AwesomeCoder, Inc

Software Design Document

Quick Charades

- **Tony Lim** atomictheorist@gmail.com
- **Joseph Park** josephpkc@gmail.com
- **Andrew Martinez** andrewman34@gmail.com
- **Christian Sydow** swert7662@gmail.com
- **Josue Segura** segura.josue65@gmail.com

Github project link : https://github.com/Josue3205/QuickCharades

Project javadocs : http://josue3205.github.io/QuickCharades/

Version 1.0

Oct 5, 2014

Table of Contents

1. Introduction

The purpose of the project is to implement a multiplayer game for android platform that is entertaining and educative. The topics covered used in the project includes:

- Android Development
- Web Services
- Relational Database
- XML Configurations
- User Interface Design
- GooglePlay (optional)

1.1. Goals and Objective

The program implements a guessing game that involves a maximum of 8 players. In detail, the game will ask each player to guess what the picture displays. For example, if the picture displays basketball star player Kobe Bryant, then the user will have to enter Kobe Bryant in the provided text box. The primary purpose of the game is to create a light-weight and easy multiplayer games that is suitable for all range of demographics.

To do this, the effort needed to understand the game mechanism has to be as minimal as possible. These can be done by creating a concise user manual and tooltips in the game. Additionally, the game has to feel natural to people of all ages. Accordingly, the final product must be quick, efficient, and extremely easy to use. It must offer useful features without overwhelming the user with options. The user interface must be intuitive and have little or no learning curve.

The gameplay will be simple: the player will have to guess what the image is using the letters given to them. These letters will of course include the correct ones that spell out the image name, but it will also include randomized incorrect letters. It is up to the player to figure out which letters are correct and in what order. All of this needs to be done within a certain time limit. There will be a timer that counts down at the beginning of each question and the player will have to figure out the answer before the timer is up. Once the timer is up, the player will no longer get any points from that question. The player will have access to a skip button, if he/she chooses that the current question is too difficult. The player will also have access to power ups to boost their own score, or to help with a question. These power ups can be obtained through streaks, which are successful consecutive correct answers. Whenever the player successfully answers a question, their streak will go up. However, when the player misses a question or skips, their streak will reset. Once a certain streak is reached, a power up is then added to the players' arsenal. Two players can also go head-to-head in the online mode and compete with each other to see who can score more points or who can guess the questions faster. These scores will then be added to the player's overall stats and be adjusted for their ranking in the leaderboards.

When the player first boots up the game, he/she will have several options: playing the game, adjusting the settings, checking up on friends, and looking through the leaderboard. The leaderboard option will show the player's current ranking in the leaderboard. The friends list will bring up all of the friends the player has added, as well as the option to add and remove friends. The settings option allows the player to adjust brightness, volume, or their in-game name. Finally, the play option allows the player to play the game.

When the player presses on the play button, he/she will be given a few choices: what game mode to choose from (online with friend, online with random, or single), what difficulty the questions are, and what category the images will be from. Each question (and their images) will have a certain difficulty and category that the player can choose from.

The player will have an online account that is automatically assigned a unique player ID. This is tied to the device. The player will also have a unique name that they can change at any moment.

1.2. Project Overview and Scope

The *QuickCharades* game is composed of several; primary components: a client-side application that receives user's input and performs calculations, and a server-side application which loads and select categories of pictures and updates and synchronizes scores to the server. The game features can be broken up into two groups as well: core features, which are essential to the function of the application, and additional features, which are only meant to add extra functionality.

Bullet Points Descriptions

1.2.1. Core Features

1.2.1.1. Displays user’s name and current highscore

- Displays following buttons below player info:
    - Image Category (see section 1.2.1.2 for details)
    - Options (see section 1.2.1.4 for details)
    - Start Game Against Friend
        - Allows user to choose a friend to play against
    - Start Game with Random Opponent
        - Connects user to a random player
    - Leaderboards (see section 1.2.1.3 for details)

1.2.1.2. Categories of picture

- Contains different groups of pictures that user’s can choose from (Ex: Superheroes, Sports, etc.)
- Displays chosen category during gameplay
- Divides pictures by difficulty; predefined in database

1.2.1.3. Leaderboards

- Section where users can compare their highscore with other players
    - Local
        - Display user’s highscore compared to their friends
    - Global
        - Display user’s highscore compared to all players

1.2.1.4. Options

- Section where users can access game options
    - If user is in game, there will be a button that will allow them to go back to the main menu
    - Volume Controls
        - Users can change their devices volume
    - Brightness
        - Users can change their devices brightness

  Exit Button
        - Users can end their game, if they do not want to finish their current game

1.2.2. Additional Features

1.2.2.1. Skip buttons (will reduce players current points if used)

- A button specifically for the user to move on to the next image without answering the current image.
- Used when user does not know the answer to the picture.

