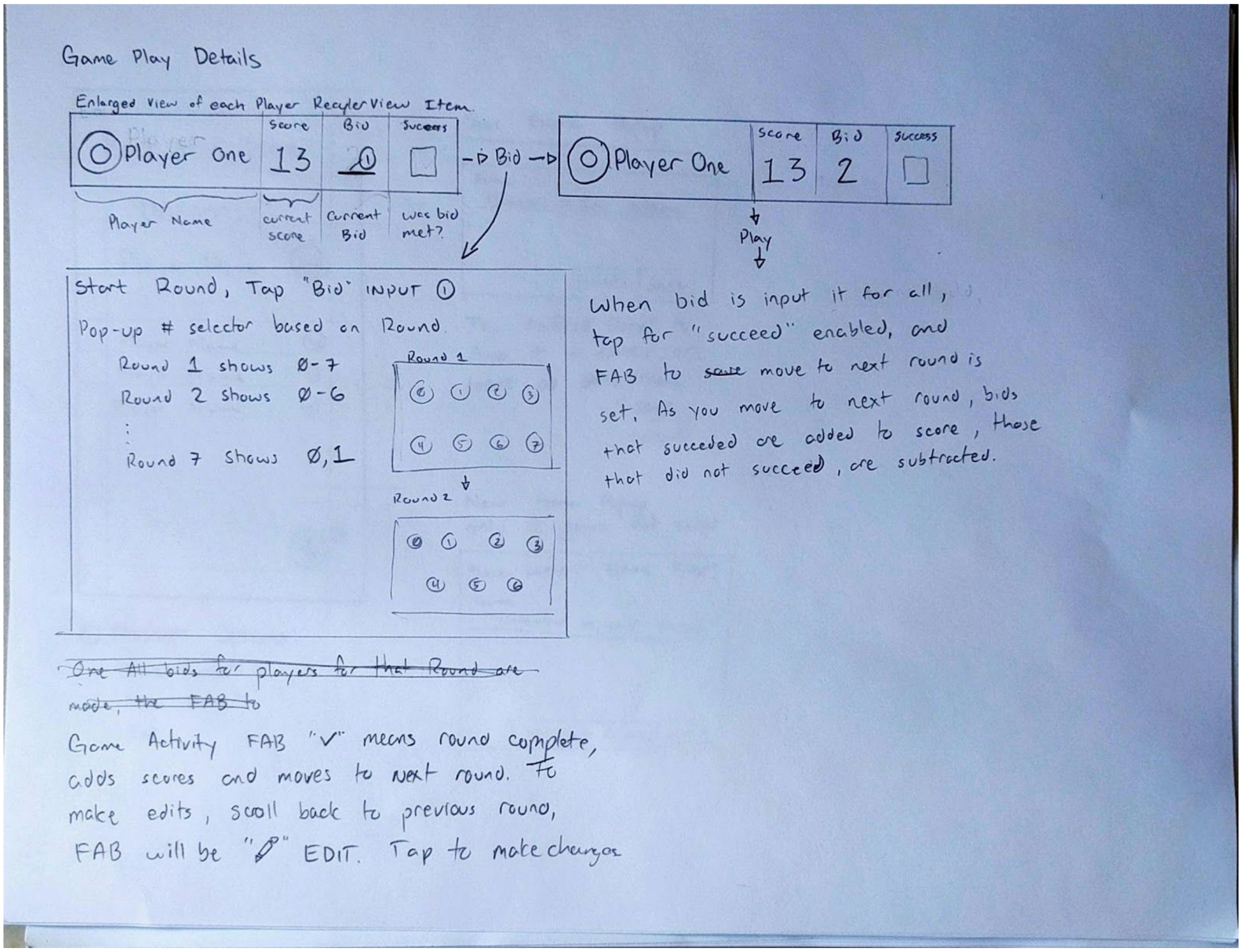
"New Game" of "Add Players".
Side scroll List of Players,
selected players added to list,
create new player FAB
Ready checkmark FAB

"Game Activity" with PagerViews
Seven tabs for each round of
the game.

