GitHub Username: grimesmea (https://github.com/grimesmea)

PrickleFit

Description

Pricklefit is a step counting that allows you to collect adorable hedgehogs as you reach fitness milestones. Hedgehogs you have collected can be used for your watchface but be sure to stay active as your hedgehogs will get grumpy if not active enough.

Intended User

PrickleFit is intended for anyone that wants to track how many steps they are taking in a day and who like adorable hedgehogs.

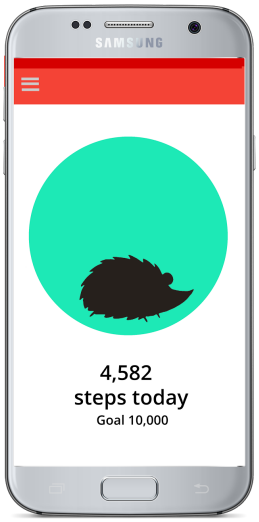
Features

- Uses Google Fit API to gather and store steps data
- Displays current day steps
- Past stats compare the current week's steps with those of the previous week (additional stats may be added)

- Watchface with percentage of step goal completed for the day displayed
- Collection of hedgehogs to unlock by meeting criteria (fitness goals and unlocked hedgehog happiness levels), unlocked hedgehogs can be selected to be shown on the app main activity and watchface

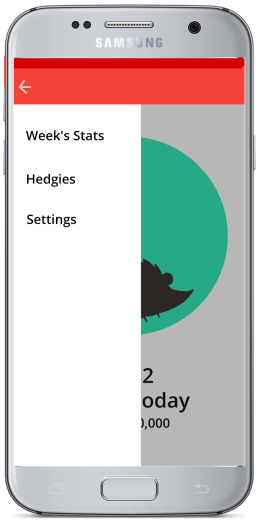
User Interface Mocks

Screen 1 - Main Activity



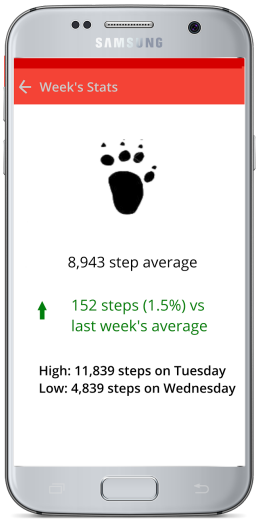
App's main view showing current day steps taken, and the currently selected hedgehog's happiness level (shown in hearts).

Screen 2 - Open Navigation Drawer



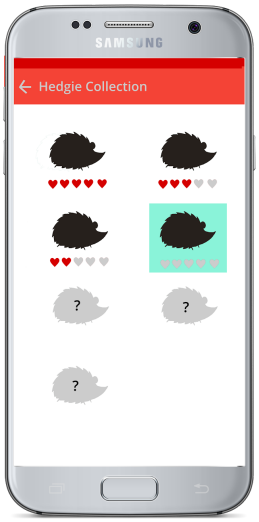
App's view with side navigation drawer open.

Screen 3 - Week Stats



App's weekly statistics. Displays basic stats about the current week's activity.

Screen 4 - Hedgie Collection



Shows the user's hedgehog collection, including unlock and lock hedgehogs.

Screen 5 - Individual Hedgehog

Detail view of each hedgehog. This fragment is also where the user can select the current hedgehog as the one to show in the main fragment and watchface.

Screen 6 - Watchface

PrickleFit watchface. Watchface will display the selected hedgehog. Red circle around main watchface represents the percentage of the user's step goal they have made in the day. Small red tick functions as a second hand and will give the illusion of the wheel spinning.

Key Considerations

How will your app handle data persistence?

PrickleFit will connect with Google Fit API fitness store to query and store data in addition to having its own content provider to store app "progress" in the form of user's unlocked hedgehogs and each hedgehog's happiness status.

Describe any corner cases in the UX.

NA

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

Google Play Services - Fitness: Will expose sensors allowing fitness data to be collected. Will also record and store fitness data.

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

- Configure libraries
- Get an OAuth 2.0 client ID

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Main Activity/Fragment
    - Includes implementation of functioning Navigation Drawer once necessary
- Build UI for PastStats Activity/Fragment
- Build UI for Hedgehog Collection Activity/Fragment
- Build UI for Hedgehog Details Fragment
- Build UI for Watchface
- Create Settings Activity with options below
    - Set a different daily step goal, default 10,000
    - Enable/disable notifications
- Add shared preference to store date app was first used (this date will be used to exclude data previously collected by Google Fit apps in calculations to determine if a hedgehog can be unlocked)

Task 3: Implement Landscape and Tablet UIs

- Main Activity, PastStats Activity, and Hedgehog Details Fragment will be reflow in landscape
- Hedgehog Collection Activity will have a master detail flow for sw-600
- Have changes to daily step goal be reflected on the main activity view

Task 4: Create Content Provider to Store App Progress

- Build content provider to store hedgehog and hedgehog progress data (unlocked hedgehog and their happiness level)
- Hedgehog Table:
    - Name (String)
    - Unlocked status (boolean)
    - Image name(String)
    - Silhouette image name (String)

    - Description (String)
    - Unlock criteria 1 (String)
    - Unlock criteria 2 (String)
    - Unlock criteria (String)
    - Happiness level (int 0-5)
    - Fitness Requirements (int 0-2)
- Setup up XML file with starting hedgehog data info that will be inserted into the hedgehog sqlite table when app is first started (determined by whether or not the hedgehogs table is empty)

Task 5: Implement Google Play Services - Fitness

- Add basic Fit Recording API calls to enable automated storage of sensor data and Sensor API calls to allow real-time display of data in the app
- Add basic Fit History API calls so app can store and access data in the fitness store

Task 6: Setup UI to Display Fitness Data

- Setup app UI to display the basic fitness data from the Sensors API
- Setup watch face UI to display the basic fitness data from the Sensors API
- Setup watchface goal progress meter surrounding main watch face to show percentage of step goal made in the the day

Task 8: Setup UI to Display Hedgehog Data

- Create Hedgehog class to allow for easy handling data
- Query Hedgehogs and display hedgehogs and hedgehog details based on returned cursor

Task 9: Add Logic to Check for Goal and Hedgehog Unlock Progress

- Add logic that will periodically check if the user is 50% or 100% to their goal, upon which a notification will be issued
- Add logic that will check for Hedgehog unlock criteria being met
- Add logic that at the end of the day will determine the hedgehog's happiness level