1.2.2.2. Multiplier/Streak

- Keeps count of the amount of pictures answered correctly in a row without getting one wrong.
- Used to declare user's power ups.
- At certain streaks (e.g. 5 in a row, 10 in a row), player gets a random power up. The type of power up depends on the level of the streak (e.g. 5 and 10 in a row get relatively small power ups (Tier 1), and 20 to 30 in a row get larger power ups (Tier 2 -3)). These power ups are chosen at random from each tier pool depending on the streak.

1.2.2.3. Friends List

- Section where users can see their online friends
- Send game invites
  - Users can send invites to their friends
- Delete friends
  - Users can delete no longer wanted friends
- Add Friends
  - Users can increase their friends list by sending them a request

1.2.2.4. Power Ups

- Special abilities/advantages that user's get for answering images correctly
- As stated in the multiplier/streak section (1.2.2.2), user's will gain better power ups if they increase the amount of images answered and do not get an image wrong
- Players will not lose power ups if streak is lost
- Will not roll-over to other games

1.2.2.5. Sample Power Ups

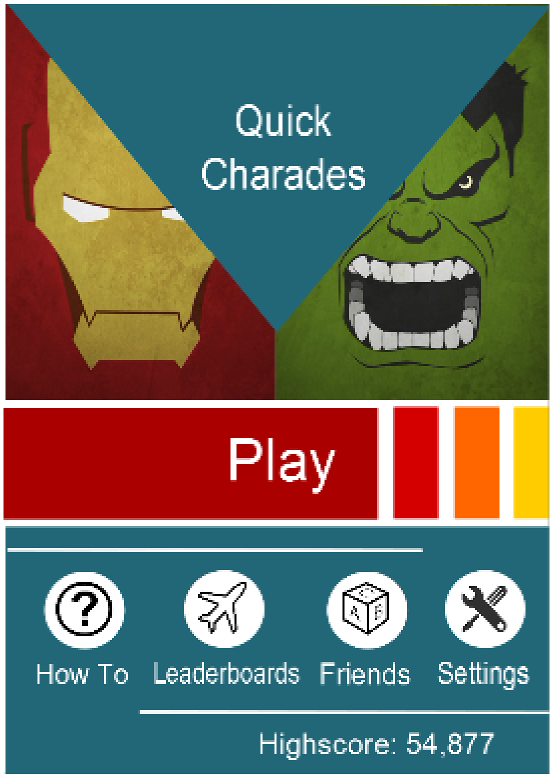
- Reverse Image: this power up causes the image displayed to reverse
  - Sample of Tier 1 power up
- Input Delay: this power up causes a latency when a player inputs a letter; will take longer to input answer
  - Sample of Tier 2 power up

- Black Out: this power up causes an opposing player's screen to turn black
  - Sample of Tier 3 power up
- Fade: this power up causes the image to fade after a certain time
  - Sample of Tier 4 power up
- Auto-Error: this power up causes an opponent to automatically get their current image wrong
  - Sample of Tier 5 power up

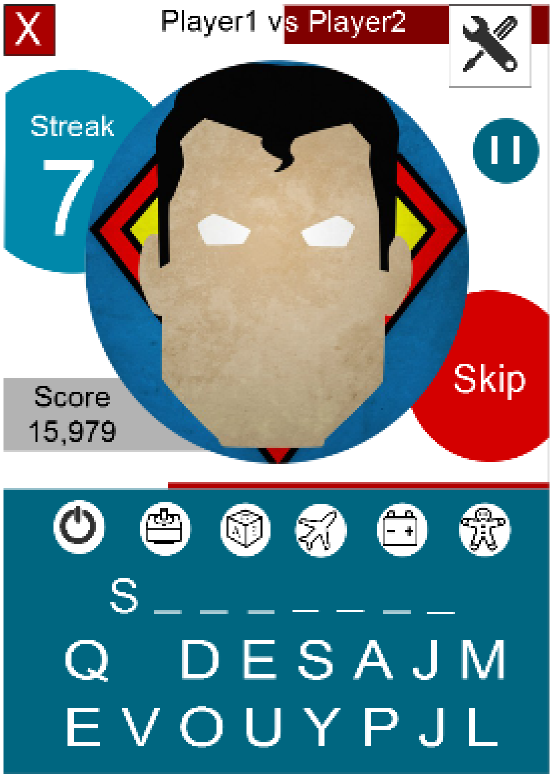
2. Interface Design

The interface implementation should be simple but understandable. Below are the first versions of the interface implementation of the game.