④ see "Game Play Details"
when all bids placed, FAB
with "√" check displays,
when tapped, bidding done,
"√" FAB Changes to "✎" Edit

These are the main game play screens. Users start at the splash screen and can start a new game from there. Or, if an incomplete game is detected, they may continue the game. Next they select the players for the game or add a new player to the player list. Then they are in the pageview tabs for the 7 rounds of the game. Players move through the “Game Play Details” activity and tabs from one round to the next, entering bids and taking credit for success. The “Check” FAB does not appear until all bids are placed. By selecting the FAB - users indicate they are done with that round - and are moved to the next round. At the seventh round, the FAB ends the game, bringing users to the final fragment. (see below). If at any point a change or correction is needed to a prior round, users can swipe left/right to get to that round and edit the information on that tab for that round.

Additionally, details for each players recyclerview list item is shown below. The left side shows the player name, on the right is the current score, bid, and success checkbox. The bid and checkbox are user inputs. The bid input allows for predetermined data only. For instance in the first round of play the bid may be anything from 0-7, while on the last (7th) round of play the bid can only be 0 or 1. The success criteria checkbox will most likely be checked by default indicating the player met their bid goals. The players can uncheck if the bid goal was not met.



Game Play Details

Start Round, Tap "Bid" INPUT ①

Pop-up # selector based on Round.

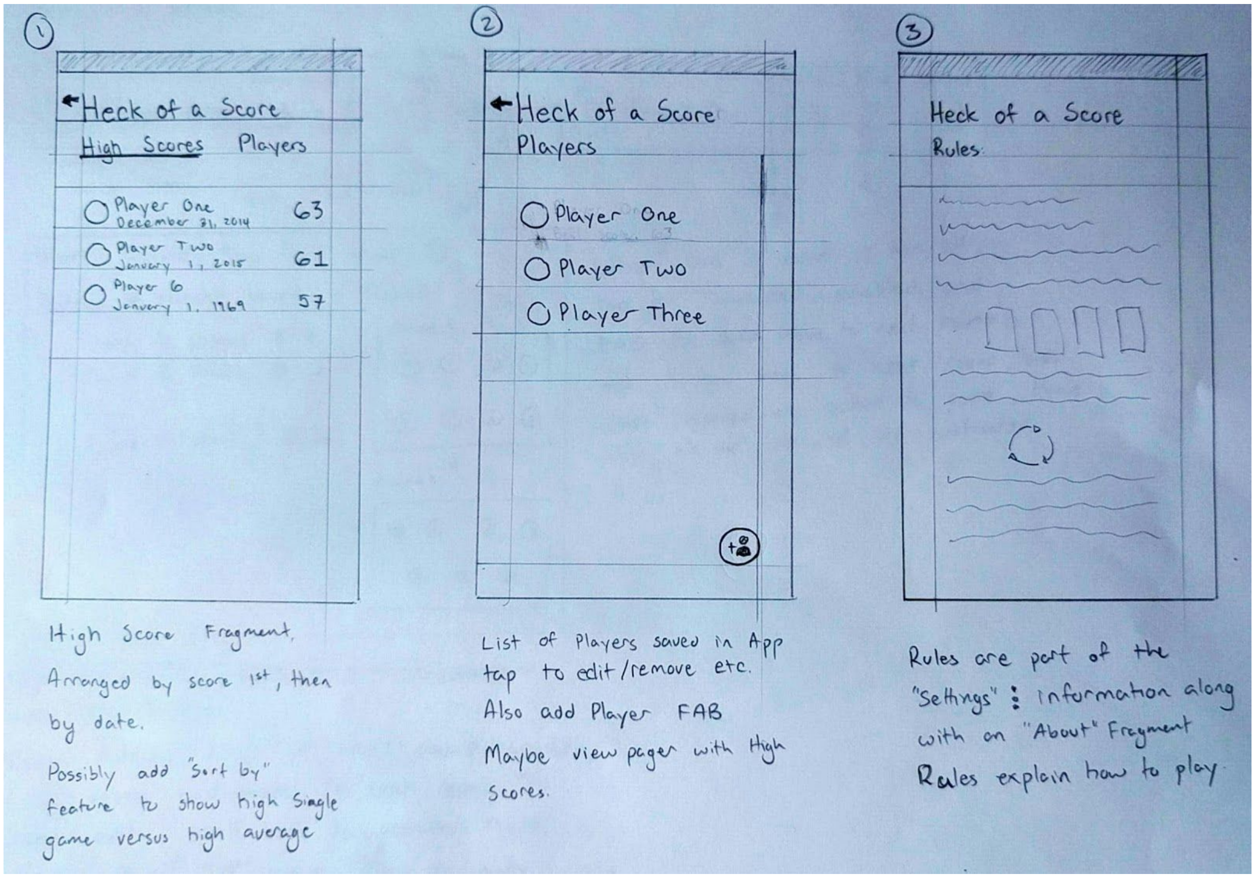
Round 1 shows Ø-7
Round 2 shows Ø-6
:
Round 7 shows Ø,1

