

Distinction Assignment

Introduction

I created an application to aid in playing a card game by the name of The Resistance: Avalon. Using this application will allow people to be privately assigned characters that they will be able to use in the game. The game re-enacts good guys vs. bad guy scenarios, where the bad guys know who is on their team but it is up to the good guys to find out who is on theirs and out the bad guys. The application will allow users to choose settings for the game, input their names and view the rules. The app will randomly assign a character to each person, and they will see a description of the role of their character.

Design Communication Perspective

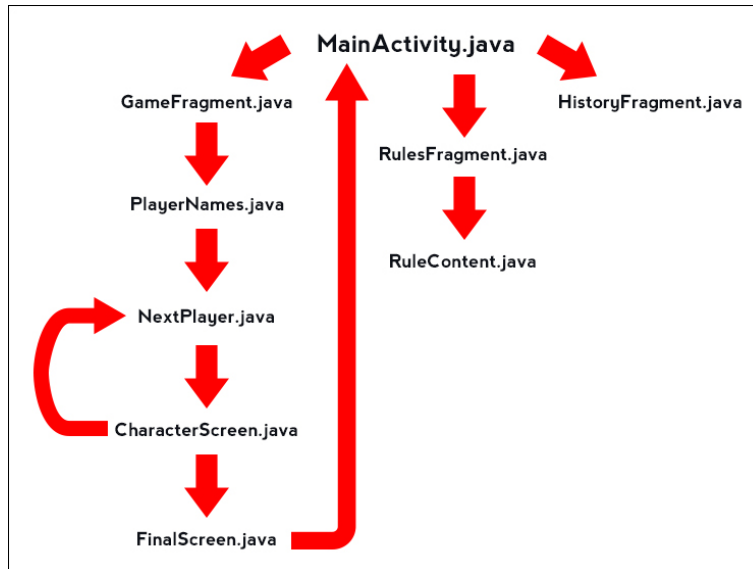
The application now eliminates the need of its original cards, and it facilitates using just a simple deck of cards (enabling someone to play a game on the go).

The first navigation only consisted of one tabbed menu that indicated "Home" and "Rules". These two initial screens utilized fragments and the tabs were used as a global navigation. Within the Rules page there is a sub navigation that is displayed as a list. When clicking on these sections, a new activity would open up and the back-button in the navigation bar was turned on so that if the user clicked it, they would be returned to the Rules' list screen.

I wanted to create a similar type navigation for the Game setup section, however I had something else to consider. Passing the application around to friends is great for secrecy, but it easily allowed for people to cheat (by continuously press back on the application to see who has been assigned). For this reason, I disabled the `onBackPressed` within the `NextPlayer.class` so that users are unable to see the previous player's character. Ultimately there are still ways to cheat, but this thwarted the initial cheaters.

```
@Override
public void onBackPressed() {
    // Prevent using back button
}
```

I eventually did add a home button (pseudo back button) to the Action Bar. A minor usability test in the early stages of the interface showed that users sometimes miss-entered content or wanted to restart the configuration. The only way to get back to the main activity was to repeatedly press back. This becomes tedious when you have 10 players and 20 screens to navigate (and also impossible with the `onBackPressed` function activated).



Class Model

The flow of this application consists of navigating the configuration screen, adding players and then looping through each player to assign them random characters. Once all the players receive their characters, the loop ends and brings the user to a final screen indicating the selection process is over. It then asks who won the game. Upon clicking either "Minions of Mordred" or "Servants of Arthur", an Object is sent to Parse.com which indicates who won, the players in the game, and the date the game ended.