Example of the login screen design:



Example of in-game interface:



4. List of all Objects and Data from Requirement

- Player (User)
- Image
- Question
- Answer
- Letter
- Correct Letter
- Incorrect Letter
- Image Category
- Image Difficulty
- Game Mode
- Friend
- Time Remaining
- Friends List
- Categories List
- Game Modes List
- Timer
- Skip
- Power Ups
- Streak
- Current Score
- Highest Score
- Win/Loss Ratio
- Time Elapsed
- Leaderboard (Rankings List)
- Player Rank
- Volume
- Brightness
- Player Name
- Image Name
- Player ID
- Power Up Streak
- Main Menu
- Categories Menu
- Game Mode Menu
- Options Menu

5. Grouped Objects with Data and Functions

1. Player (User)

- the Player Class which holds all of the data relevant to the user/player

Data	Methods
○ Player ID ○ Player Name ○ Player Stats ■ Highest Score ■ Highest Streak ■ W/L Ratio ■ Wins ■ Losses	○ Get Name ○ Get Stats ○ Get Current Score ○ Get Current Streak ○ Get Status ○ Get Friends List ○ Get Rank ○ Update Name

○ Current Score ○ Current Streak ○ Settings Preference ■ Volume Setting ■ Brightness Setting ○ Status (Online/Offline/Busy/In-game) ○ List of Friends ○ Leaderboard Rank	■ New Name ○ Update Stats ■ New Stats ○ Update Current Score ■ New Current Score ○ Update Settings ■ New Settings ○ Update Status ■ New Status ○ Update Friends ■ New Friend ■ Friend to Remove ○ Update Rank ■ New Rank ○ Update Current Streak ■ New Streak ○ Reset Rank ○ Reset Score ○ Reset Stats

2. Question

- the Question class which holds the image class, answer in text, and randomized letters set

Data	**Methods**
○ Image (Object) ○ List of Letters ■ Correct Letters ■ Incorrect Letters	● Methods: ● Display Question ● Get Image ● Get Letters List ● Add Letters ○ New Letter ● Remove Letters ○ Letter to Remove ● Randomize Letters ○ Number of Letters to Randomize ○ Bounds of Letters ● Check if Answer is Correct

	○ Answer

3. Image

- the Image class holds the image

Data	Methods
○ Image URL ○ Image Name ○ Image Difficulty ○ Image Category	○ Get URL ○ Get Name ○ Get Difficulty ○ Get Category ○ Set URL ■ New URL ○ Set Name ■ New Name ○ Set Difficulty ■ New Difficulty ○ Set Category ■ New Category ○ Display Image

4. Timer

- The object is used to display, store and be responsible for the screen timer that will be displayed as the question is asked to the player

Data	Methods
○ Time Duration ○ Time Elapsed ○ Is On (boolean for whether the timer is counting down)	○ Start Time ○ Stop Time ○ Get Start Time ○ Get End Time ○ Get Duration ○ Set Duration ■ New Duration ○ Get Time Elapsed ○ Get Time Remaining ○ Check if Timer has Ended

5. Power Up

- Power ups will inherit from this class
- Store the unique behavior that will manipulate many condition in the game

Data	Methods
○ Power Up Name ○ Power Up Requirement	○ Get Name ○ Set Name ■ New Name ○ Get Requirement ○ Set Requirement ■ New Requirement ○ Check if Power Up is available ■ Current Streak

6. Streak

- The object's responsibility is to keep track of the amount of times player succeed in answering question without failure.
- The object will also calculate the score

Data	Methods
○ Current Streak	○ Get Streak ○ Set Streak ■ New Streak ○ Increment Streak ○ Reset Streak ○ Decrement Streak

7. Leaderboard

- The object is used to access and process the collection of players and it's scores

Data	Methods
○ Sorted List of Players	○ Display Leaderboard ○ Sort Leaderboard

	○ Reset Leaderboard ○ Add Player ■ New Player ○ Remove Player ■ Player to Remove

8. Game Mode

- all game modes inherit from this class
- Game mode may implements activity or fragments class

