COMP 3717 DTC Midterm 202130:   **90 points**

Hippo Zoo

Layout criteria: **30 points**

- You must use a Constraint, Linear and Frame layout at least once
- Inside *activity_main.xml*
    - Views
        - TextView
        - Spinner
        - FragmentContainerView
        - Button
        - Button
- Create two Fragments (*PreviewFragment* and *ContactFragment)*
- Inside *fragment_preview.xml*
    - Views
        - TextView
        - ImageView
- Inside *fragment_contact.xm*l
    - Views
        - 5 TextView's
        - EditText/PlainText
        - Button
- Create another Activity (*InventoryActivity*)
- Inside *activity_inventory.xm*l
    - Views
        - RecyclerView
- Create an item layout resource file (*item_inventory*) for your RecyclerView
    - Views
        - 6 TextView's
        - ImageView
- All layout files may contain more then one View layout (Constraint, Linear, Frame), it is up to you to determine where they should go

Functionality criteria: **30 points**

- Create a model class (*Hippo.java)*
  - Variables
    - name (*String*)
    - age (*int*)
    - food (*String*)
    - resId (*int*)
  - Create any needed constructors and getters/setters
- When an item in your Spinner is selected it should update the FragmentContainerView with an appropriate *PreviewFragment*
- When the 'Full Inventory' Button is clicked, it should start the *InventoryActivity*
  - You must send data through an intent using an Extra and/or Bundle which is then used for your RecyclerView Adapter
- When the 'Contact' Button is clicked, it should update the FragmentContainerView with the *ContactFragment*
  - Entering anything in the EditText and clicking the 'Submit' Button should update the TextView above it to: *"Submitted"*
  - Entering *"Craig"* in the EditText and clicking the 'Submit' Button should update the TextView above it to: *"Submitted :D"*

Organization criteria: **10 points**

- Your code should be clean, tidy, and elegant
- There should be no files, classes or resources that aren't being used in your project
- Satisfy any 'Hardcoded Text' warnings by using a string resource from your strings.xml
- Use meaningful names for your strings and id's and avoid names like Button1, TextView1, etc. Use the 'Naming Convention Cheat Sheet' as reference

How closely does your app match the example: **15 points**

- Styles/Colors/View positionings/Strings
- For max points I will be picky on this

Submission Rules: **5 points**

- Submit zip file of your project (yourname_midterm)

App does not run: **-10 points**