

Oefeningen

Werkomgeving

1. Download en installeer Android Studio
2. **Voorbeeld:**
Maak een nieuw Android Studio project met de naam **Voorbeeld**. Kies voor API level 17 of hoger voor een phone en tablet. Kies daarna voor een Blank Activity. De naam van de hoofdactiviteit mag je laten zoals voorgesteld.
3. Open de AVD manager en maak 2 virtuele devices aan. Het eerste device voor Android versie 4.2.2 en het tweede device voor Android versie 4.4.2. Kies een 7 inch schermweergave. Start elk device op en test het.
4. Open de SDK manager en zorg ervoor dat de devices uit oefening 4 de Intel X86 Emulator Accelerator kunnen gebruiken.

App onderdelen

1. Open het project venster van het project **Voorbeeld** (Werkomgeving/oefening 2). Bestudeer de verschillende weergaves (Android, Project, ...).
2. Voeg aan het project **Voorbeeld** een menu-item toe met als titel "Afsluiten". Bij het uitvoeren van dit menu-item moet de functie finish() aangeroepen worden.
3. Voeg aan het project **Voorbeeld** een TextView toe en voorzie een string in het strings.xml bestand die gebruikt wordt in de TextView.
4. Voeg in het Manifest bestand van het project **Voorbeeld** een toelating toe om het internet te kunnen gebruiken.

Java code onderdelen

1. **Gezegde:**
Maak een nieuw project met de naam **Gezegde**. Declareer een string in de onCreate() functie en vul die op met een zelf gekozen gezegde. Voorzie daarna een For-lus om alle letters van het gezegde van achter naar voor af te lopen en in een nieuwe string te stoppen. Toon de oude en nieuwe string op het scherm in een TextView.
2. **Array:**
Maak een nieuw project met de naam **Array**. Declareer een array met de naam waarden en vul die array op met voornamen. Gebruik dan een For-each-lus om elk item uit de array tijdelijk op het scherm te tonen met een Toast object.

Activiteiten

1. Dashboard:

Maak een nieuw project met de naam **Dashboard**. Voorzie in de hoofdactiviteit 3 knoppen: act1, act2 en act3. Als op een knop wordt gedrukt moet er een nieuwe activiteit opgestart worden. Maak in het project 3 extra activiteiten aan: act1, act2 en act3. Op het scherm van elk van die activiteiten moet een TextView aanwezig zijn waarin de naam van de activiteit getoond wordt en een knop "terug" om terug te keren naar de hoofdactiviteit.

2. Wijzig de layout van de act3 activiteit door in de source code Java code te schrijven om een LinearLayout te definiëren. Schrijf ook code om een TextView en een Button te definiëren. De TextView en de Button moeten toegevoegd worden aan de LinearLayout.

Fragmenten

1. Talen:

Maak een nieuw project met de naam **Talen**. Kies bij de aanmaak voor een Navigation Drawer Activity. Voorzie in de lijst van de schuifbalk 3 items: categorie, woorden en zinnen. Maak een fragment aan voor elk van de items: CategorieFragment, WoordenFragment en ZinnenFragment.

Het scherm van het CategorieFragment bevat een invulveld naam. Het scherm van het WoordenFragment bevat een invulveld woord en een invulveld categorie. Het scherm van het ZinnenFragment bevat een invulveld zin en een invulveld categorie.

2. Voeg aan het project **Talen** voor elk fragment 3 menu knoppen toe. Een knop om toe te voegen, een knop om te wijzigen en een knop om te verwijderen.

UI componenten

1. Voeg aan het project **Talen** in elk fragment een ListView toe die opgevuld wordt met de gegevens die in het invulformulier zijn ingegeven. De ListView moet vernieuwd worden telkens als op de toevoegen knop wordt gedrukt.
2. Voeg aan het project **Talen** een nieuw fragment toe met de naam ZoekenFragment. Het scherm van dit fragment is opgebouwd uit een RadioGroup met 2 mogelijkheden: Woorden of Zinnen. Daaronder komt een Spinner van de Categorie en daaronder een ListView waarin de woorden of zinnen van de gekozen categorie kunnen getoond worden. Zorg voor de nodige arrays om in de Spinner en ListView te kunnen tonen.
3. Toon in het project **Talen** een AlertDialog bij het verwijderen van een gegeven. Als op de knop verwijderen in de AlertDialog wordt gedrukt toon je het gegeven dat zal verwijderd worden met een Toast object.

4. **Speelkaarten:**

Maak een nieuw project met de naam **Speelkaarten**. In de hoofdactiviteit voorzie je een GridView. In de GridView worden speelkaarten in willekeurige volgorde getoond. Als op een speelkaart wordt gedrukt dan wordt die kaart uitvergroot getoond in een andere activiteit in een ImageView. Zoek op het internet naar gepaste afbeeldingen van speelkaarten.

Databases

1. Voeg aan het project **Talen** een database helper klasse toe. In deze klasse voorzie je de nodige code om de gegevens van het CategorieFragment, het WoordenFragment en het ZinnenFragment te bewaren. Zorg voor een database tabel voor de categorieën, woorden en zinnen. Schrijf ook de nodige functies om de gegevens te kunnen toevoegen, wijzigen, verwijderen of selecteren.
2. Gebruik voor de Spinner en de ListView van het ZoekenFragment van het project **Talen** de nodige database functies om de adapters op te vullen.

Graphics

1. **FingerPaint:**

Maak een nieuw project met de naam **FingerPaint**. Maak van de hoofdactiviteit een klasse FingerPaint en kopieer de source code van de cursus in de activiteit. Maak een ColorPickerDialog klasse en kopieer de source code van de cursus. Voeg aan de menu van de FingerPaint klasse een menu-item "Save" toe. Schrijf de nodige code om de tekening te kunnen opslaan.

2. Voeg aan het project **FingerPaint** een menu-item toe om de dikte van de lijnen aan te kunnen aanpassen van flinterdun naar extra-dik en omgekeerd.