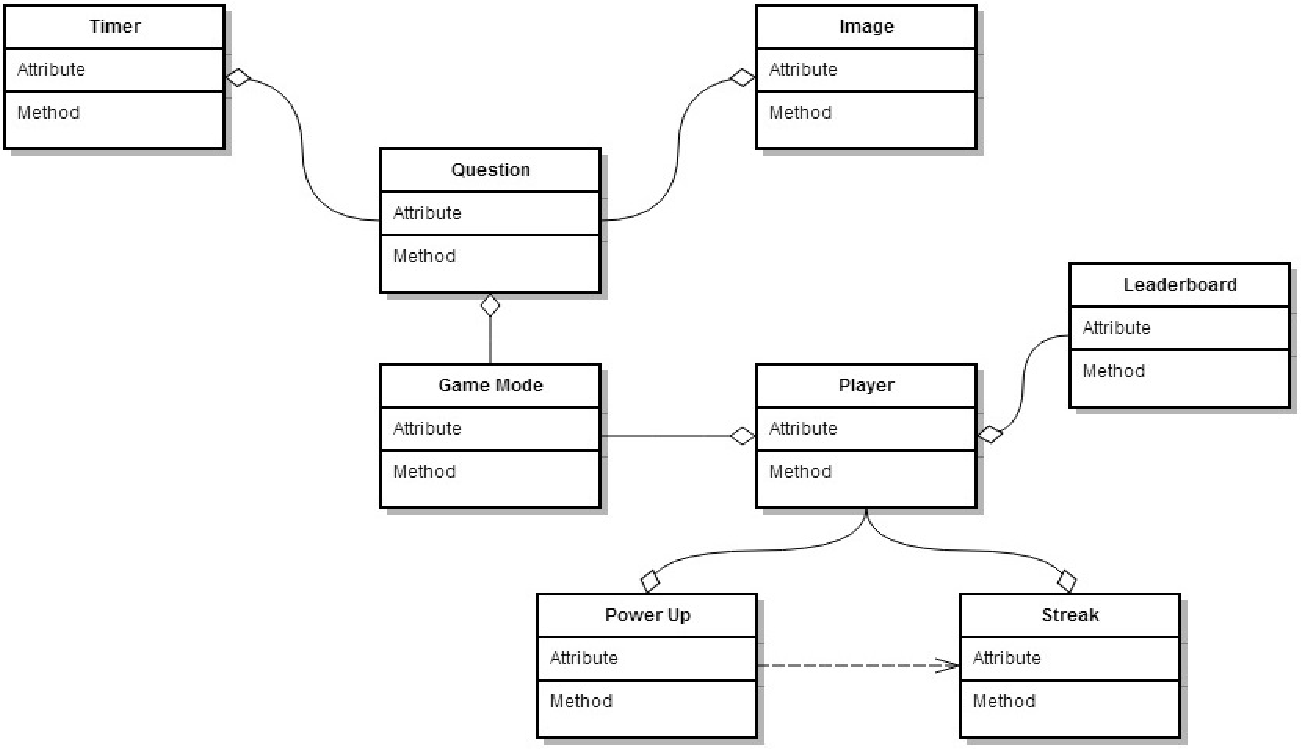
Data	Methods
○ Sorted List of Players ○ Game Mode Name ○ Online Status ○ Game Modifiers ■ Number of Questions ■ Question Category List ■ Question Difficulty List ■ Number of Players ■ Is Timed ■ Time Duration	○ StartGame ○ EndGame ○ ExitGame

6. Grouped List of Objects according to Inheritance and Aggregation

1. Player
    - has:
        - Player (List of Friends)
        - Streak
        - Power Up
    - is:
        - Account
2. Question
    - has:
        - Image
        - Letters
        - Timer

   - is:
3. Image
4. Leaderboard
   - has:
     - Player
5. Timer
6. Game Mode
   - has:
     - Question (List)
     - Player
   - is:
     - Online
     - Single
7. Streak
8. Power Up
   - is:
     - Reverse Image
     - Input Delay
     - Black Out
     - Fade
     - Auto-Error
     - Other Power Ups

7. UML Diagram



8. Table for Data Range and Exception

Data	Range
Player ID	0-9; Max 10 Char
Player Name	Letters (Upper & Lower), Numbers, Underscore, Space; Max 14 Char - Min 4 Char
Highest Score	Greater than 0 & Less than 999,999
Highest Streak	Greater than 0 & Less than 999,999
Wins	Greater than 0 & Less than 999,999
Losses	Greater than 0 & Less than 999,999

