GitHub Username: JustinPriday

Equest-X

Description

Quickly learn your Eventing Cross Country course with run throughs of course notes and pictures of your obstacles. Then take your Android Wear device out on course for course feedback to nail that perfect round.

Intended User

Equestri-X is aimed at any Equestrian Eventing competitor who is taking part in their Cross Country round. The app will learn a course by accompanying the user when they walk their course. The user should also be able to carry their device with them when competing for course feedback. Users with an Android Wear device, though not mandatory, will benefit from the remote visual feedback that their device cannot afford.

Features

- Quickly learns a course when in course learning mode by tracking the user's route as they walk the course.
- Accepts course notes and uses the devices camera to capture obstacle images for review. This includes obstacle details like whether it is a flagged feature
- Accepts course specifications like the specified length and allowed time.
- Runs the user through the course with a course map overlayed over satellite images. Course notes and images are shown as the user progresses through reviewing the course, allowing them to review in the order they need to be considered.
- Competition mode is triggered before the user starts their timed course. The user will trigger the start of their course to get on course feedback on progress and time allowed.
- An application widget is available to quickly use a spare moment to look over your course, or jump to your course creation tool.
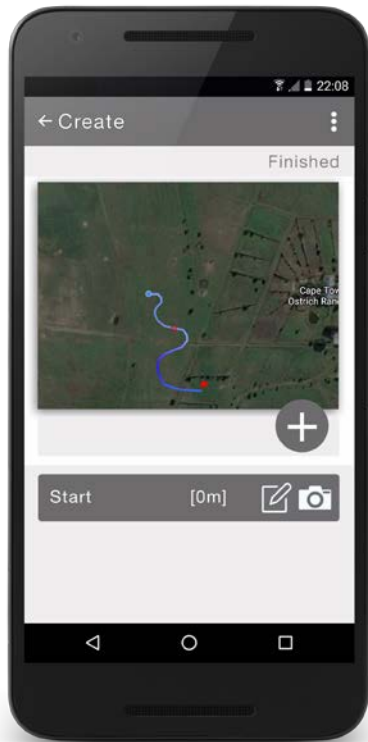
User Interface Mocks

Screen 1
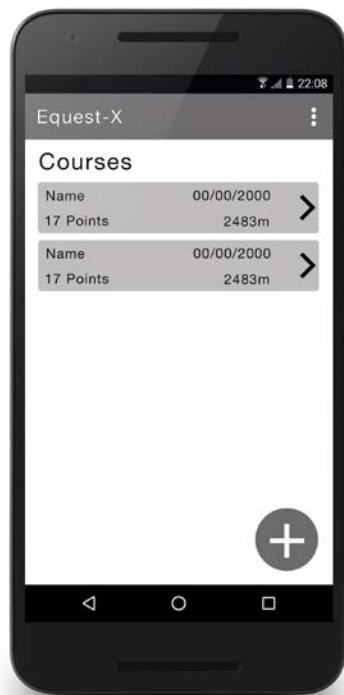
The list view provides the user ability to select a course that has been created on the device, or to create a new course using the floating action button. This list will also provide user the opportunity to manage their courses.
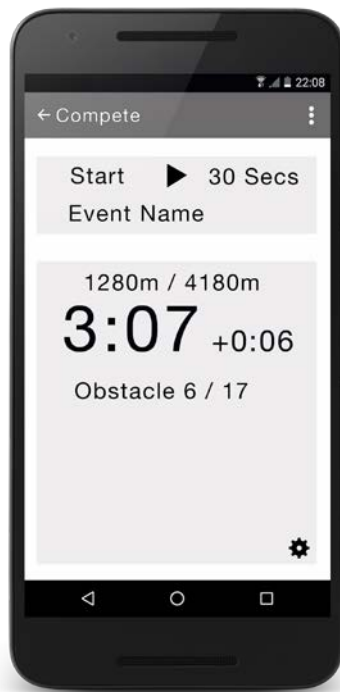
Screen 2

The creation screen allows the application to learn a course while the user walks the route. Typically in Cross Country the rider is afforded the opportunity to walk the course and inspect the route prior to the event taking place. When creating the course the user would flag the start of the route. The application would then record the user's location at key points so that it can learn the route the rider will follow. The user can then add obstacles or note points while walking, taking pictures of those points for later reference. The user would also set the direction in which the obstacle needs to be negotiated, as well as flagging the obstacle as a required obstacle for the course, or just a point that requires attention. Upon arriving at the finish line the user would flag the finish line and save the course. At this stage the user would be allowed to add further notes, like a description of the course.