objectId	String	players	String	winner	String	updatedAt	Date
UNDgmcws0p		Eric was Servant of Arthur, Angel was Minion of Mordred, Jess was Morgana, Gabe was Oberon, Sam was Merlin, Dan was Servant of Arthur, Aaron was Ser...		Mordred		Nov 21, 2014, 01:12	
6ZKq5HQNh8		Sam was Servant of Arthur, Audrey was Percival, Dan was Minion of Mordred, Gabe was Merlin, Sophie was Mordred,		Arthur		Nov 21, 2014, 01:11	
RZ7Edj6M1		was Minion of Mordred, was Merlin, was Servant of Arthur, was Minion of Mordred, was Servant of Arthur,		Mordred		Nov 20, 2014, 15:35	
codzrH9sHm		was Minion of Mordred, was Servant of Arthur, was Minion of Mordred, was Merlin, was Servant of Arthur,		Mordred		Nov 20, 2014, 13:09	
jFL19VORDh		Sam was Servant of Arthur, Sophie was Minion of Mordred, Gabe was Minion of Mordred, Dan was Merlin,		Mordred		Nov 20, 2014, 12:59	
NiHo021nuV		Sam was Servant of Arthur, Sophie was Servant of Arthur, Angel was Minion of Mordred, Gabe was Minion of Mordred, Dan was Merlin,		Mordred		Nov 20, 2014, 12:29	
ecroOV4hlo		Dan was Servant of Arthur, Angel was Percival, Gabe was Minion of Mordred, Sam was Merlin, Sophie was Minion of Mordred,		Mordred		Nov 20, 2014, 12:24	
895FD0R2GC		dan was Servant of Arthur, sam was Merlin, Jose was Minion of Mordred, soph was Servant of Arthur, gabe was Minion of Mordred,		Mordred		Nov 20, 2014, 10:34	

Data Table of content uploaded to Parse

```

Button mordredWon = (Button) findViewById(R.id.mordredWon);
Button arthurWon = (Button) findViewById(R.id.arthurWon);

mordredWon.setOnClickListener((v) -> {
    ParseObject recentWins = new ParseObject("RecentWins");
    recentWins.put("winner", "Mordred");

    String playerString = "";
    for(int i = 0; i < playerNum; i++) {
        playerString += playerArray[i] + " was " + characterArray[i] + ", ";
    }
    recentWins.put("players", playerString);
    recentWins.saveInBackground();

    Intent intent = new Intent();
    intent.setClass(getApplicationContext(), MainActivity.class);
    startActivity(intent);
});
  
```

Code Snippet of Data to be sent through ParseObject

The only limitations I ran into was the ability to host content on a server (by manually importing it, not uploading it VIA the application then re-downloading the same content), I tried to use Parse for this but I was not altogether sure of their hosting capabilities of raw data and not just CSV files.

The other issue I ran into was String switch cases. After having done some research, Android Studio is meant to be used with the 1.6 JDK, and I have 1.8 JDK installed. Because of this, Android was refusing to allow Strings as the switch comparison. However, I was able to rectify this by adjusting the compile options in the build.gradle file.

```
compileOptions {  
    sourceCompatibility JavaVersion.VERSION_1_7  
    targetCompatibility JavaVersion.VERSION_1_7  
}
```

Summary

In my original design I wanted to work with concatenating audio files to replace the narration required for this game. It removed the need for one specific player to read out the script and not give away their own assigned character. However, I believed this would not have sufficient activities to fulfil the requirements of the project. The application was then modified to still modify the narration, by removing it altogether. In the future, I think I will work on making this application give the option to either narrate the roles for the players, or remain as is; a silent narrator.

Unfortunately, I was unable to retrieve the data from Parse to display it in the history tab of the application. The data was being successfully sent, but I was unable to query the class and retrieve all the objects to display them. This was a final addition to the application and was the last on the list of things to concentrate on, so it unfortunately didn't receive enough attention that it should of.

Overall, I achieved all but one of the goals I set out in completing this application. The aesthetics were not exactly as I planned, but the overall structure did remain the same from beginning to end.