W/L Ratio	Greater than 0.0 & Less than 999,999.0
Current Score	Greater than 0 & Less than 999,999
Current Streak	Greater than 0 & Less than 999,999
Volume Preference	Greater than 0 & Less than 10
Brightness Preference	Greater than 0 & Less than 10
Online Status	Online, Offline, In-Game, Busy
Friends List	Number of Friends: Greater than 0 & Less than 99 Friend: Has to be a Player
Leaderboard Rank	Greater than 1 & Less than 999,999,999
Correct Letters	Letter Value: A-Z, 0-9
Incorrect Letters	Letter Value: A-Z, 0-9
Image URL	Image Type : jpeg Image URL:
Image Name	Letters: A-Z, 0-9 Max 10 Char, Min 1 Char
Image Difficulty	Easy, Medium, Hard
Image Category	Category Name: A-Z Max 10 Char
Letter Value	A-Z, 0-9
Time Duration	Greater than 0 & Less than 999,999,999
Time Elapsed	Greater than 0 & Less than 999,999,999
Time Remaining	Greater than 0 & Less than 999,999,999
Timer is On	True, False
Power Up Name	Letters(Upper & Lower)
Power Up Requirement	Greater than 0 & Less than 999
List of Players	Number of Players: Greater than 0 & Less than 999,999,999 Players: Must be a Player

Input Answer	Must be of Letters A-Z, 0-9
Categories List	Must be of Categories Number of Categories: Greater than 1 & Less than 99
Game Modes List	Must be of Game Modes Number of Game Modes: Greater than 1 & Less than 99

6. Model-View-Controller Design Patterns

Since Android applications are designed around MVC design patterns, all objects in the program will be classified into 3 types:

Model

1.1. Model objects are responsible in holding obvious data that user is concerned with. For
example, user, product, photo, questions and answer.

1.2. These objects is usually contained in *NameActivity.java*.

View

1.3. View objects involves things like screen and user interaction (Anything that can be visually seen on screen). Ex : Animations, buttons, menu bar.

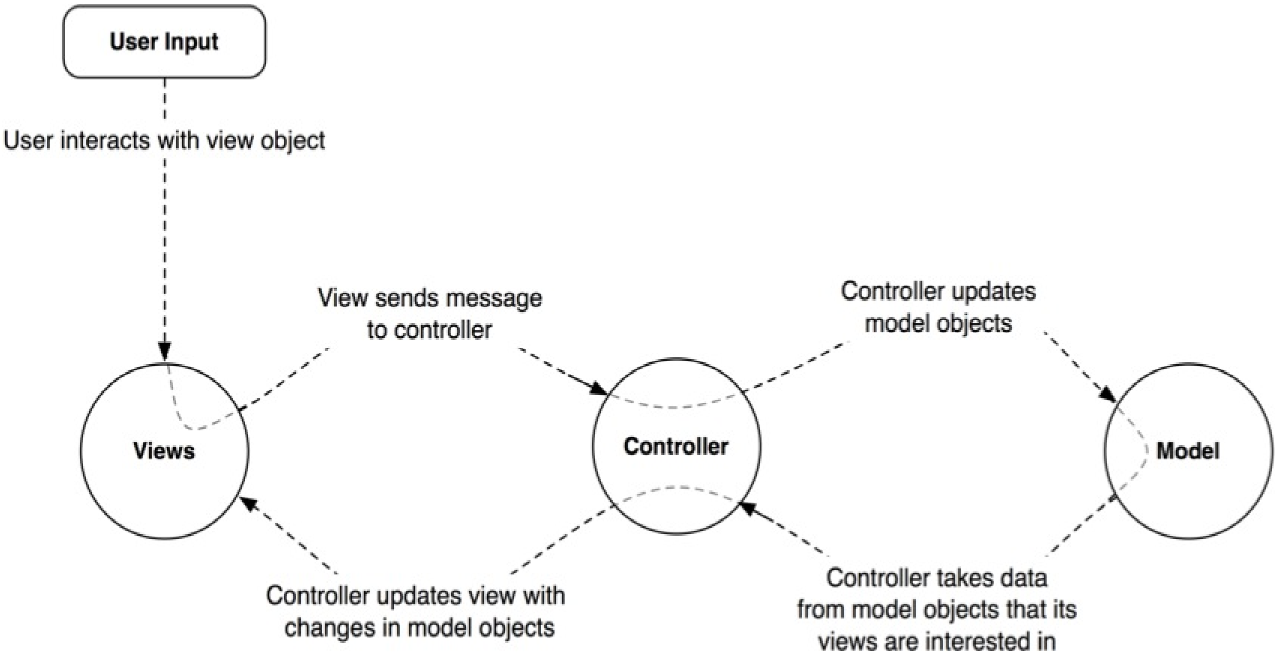
1.4. Android has many view class that is available for use