Screen 3

After finishing the course learning process, the user would be able to review the route. This is entirely for the user to learn the sequence of obstacles on the route and review notes they had attached to each of those obstacles. The user would be able to step through the route, from key point to key point, reviewing both notes on individual points and viewing transitions from point to point on the map of the course. The user would be able to move forward or backward through the sequence of key points and would be able to view the images in detail by tapping the image thumbnail. The application should indicate the obstacle name, how far along the course it would be found, as well as whether this was a flagged required obstacle.

Screen 4

Although marked as screen four, a cross country rider would not typically be afforded the opportunity to see the screen while competing in the event. For this reason the rider would select a course from the course list and would start the activity before lining up at the start line. This would typically be using a timed start delay before the round starts, allowing the user to safely stow their device. Feedback to the user would be given using audio feedback.

A further option is an Android Wear companion app, allowing the user far more useful event feedback, also allowing them to trigger a more accurate start time. This may be outside the scope of this delivery.

Key Considerations

How will your app handle data persistence?

Data will be persisted using a SQLite local database. A Content Provider will be built to expose the data to the application. Loaders will be used to provide data lists and data for individual courses as required by the activities.

Describe any corner cases in the UX.

The application provides two very time and location sensitive activities in the way of recording a new course and running that course. When creating the course the application will save the partially created course in the instance variable when the application is paused. When resuming the application will attempt to resume course creation, this will be dependent on the user still being in the vicinity of the course being created, likely dependent on being within a threshold distance from the most recent recorded point.

When competing on the course this activity is both time and location dependent so resuming a competition course will depend on both being in the correct location and within the expected time frame. This activity however should try to recover from the break by trying to ascertain where on the course the rider is based on their location and the elapsed time. The application should try to best resume from this point in the course.

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

Picasso – For image management and easy handling of unavailable images. Loading images would also be useful though images are stored locally so this image is unlikely to be experienced by the user.

Butterknife – For easy field and method binding with more readable code.

Describe how you will implement Google Play Services.

Location Play Service will be used to find the users location for both the course learning and the course playback when competing.

Maps Play Service will be used to integrate mapping with the learning of courses by overlaying the course, as it is created, on a map of the user's immediate vicinity.

Next Steps: Required Tasks

Task 1: Project Setup

- Create new Android studio project for Equest-X app.
- Configure and include libraries used in the app.
- Configure application permissions required for the app, most notably the location tracking permission.
- Create a GitHub repository for the project.

Task 2: Implement UI for Each Activity and Fragment

Implement User Interface for the application, using Material Design guidelines with special attention to Material Design specifications in the Project 8 rubric. Specific activities and fragments are as follows:

- Build user interface for Courses List activity using a recyclerview to show the list of available courses.
- Build user interface for course learning screen.
- Build user interface for Course Review screen.
- Build user interface for Compete screen.
- Build tablet specific User Interfaces.

This application doesn't particularly lend itself to the tablet form factor due to the applications reliance on a small device that's stowable on the user during competition. The tablet user interfaces will be included to meet requirements of project 8. Tablet interface will become useful when portable devices can share their courses with tablet devices, for course review on a larger screen. This sharing functionality will be outside of the scope of this deliverable.

Task 3: Build Data Models and Data Persistence.

Implement all data models to support a course and its key points. Build system to save camera images and associate these images to the data models. Build Content Provider, Cursor Adaptors to make this data available to the application.

Task 4: Implement Location Tracking

Implement Location Tracking for the course learning activity. Implement system to follow this location to build a course with user interface to mark start and end points for the course. Implement a system to tag points of interest with systems to capture relevant data for these points of interest including:

- A photograph of the obstacle.
- Obstacle title.
- Obstacle description.
- Is the obstacle flagged as a required obstacle for the course?

Task 5: Implement Course Review

Implement a system to allow the user to review the course between learning and competing. This system should allow the user to step forward and backwards through a course as well as quickly jump to a particular point on the course. The user should step through points of interest with interface providing all relevant information for key points.

Task 6: Implement Competition Activity

Implement a system to guide the rider through their chosen course. The system needs to assume that the user both knows their course and that they are not necessarily be able to directly view their device screen. The device needs to provide audible queues to show the user their progression relative to their ideal position based on the courses optimal time, as well as warning users of specifically noted points of interest. The user interface does need to provide the user a system to pause and resume their course, should they be stopped by the course judge.

Task 7: Implement Application Widget

Implement an application widget to display an overview of the users most recently created course. The widget should show some vital statistics of the course and provide the user a control to quickly jump in to the course review activity of the application. The user should also be afforded control to quickly jump into the course creation activity.

Task 8: Implement Wearable Companion

Implement a system to give the user visual and access to course controls via an Android Wearable device, to be used during the competition activity.