References

I used the Parse API to send data from my application to the network.
(https://www.parse.com/docs/android_guide)

Installing SDK:

```
dependencies {  
    compile 'com.parse.bolts:bolts-android:1.1.2'  
    compile fileTree(dir: 'libs', include: 'Parse-*.jar')  
}
```

Connecting application to Parse:

- 1- Call `Parse.initialize` from the `onCreate` method of your `Application` class to set your application id and client key
- 2- Your app must request the `INTERNET` and `ACCESS_NETWORK_STATE` permissions, if it isn't doing so already. Add these lines before the `<application>` tag in your `AndroidManifest.xml`

Saving Objects:

```
ParseObject gameScore = new ParseObject("GameScore");  
gameScore.put("score", 1337);  
gameScore.put("playerName", "Sean Plott");  
gameScore.put("cheatMode", false);  
gameScore.saveInBackground();
```

Stack Overflow was a huge help with various edits from different questions. In one case, I collected the most useful answers and compiled them into a new question:

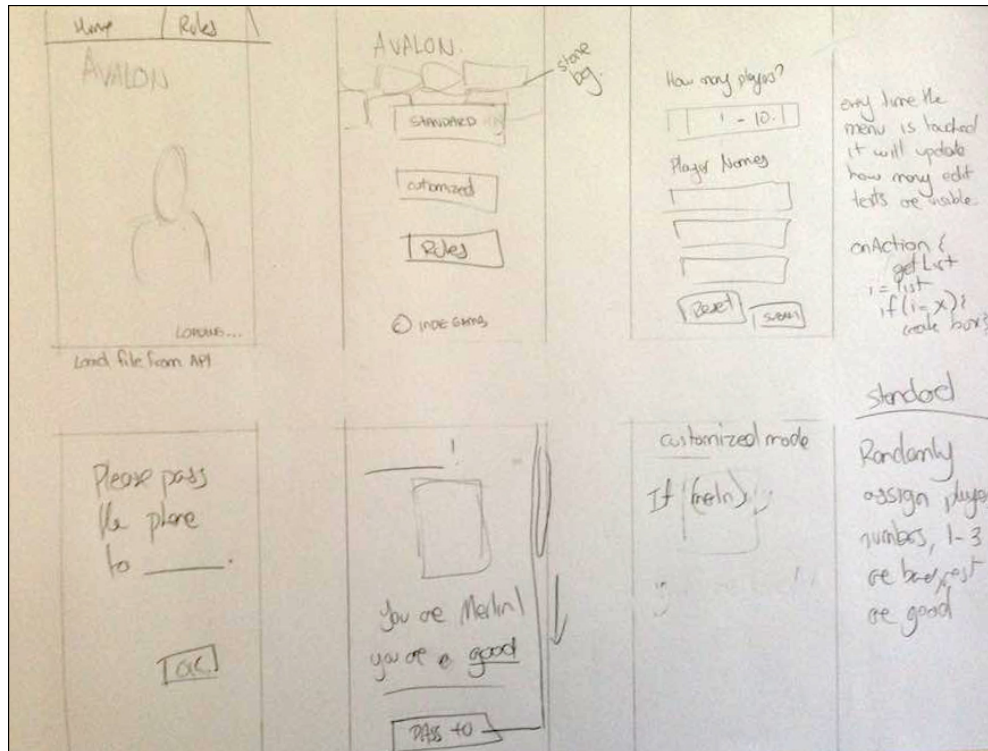
Can't load FragmentManager:

<http://stackoverflow.com/questions/26856200/cant-load-fragmentmanager/26856201#26856201>