1.5. Usually handled in *activity_name.xml* (See : Inflated view objects).

Controller

1.6. Controller objects connects Model objects and View objects together.

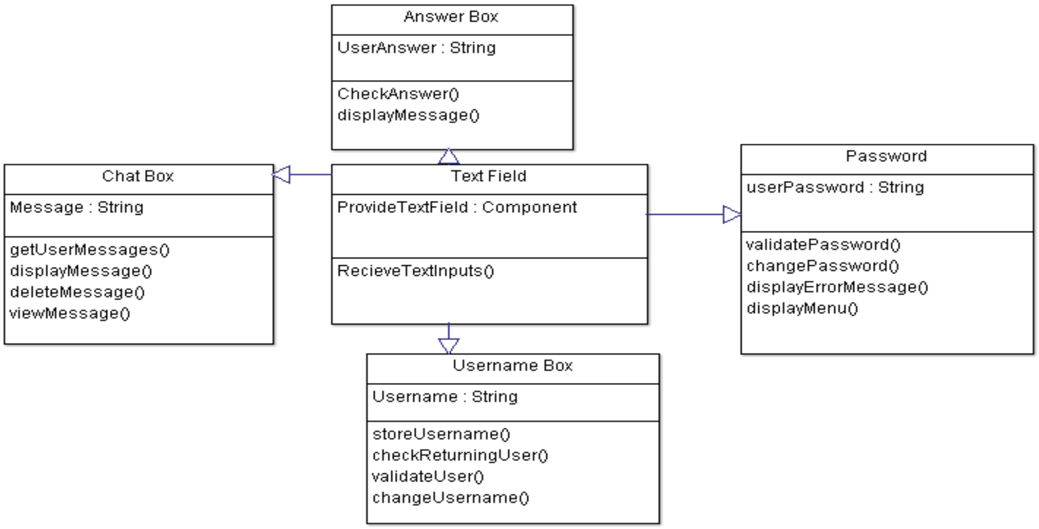
1.7. These includes *Listener,* Subclass *of Activity, etc.*


User Input
User interacts with view object
View sends message to controller
Controller updates model objects
Views
Controller
Model
Controller updates view with changes in model objects
Controller takes data from model objects that its views are interested in

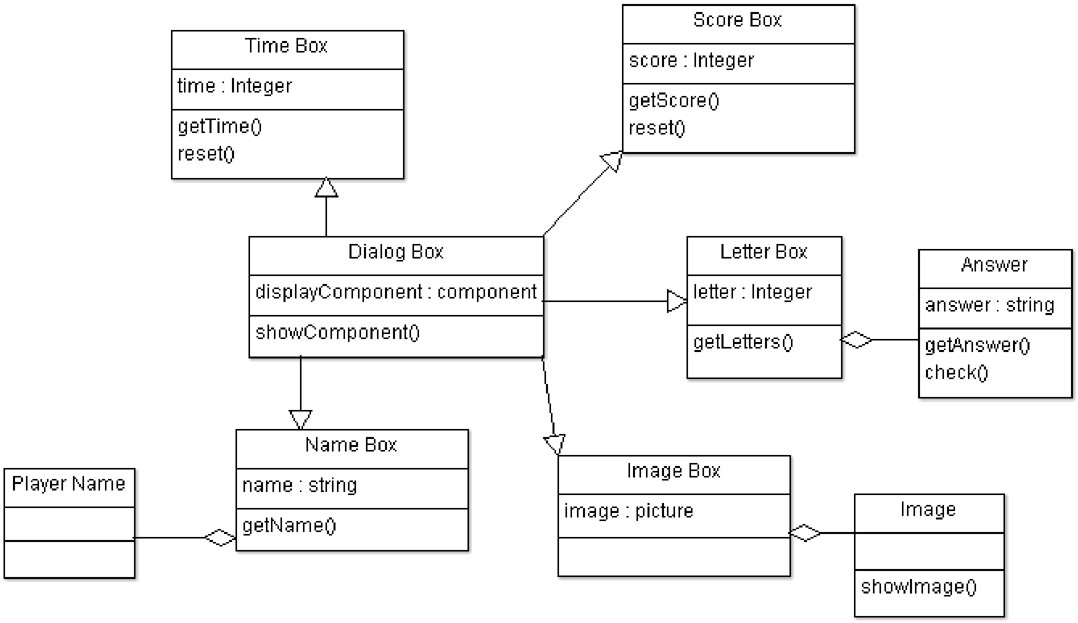
Source : Big Nerd Ranch Android Development