When bid is input it for all, tap for "succeed" enabled, and FAB to ~~save~~ move to next round is set. As you move to next round, bids that succeded are added to score, those that did not succeed, are subtracted.

~~One All bids for players for that Round are made, the FAB to~~

Game Activity FAB "✓" means round complete, adds scores and moves to next round. To make edits, scoll back to previous round, FAB will be "✎" EDIT. Tap to make changes

Screens 4-6



High Score Fragment, Arranged by score 1st, then by date.

Possibly add "Sort by" feature to show high Single game versus high average

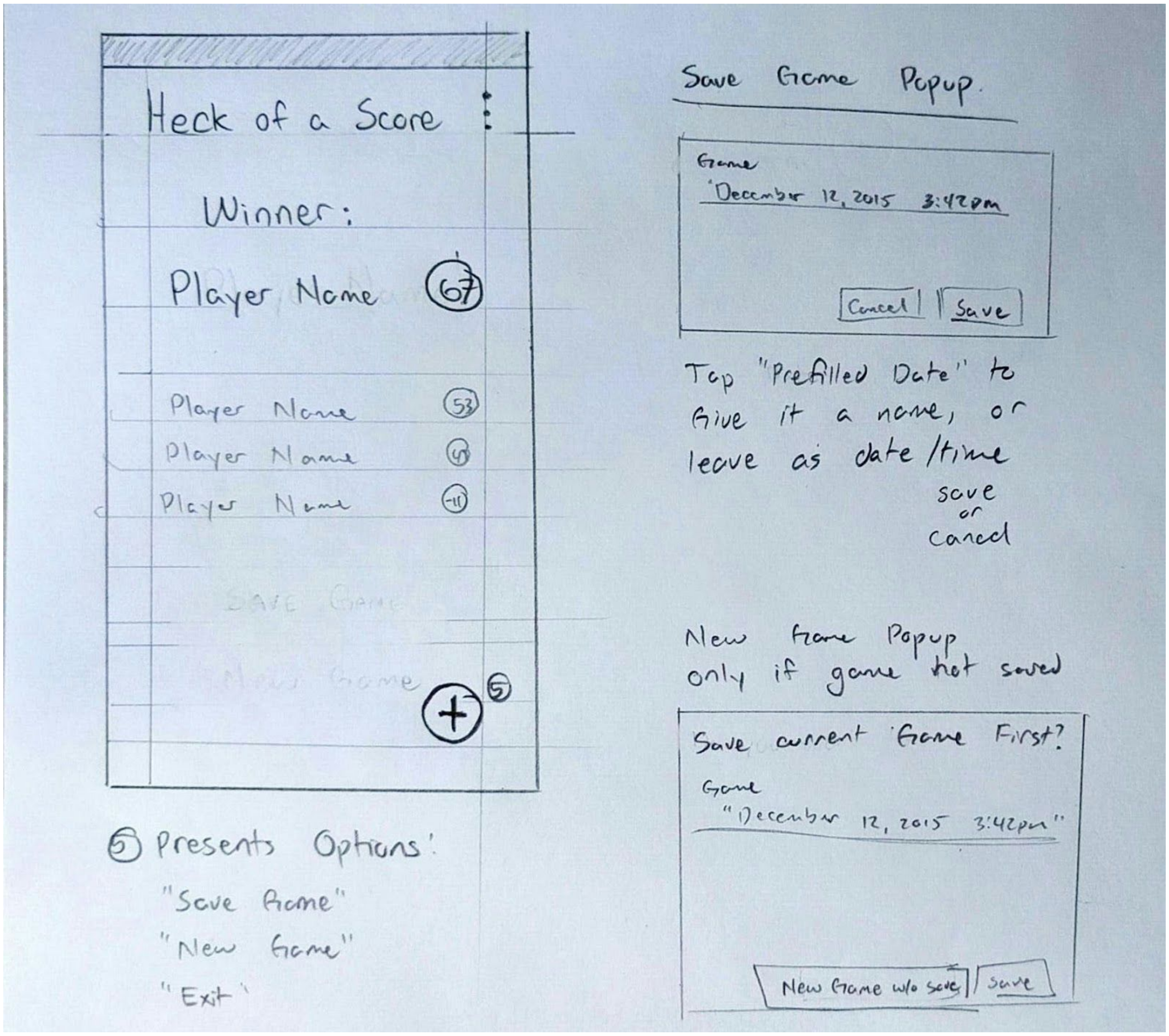
List of Players saved in App tap to edit/remove etc.
Also add Player FAB
Maybe view pager with High Scores.