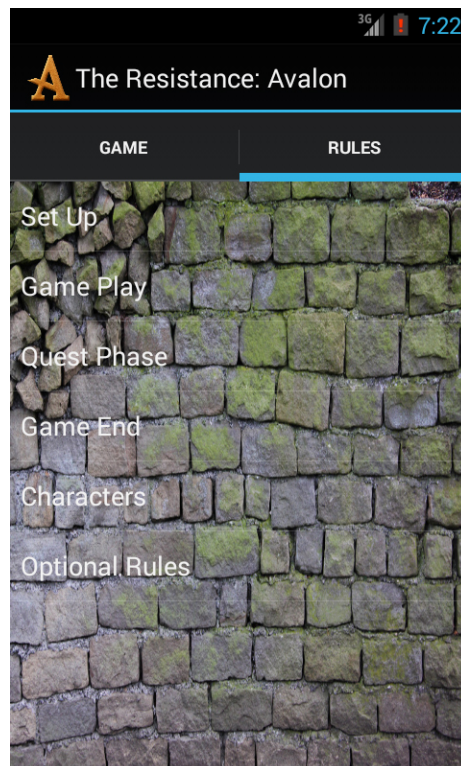
Using Android Studio with Java 1.7:

<http://stackoverflow.com/questions/16636146/using-android-studio-with-java-1-7/21622710#21622710>