UML Diagram for Technical Design

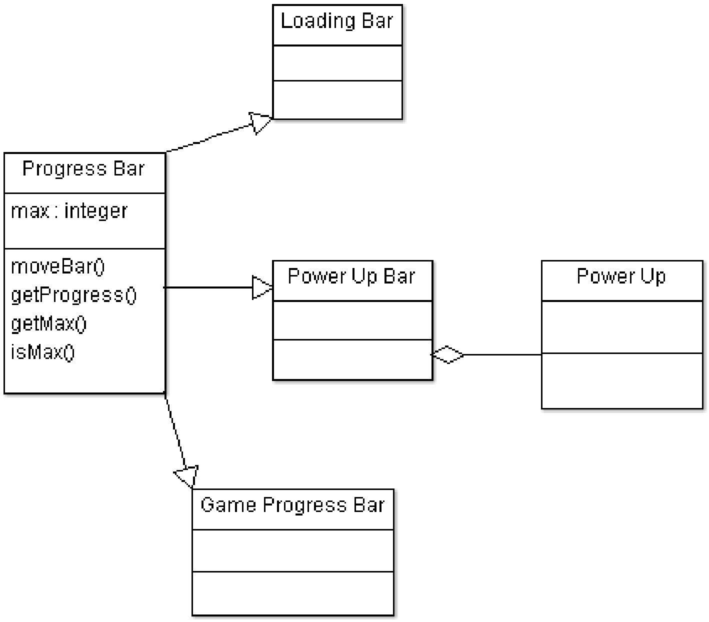
Text Field



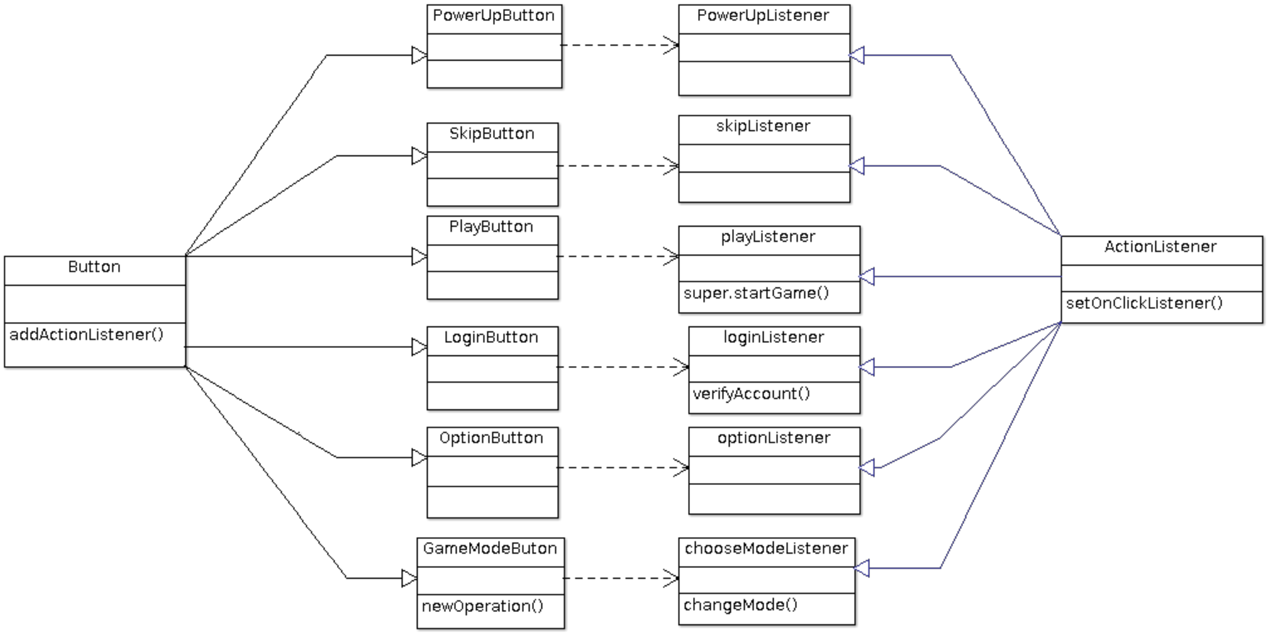
Dialog Box



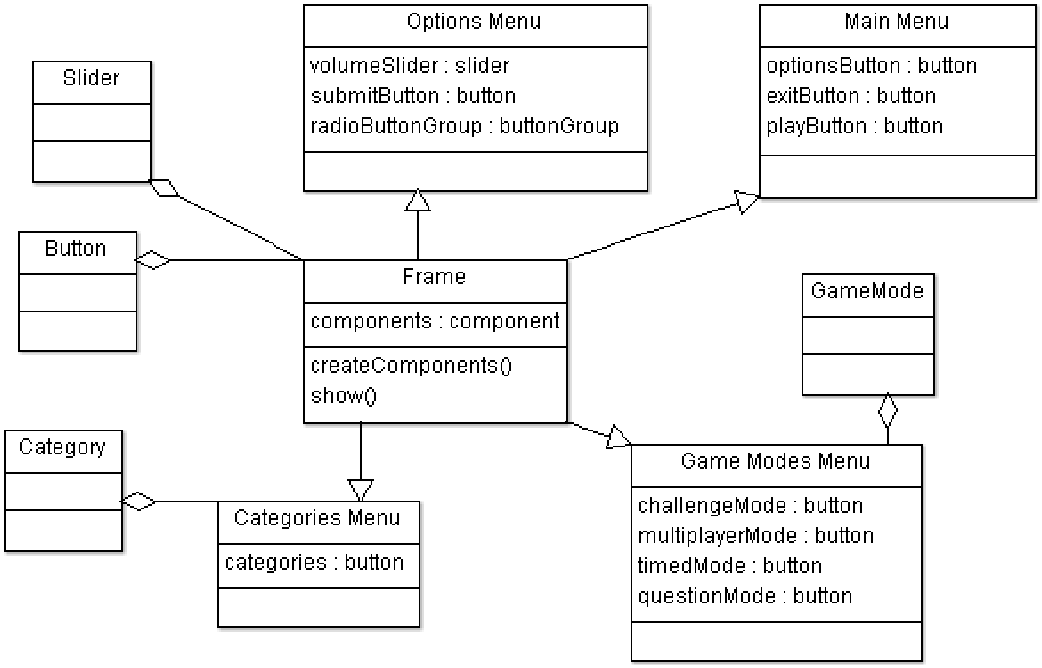
Progress Bar



Buttons



Menu Frame



7. Installation

The application is not published on google play store due to licensing policy. Hence, the application will be available for download with a public link which can be found on the project's github page (https://github.com/Josue3205/QuickCharades).

To allow the installation of third-party app on android device :

1. *Step 1 – Go to Settings.*
2. *Step 2 – Go to Applications.*
3. *Step 3 – Tap and check "Unknown sources" box.*
4. *Step 4 – Tap 'OK" when it show the warning.*

Then, to download the app, use the browser to go to the link : https://docs.google.com/uc?authuser=0&id=0B-BySq1dcRE_Ukw1bVROVIFLS2c&export=download

Allow the app to access resources from the phone to allow installation. Enjoy !

8. Project Analysis

1. Areas of Requirement Completed:
    a. **Player Name And High Score/Other Stats -** We incorporated player stats and names, storing them into a database. We pull the player’s info at the start of the app and after each game the player plays, we update the info onto the database.
    b. **Image/Question Categories -** We store the questions onto a database. When a game begins, we use the information the player puts in during the setup to pull a randomly selected questions that have the same category from the database. The number of questions is also set in the setup.
    c. **Skip Button During The Game -** The skip button allows the players to skip a question, if they do not know the answer.
    d. **Streaks -** The streak counter keeps track of how many questions the player has answered correctly in a row- if a player gets a question wrong, the counter will reset from zero.
    e. **Different Difficulties -** We have different difficulties that influence how much time a player has to answer each question. This is set during the setup.
