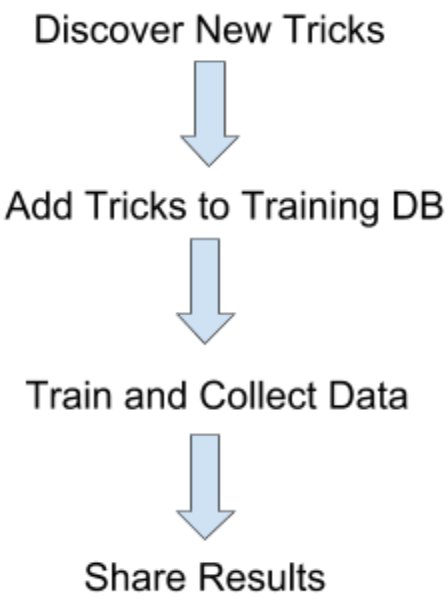
GitHub Username: smeschke

Big Top Tricks

Description

Train smarter, not harder. Big Top Tricks will help you maximize your learning rate and have more fun while juggling!



Big Top Tricks takes you through the complete spectrum of juggling training. Go from muggler to juggler in no time!

Begin with discovering new juggling tricks. Pick a trick from the big list of juggling tricks (ranked by difficulty). Read the description and watch video of the trick.

Add several tricks into your training DB. When you train juggling, practice the tricks in the training DB to stay focused and productive.

Use the tools in the training activity to progress faster. Record training data like number of catches, and hit/miss. Along with the timestamp and place data, this data is entered into a database.

After training, check out your progress! Share graphs of training data. The data can also be written into a .csv file for analysis in a spreadsheet.

Intended User

Big Top Tricks is intended for jugglers who train for 3+ hours a week. Many of the features will also be useful for training other repetition based activities.

User Interface Mocks

Main Activity

Feature: List of tricks from the training database is shown here.

The app launches to this activity. The list is shown in a RecyclerView.

The RecyclerView is populated with data that is stored in a database and accessed with a Cursorloader.

The pop-up menu has links to the following activities:

- Add Trick to DB
- Add Trick from List
- View Training DB

When a user is ready to train, a tap on one of the list items sends the user to the training activity.

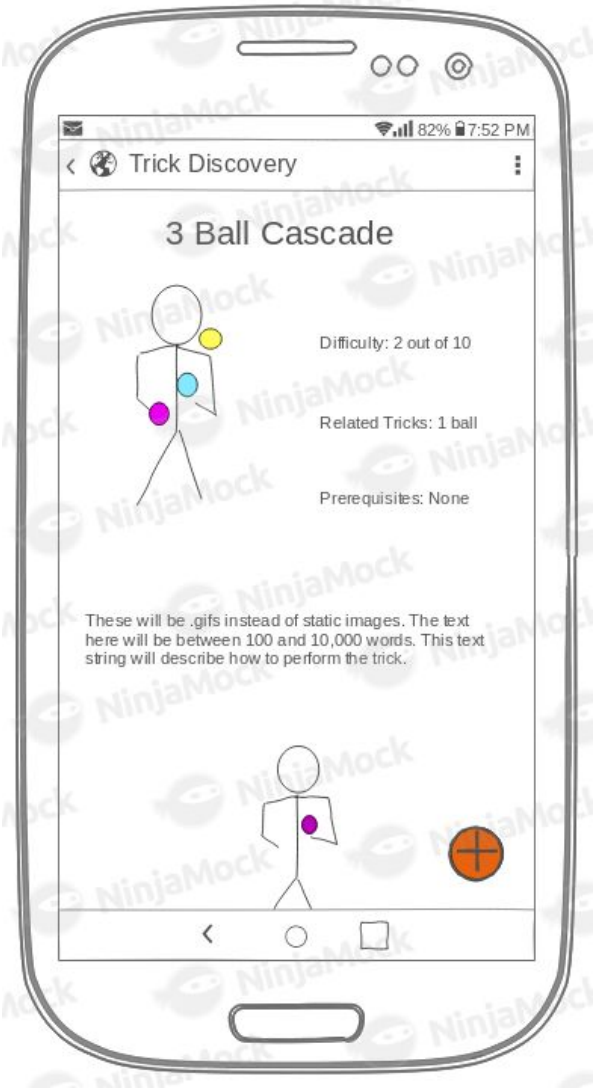
User Interface Mocks (continued)

Big List of Tricks Activity

Feature: Discover new tricks from this big list of tricks (list stored online).

This activity displays a big list of juggling tricks in a RecyclerView. The list is populated with values from a JSON file. The JSON file is stored online and accessed with an AsyncTask.

Users can click on a trick to see more details in the discovery activity (below).



Trick Discovery Activity

Feature: Text, video, and animations describe and teach tricks.

User Interface Mocks (continued)

Add Tricks Activity

Feature: Users can add a trick to the Training DB.

The user can populate these values in two ways:

1. Create a custom trick by filling out the values manually.
2. Select a trick from the list of tricks, where the values will be populated by data from the list.

Big Top Tricks

Training DB:

- 3 ball cascade
- Siteswap: 423
- 2 ball juggling in one hand
- 4 ball fountain
- Siteswap: 552
- Siteswap: 55550
- Siteswap: 55514
- Five ball cascade

Widget Activity

Feature: A list of the tricks in the training database is shown here.