Appendix



Initial Sketches








Preliminary Design

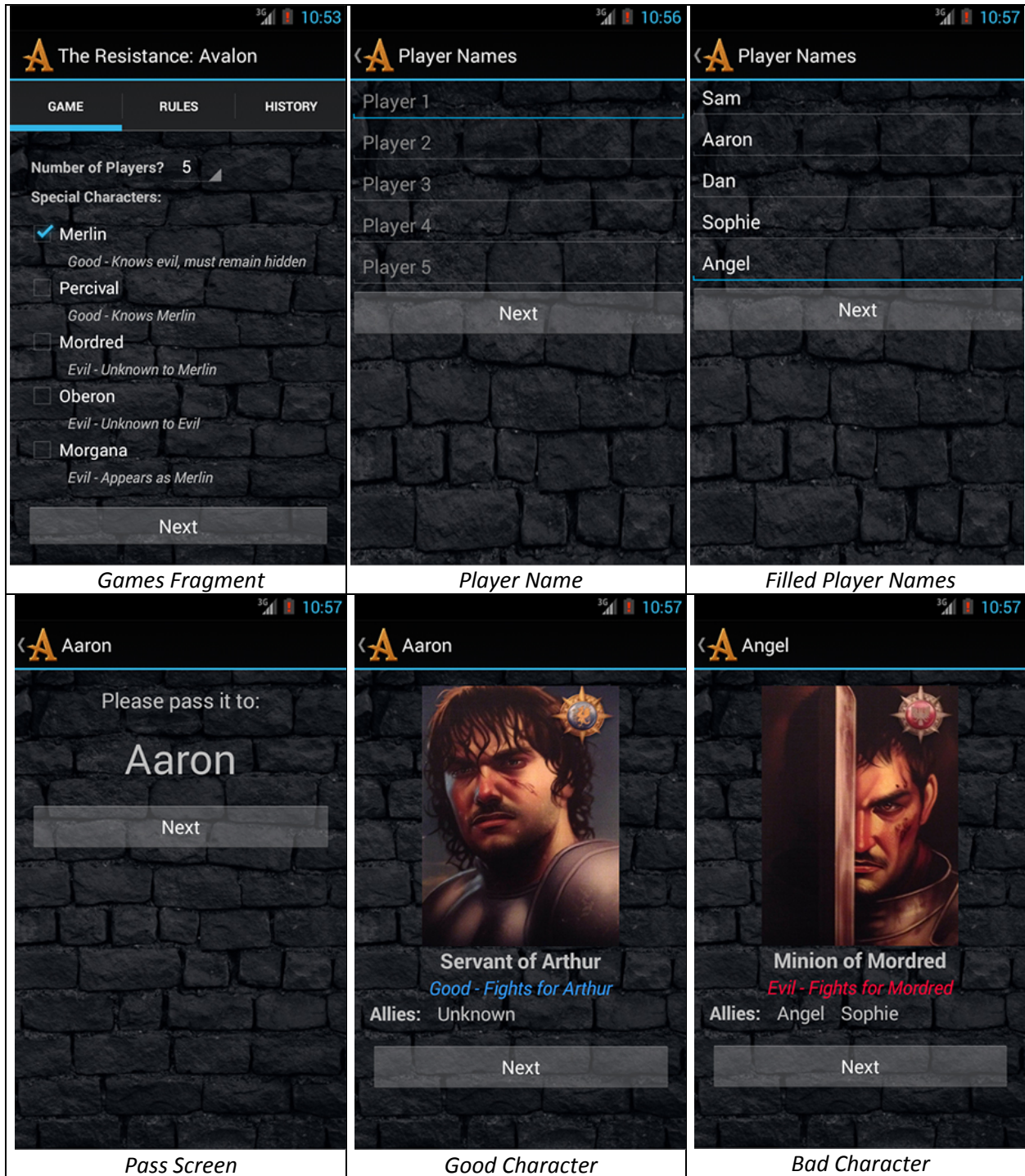

```

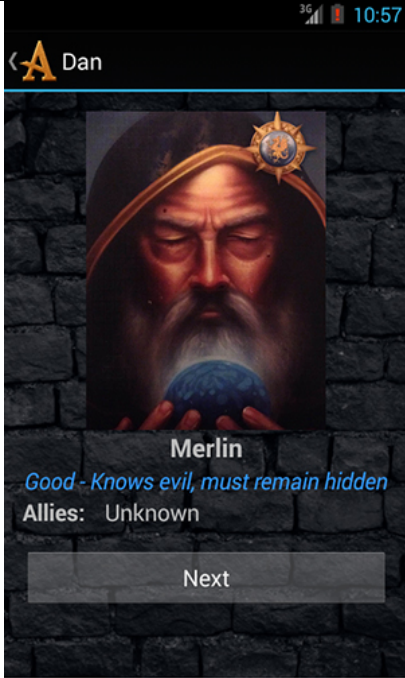
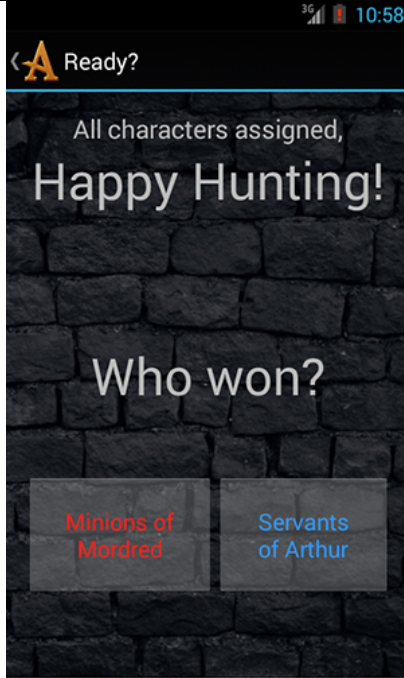
849-849/resistance.avalon I/System.out: Player Sam is Merlin
849-849/resistance.avalon I/System.out: Player Andrew is Minion of Mordred
849-849/resistance.avalon I/System.out: Bad Guy: Player Andrew
849-849/resistance.avalon I/System.out: Player Sophie is Servant of Arthur
849-849/resistance.avalon I/System.out: Player Cassie is Minion of Mordred
849-849/resistance.avalon I/System.out: Bad Guy: Player Cassie
849-849/resistance.avalon I/System.out: Player Dan is Servant of Arthur
849-849/resistance.avalon W/EGL_emulation: eglSurfaceAttrib not implemented