    f. **Game Results -** We have a results screen that lists the setup choices the player has made, as well as, the score, streak, and time from the game. It allows the player to play again with the same exact settings from the setup, or go back to the main menu.
2. Areas of Requirement Missing:
    a. **Game Settings -** We do not have a fully functioning options menu. This will later include brightness, volume, and name changes.
    b. **Leaderboards -** We do not have a functioning leaderboard. This will in the future get every single player’s rank and sort accordingly.
    c. **Friends List -** The friends list will eventually be added. This will hold all of a player’s friends, allowing for quick messages or invites, as well as adding or removing friends.
    d. **Multiplayer Matchmaking -** Multiplayer matchmaking will be added in the future. This allows for players to go head-to-head against friends or against random opponents of a similar skill level.
    e. **Power Ups -** Power Ups for both single and multiplayer. These will add new twists to the game.
3. Future Enhancements:
    a. **Multiplayer Matchmaking -** Better matchmaking algorithm (than the one planned), that will match players based on category skill, not just overall skill.
    b. **Power Ups -** More interactive power ups.
    c. **Friends/Social -** Better social integration, allowing messaging between friends, inviting to games, etc.

   d. **Achievements** - Achievements for players to achieve.
   e. **Game Modes**- More game modes, including challenges, and maybe a customizable one.

4. Suggestions for Starting:
   a. **Learn Android Programming Properly** - Learning how to program android applications is extremely difficult. Finding a good book or a good video series is important in understanding on to create an application. Using Android Studio as opposed to Eclipse is also rather helpful as the application preview helps with interface creation.
   b. **Learn How To Create And Use Databases for Android** - Database are very helpful, and you are most likely going to need one to store most, if not all, of the info on players, images, etc.
   c. **Use GitHub Or Something Similar** - Version Control is very important in a project as large as this.