Rules are part of the "Settings" ⋮ information along with an "About" Fragment
Rules explain how to play.

Supplement screens: Users can check player high scores from the main splash screen as well as enter/edit/remove players from the player list from the main screen as well. The high score screen will be simple at first, showing just the highest game score of all the saved games for the top ten scores. But, it could easily be adopted to show highest average score, best game (total score of all users), worst game, etc.

The players list screen will show the list of players saved in the app. Tapping a player will allow the user to edit the information or delete/remove the player. Also, new players can be added from this screen.

The “Rules” screen and “About” screens can be accessed at any point in the app through the menu in the upper right. The Rules describe game play and rules of the game, while the about page will have a short application description.



Final - Winner Screen
The final screen displays the results of the game, including the winner and their score. From here the users can tap the FAB and either "Save Game", "New Game", or "Exit" back to the home screen. Games may be saved by default - haven't decided yet. I will also add a share action to share the winner information to friends. :)

Key Considerations

How will your app handle data persistence?

Data will be written to the database as entered, such that at any point, if the user leaves the app, they can continue the game where they were. Any other changes such as screen rotations will be handled in the typical fashion via saved instance state.

Describe any corner cases in the UX.

If the users exit the application mid game, data will be saved in it’s current state, such that the user can continue where they were.

Users can return to a previous round to fix errors in either the bid amount or the success of that round.

Describe any libraries you’ll be using and share your reasoning for including them.

I plan on using butterknife to bind views and possibly resources. I will use the Google admob and analytics libraries to connect to produce a ad in my application as well as track usage of the app and see where improvements can be made.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup - backend

I will plan out all data structure first - how I plan to share between activities and fragments.

You may want to list the subtasks. For example:

- Configure the Google Play Services Libraries and all required information
- Configure the Butterknife library
- Setup the database structure, and all supporting data such as the contract and content provider, etc.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for Splash Screen
- Build UI for selecting players, adding, editing, and removing players
- Build UI for main game play screen with 7 tabs.
- Build UI for secondary screens:
    - High Score

    - List of Players
    - Rules
    - About
- Any remaining UI elements not yet implemented.

Task 3: Backend and Frontend Integration

I will tie together the backend data with the UI to work through basic game play scenarios. I will tease out errors and any other use cases not previously imagined.

Task 4: Tie in Admob

Create an interstitial ad that is displayed at the end of a game.

Task 5: Tie in Analytics

I will determine what data might be most useful to me as the developer.
The add in any additional analytics data and requirements, and check the reports online to ensure they are providing the data needed.

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”