Utvecklarmanual

Hämta projektet

1. Hämta git-arkivet från https://github.com/gitsjogren/segr4.git
2. Se till att du har Android SDK, en Android (Virtual) Device samt Java 6 SE installerat som utvecklingsmiljö.

Importera projektet

1. Öppna Eclipse.
2. Importera projektet.

Om Eclipse klagar över compiler version, gör följande:

1. Högerklicka på projektet under Package Explorer.
2. Välj Properties längst ner i drop-down listan.
3. Gå till Java Compiler fliken i fönstret som öppnas.
4. Ändra Compiler compliance level till 1.6.

Hur man kör testen

Pga få testfall som kan möjliggöras och testas i vår applikation och brist på tid så finns dessvärre inga test tillgängliga.

Arkitekturspecifikation

När applikationen startar så startas först aktiviteten WelcomeScreen som visar vår logotyp ett bestämt antal millisekunder innan aktiviteten TheMap startas.

Innan kartan visas anropas checkIfGpsIsEnabled() som frågar om man vill aktivera sin GPS. Vidare animeras man till det campus som man valt som standard, detta väljs första gången applikationen startas då en dialogruta visas upp. Detta lagras sedan som cache på sin egen telefon.

TheMap använder sig av com.google.android.maps biblioteket och ärver från MapActivity, detta krävs för att kunna visa upp själva kartan genom mapView. I layouten showthemap ärvs mapView för TheMap genom klassen OnDoubleTap. Anledningen är för att kunna implementera doubleTap. TheMap är den huvudsakliga klassen i projektet och har hand om allt som sker på kartan, alla nuvarande metoder samt menyer.

TheMap använder sig av com.google.android.maps.MyLocationOverlay för att visa upp en punkt på kartan som visar en användares nuvarande GPS-position. Pubbarna visas upp på kartan i form av små ikoner genom att använda mapOverlay från com.google.android.maps.Overlay. En for-loop går igenom OverlayItem[], en lista med pubbarnas geopoints, och lägger till dessa till List<Overlay> för de Geopoints som ska få mapOverlay, alltså på vilken position en ikon ska sättas ut.

En geopoint är ett object som innehåller koordinater som com.google.android.maps.MapController använder för att genom animateTo(geopoint_object) förflytta sig till en specifik position med angivna koordinater. Denna metod används vid byte av campus.

Knapparna i menyn högst upp på kartan anropar varje en metod, detta sker genom en OnClickListener() på varje knapp som anropar metoden onClick() som har hand om det som ska ske när man trycker på knappen. Vid tryck på “Min position” anropas metoden showTheCurrentPosition(), “Byt campus” kontrollerar med en if-sats en boolean för vilket campus som man ska förflyttas till, changeToCampusLindholmen() respektive changeToJohanneberg. Knappen “Publista” startar klassen PubList där en lista visas. Vid tryck på en pub på kartan eller i publistan startas den gemensamma klassen PubLayout som har hand om all layout gällande samtliga pubbar. PubLayout visar upp specifik information för den tryckta pubben från xml-filer.

Längst ner på kartan visas en OnOptionsMenu upp vid tryck på telefonens inbyggda menyknapp. Vid tryck på en knapp går onOptionsItemSelected(MenuItem Item) in i en switch-sats där det “Item” som trycktes på körs. I fallet “Ändra vy” kontrolleras nuvarande mapView och sätts till den vyn användaren valt. Välja satellite mode görs tex. med mapView.setSatellite(true). “Inställningar” startar aktiviteten SettingsMenu där default campus kan ändras (dialogruta visas upp), skicka feedback (ny aktivitet FeedbackForm startas) och about (startar AboutPage) där man får information om applikationen. “Tipsa” anropar ACTION_SEND från Android systemet där man kan välja hur man vill tipsa om applikationen.

För mer detaljerad information kring klasserna samt deras metoder hänvisas man till källkoden, klassernas/metodernas namn med tillhörande kommentarer borde täcka det mesta.

Paketets struktur

Projektet använder ingen MVC modell utan TheMap är den huvudsakliga klasser som hanterar allt som sker i applikationen, visning av karta med överliggande meny med knappar som startar nya aktiviteter utanför kartan.

Applikationen följer den allmäna kodstandarden och eftersträvar hög koppling och låg kohesion gällande design mellan klasser och metoder. Som ett komplement till Javas API:er används Google egna API:er då applikationen är ett Android projekt.