

## Möte 21/9

Närvarande: alla

### 1. Vad har vi gjort sedan sist

Talar om veckan som gått. Inte så mycket eftersom det var första veckan och lite lugnt.

Patrik har lagt in ett UML-diagram av projektet och experimenterat lite med androirdramverket.

Henrik har undersökt olika tester och funnit ett testpaket baserat på JUnit.

Gustaf tittat lite på databashantering. Övervägde att spara en sträng men istället spara smidigt i databas genom indexerade kolumner. En egen tabell med alla checkpoints och en för slingor. Tabell med kolumner: Routes - id, name, description, type, timeCoach, lengthCoach. Inställningar ska sparas i klassen shared preferences. Tabell två: Checkpoints - id, routeld, sounds, name, description. Tabell 3: results - id, routeld, time, speed, decrion calories etc. Tabell 4 - att checkPoints sparar geoPointId.

Leder även till diskussion huruvida CP och GP ska spara olika värden. Vi komer fram till att de bör vara separata för att uppnå god kohesion. Tankar om hur det ska sparas:  
gp - id, route, long, lat (alla ints). Checkpoints har egna long- och lat-fält  
cp - id(int), sounds(text), name(text), description(text), gpId(int)

Markus - Skapade ett första utkast till GUI-tänket. Detta för att få en övergripande tanke om design.

Daniel - Har undersökt och använt Overlay-klassen, överridar draw-metoden. I denna lista finns punkter som är intressant. Med hjälp av denna kan vi rita streck på kartan. Skapat grundläggande funktionalitet som behövs i programmet. Skapat dagordningen till detta möte.

### 2. Dokumenthantering

Diskuterar om hur vi lättast håller koll på alla dokument vi genererar under kursens gång. Spara dokument i Dropbox vs Google Docs. Vi bestämmer att vi ska använda oss av Dropbox.

### 3. Nyckelhantering

Planering hur man ska använda keystores för att skapa en .apk. Vi bestämmer oss för att alla ska göra det en gång var. Lite också för att testa hur keystore-funktionen fungerar. Patrik bestämmer sig för att göra det första gången.

### 4. Dokumentering av möten etc

Inlämningstankar, någon slags readme inför varje inlämning för vad vi skickar och möjliga förändringar för att underlätta hittandet av nya dokument. Hinner vi inte gör Daniel det.

### 5. Git

Vad bör pushas? Experiment och dylikt bör ej presenteras. Alla kan dock skapa egna repositories för fri lek och för att undersöka systemet för att testa nya android-funktioner. Branches: en branch per feature/user stories, dessa benämns som issues. Dessa ska deklarerars med issue-id

## 6. Planering av sprint 1

Ifrån vår backlog:

### A0. Main Activity

value: 100    cost: 10

\*A1

\*A2

\*Musik (om tid finns)

Röstade värden - 1, 3, 3, 40, 100

Ansvar: Daniel, 1h + ?

### A1. Spela in en slinga och spara i DB

value: 100    cost: 150 (nytt värde: 75)

Väldigt viktig. Grundläggande för programmet.

\*Spara

Röstade värden - 1, 3, 5, 5, 8

Ansvar: Markus, 5h

### A2. Starta en inspelad slinga

value: 100    cost: 75

\*Hämta

Röstade värden - 2, 2, 5, 8, 40

Ansvar: Markus, 8h

### Databas

\*spara route

\*hämta,

\*spara/hämta cp

Röstade värden - 3, 8, 20, 20, 40

Ansvar: Gustaf, Henrik, 20h

### -Dokumentation till handledare

value: 200    cost: 30

Ansvar: Daniel, 1h

**-Lägga in CP under rundan**

**value: 95      cost: 40**

\*Database

\*Create

\*CP

\*CP Overlay

Röstade värden: 2, 2, 3, 20, 40

**Ansvar: Patrik, 4 h**

## **7. Designprat**

Lyfta ut lyssnarna. Göra mer designerade model-klasser för att få mer MVC-liknande uppbyggnad.

## **8. Route**

Route-klass är bra att ha.

## **9. Fikaansvarig**

Det blir arraxsbollar.

**????**

Vi ska ha två overlays på kartan, en för att rita slingan och en för att rita checkpoints. Båda använder RouteOverlay

Databas

-----

SaveRoute(List)

SaveCP(List)

GetCP(Route)