User Interface Mocks (continued)

Training Activity

Features:

- **Graphs, details.**
- **Easy to record training data**

Users will spend the majority of the time in this activity.

When a user is ready to start training, the **start training** button is tapped. This starts the chronometer, and the **start training** button changes to a **stop training** button. When the user hits **stop training**, a dialogue appears that prompts the user to enter the number of catches achieved.

Users can record detailed information about the number of success and failures using the hit and miss buttons.

User Interface Mocks (continued)

Display Training Data Activity

Feature: Training data displayed. Can be exported to a .csv file.

This activity displays a list of all the training data from the database, displayed in a RecyclerView.

From the popup menu, user have the option to export the data to a .csv format.

Key Considerations

Data persistence:

The data is stored in an SQLite database. The data is accessed with a content provider. There is one table in the database that has the following columns:

- ID
- Trick Name
- Trick Description
- Goal
- Prop Type
- Personal Record
- Time Trained
- Hit/Miss
- Place
- Contains Metadata

Edge and corner cases in the UX:

- Users may select **start training** and then navigate away from the app before selecting **finish training**. The chronometer should pause when a user navigates away from the app, and restore the value if the user navigates back to the app within a few minutes.
- Users can enter text to describe the name, description, goal, and prop type. This must be enforced as text not null.
- Users should be able to enter multiple versions of the same trick, but each trick must have a unique name.
- After a user hits finished training, a dialogue appears asking for the number of catches. This must be enforced as an integer.
- App must not crash if user has no connection.

Libraries:

GraphView 4.2.2 will be used to display graphs of training data.
Exoplayer 2.8.0 will be used to play video.
Gradle 4.8 will be used to build the project.
Android Studio 3.1.4 will be used to create the project.
This project will be written solely in Java.

Implementation of Google Play Services:

Place data will be logged with every record. Place will be acquired with Google Play Services.

Analytics will be used to collect data on app users. Data on the amount of time that users spend in the training screen will be collected using the Google Play Services.

Required Tasks

Task 1: Project Setup

- Create New Project
    - Min SDK 15
    - Target SDK 27

Task 2: Create Database

- Package for the database classes
- Database Contract
- Database Helper to build the SQLite database
- Provider to serve the data
- Cursor Loader in the main activity

Task 3: Create RecyclerView Adapter to display Training DB

- TrainingManifestAdapter.java that extends RecyclerView
- Create training_manifest_list_item.xml.
    - Constraint layout
    - TextViews for the trick name and description.

Task 4: Implement UI for the Main Activity

- Query Training DB.
- Display a list of tricks in the MainActivity
- Add a menu with the options; ‘Add Trick’, ‘List of Tricks’, and ‘Training DB’
- Theme extends from AppCompat

Task 5: Implement UI for the Add Trick Activity

- Create a new activity called AddTrickActivity
- Add EditText views (for trick name, trick details, goals, and prop type)
- Add Button for ‘add trick’.
- Check for edge cases:
    - Text fields are null
    - Trick is already entered in DB
    - Create *Trick Not Valid* dialogue
- Add the trick to the DB
- Implement up navigation
- Theme extends from AppCompat

Task 6: Implement UI for Training Activity

- Create a new activity called Training
- Create the following views:
    - TextViews for: trick title, pr, goal, time spent training, and hit/miss ratio
    - GraphView for past performance
    - Buttons for start training and hit/miss
- Package the trick object in an intent from the MainActivity to the training activity.
- Get trick details from intent
- Bind the data to the views
- Create chronometer
- Wire start training button to chronometer
- Create a dialogue that asks the user for the number of catches
- Add the training data point to the database
- Wire the hit/miss buttons so that they update the TextView
- Wire the hit/miss buttons so that they update the database.
- Add a menu item that allows users to remove the current trick from the training DB
- Implement up navigation
- Theme extends from AppCompat

Task 7: Implement UI for Training DB Review Activity

- Create a new activity called DisplayData
- Populate the TextView with all the training data from the DB
- Add a menu item that allows user to export the data in a .csv file.
- Implement up navigation
- Theme extends from AppCompat

Task 8: Implement UI for Big List of Juggling Tricks Activity

- Create a custom RecyclerView adapter for the Juggling Tricks Activity
- Use an async task to get the JSON data from the internet
- Validate JSON data
- Bind the data to the adapter, and set up the click listener
- Implement up and back navigation
- Theme extends from AppCompat

Task 9: Implement UI for Trick Discovery Activity

- Create TextViews for the trick name and description
- Create a video view for the animation of the trick
- Bind the views with data from an intent
- Create a button that takes users to the add trick screen
- Implement up and back navigation
- Theme extends from AppCompat

Task 10: Implement Analytics from Google Play Services

- Places - CurrentPlaceDetailsOnMap
    - Instantiate the Places API client
    - Request location permission
    - Add a map
    - Get Coordinates
    - Get the current place
    - Save the map's state
    - Save current place in database
- Analytics from: *firebase.google.com/docs/analytics/android/start/*
    - Add app to Firebase project in the Firebase console
    - Add Analytics to app
    - Log event - user tap on ‘Start Training’
    - Confirm Event

Task 11: Build

- Move all strings to strings.xml
- Add content descriptions for any drawables
- Remove unused imports and implementations
- Clean
- Build and deploy using installRelease
- Keystore/password
    - Include in repo
    - Refer to by relative path