```

Use of Logs to determine if data was being sent/randomized

<p> The Resistance: Avalon</p> <p>3G 10:53</p> <p>GAME RULES HISTORY</p> <p>Set Up</p> <p>Script</p> <p>Game Play</p> <p>Quest Phase</p> <p>Game End</p> <p>Optional Characters</p> <p>Optional Rules</p> <p>Credits</p>	<p> Set Up</p> <p>3G 10:53</p> <p>Select the tableau corresponding to the number of players. Place the tableau in the center of the play area with the Score Markers, Team tokens and Quest cards adjacent to the tableau. Place the Round marker on the tableau's 1st Quest Space. Given each player a set of two Vote tokens.</p> <p>Randomly select a Leader; the leader receives the leader token. Use the chart below to determine the number of Good and Evil players.</p> <p>Shuffle the appropriate number of good Carter's cards (One of these cards will be the Merlin card, all the other good characters card will be just 'loyal servant of Author' cards) and Evil Character cards (one of these cards will be the Assassin card, all the other Evil Character cards will be just "Minion of Mordred" Cards). Deal one card to each player face down. Each player secretly looks at their assigned loyalty on their Character Card.</p> <p>Evil reveals itself, and Merlin looks into the future</p> <p>True Evil is rampant in the land. Arthur represents</p>	<p> Script</p> <p>3G 10:54</p> <p>Everyone close your eyes and extend your hand into the fist in front of you</p> <p>Minions of Mordred, not Oberon - open your eyes and look around so that you know all the agents of Evil</p> <p>Minions of Mordred close your eyes</p> <p>All players should have their eyes closed and hands in the fist in front of them</p> <p>Minions of Mordred, not Mordred himself – extend your thumb so that Merlin will know of you</p> <p>Merlin, open your eyes and see the agents of Evil</p> <p>Minions of Mordred – put thumbs down and reform your hand into a fist</p> <p>Merlin, close your eyes</p> <p>All players should have their eyes close and hands in a fist in front of them</p> <p>Merlin & Morgan - extend your thumb so that Percival may know of you</p>
Rules Fragment	Set Up	Script
<p> Game Play</p> <p>3G 10:54</p> <p>The game consists of several rounds; each round has a team building phase and a quest phase. In the Team Building phase the Leader proposes a Team to complete a Quest – all the players will either approve the proposed team and move to the Quest phase, or reject the proposed Team passing leadership to the next player and repeating the process until a Team is approved. In the Quest phase those players selected to be on the Team will determine if the quest is successful.</p> <p>Choose your team wisely. Approve only teams where you trust everyone. Even a single evil player on the team is enough for failure.</p> <p>Team assignment: After appropriate discussion, the leader takes the required number of team tokens and assigns each team token to any player.</p> <p>The leader can be on the Team, but it is not required to be so. Note a player may only be assigned one team token.</p> <p>Discuss, discuss, discuss. All the players should participate in helping the Leader make the right Choice of Players to be on the Team. Active and logical discussion is a great way to catch</p>	<p> Optional Characters</p> <p>3G 10:56</p> <p>Game play</p> <p>Four additional Characters with special powers are available to play. You can play with these cards in any combination that you would like. Different combinations will make the game harder to win for one of the side of the battle or the other. It is best to add one special Character card into a game at a time, adding more or changing when you are familiar with how they play. In most cases you will want to play with Merlin, but it is not required.</p> <p>Percival</p> <p>Percival is an optional Character on the side of Good. Percival's special power is knowledge of Merlin at the start of the game. Using Percival's knowledge wisely is key to protecting Merlin's identity. Adding Percival into a game will make the Good side more powerful and win more often.</p> <p>Mordred</p> <p>Mordred is an option Character on the side of Evil. Mordred's special power is that his identity is not revealed to Merlin at the start of the game. Adding</p>	<p> Credits</p> <p>3G 10:56</p> <p>I take no credit in designing this game, content owned and distributed by Indie Boards & Cards.</p> <p>Game Design: Don Eskridge Game Development: Travis Wrothington Graphic Design/Illustration: Luis Francisco, George PAtsouras, Marco Turini, Pablo Messuti, Luis Tomas</p> <p>www.IndieBoardsAndCards.com</p>
Game Play	Optional Characters	Credits



 <p>Optional Character</p>	 <p>Final Screen</p>	
---	--	--