

PROGRAMMING ASSIGNMENT #2

CIS 436 – MOBILE APPLICATION DEVELOPMENT – WINTER 2019
UNIVERSITY OF MICHIGAN - DEARBORN
PROF. JOHN P. BAUGH

Due: March 15, 2019 at 11:59 p.m.

Points: _____ / 100

OBJECTIVES

- To create a more complex user interface
- To use fragments
- To persist simple data

INSTRUCTIONS

You are charged with the responsibility of creating a simple “Game Character Creation” interface, giving the user the ability to set the default values of different character types. It is assumed you will use good software engineering practice throughout, including the following:

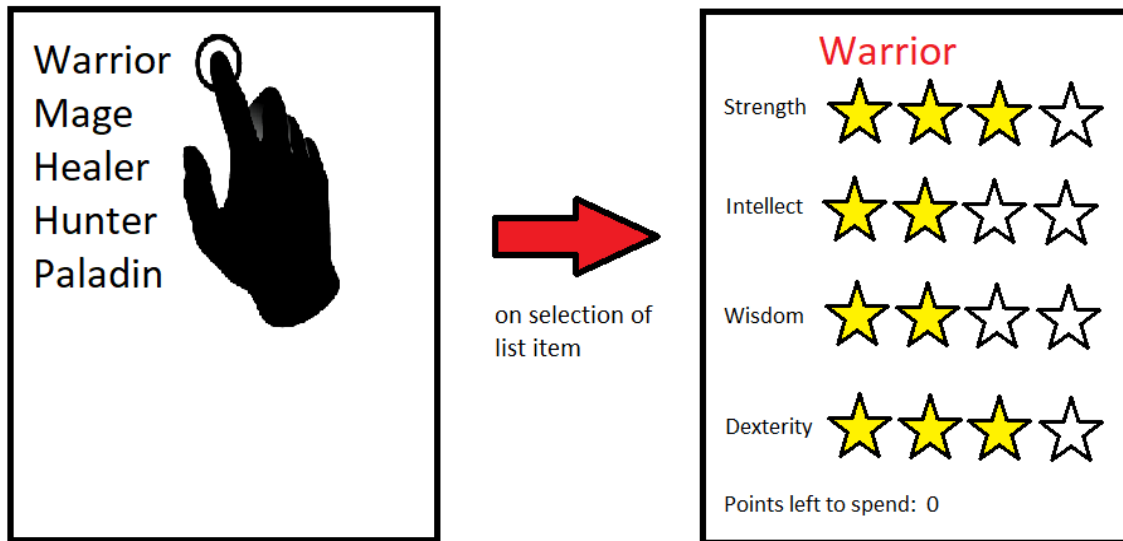
- Use comments (useful comments), including your name, instructor’s name, class you are in, and which project this is at the top of your Main Activity Java file
- Give appropriate ids to your widgets
 - Button1, button2, etc. will result in loss of points

For **each** character class is listed in a list. Note that this doesn’t mean “class” in the OOP sense – a “class” in traditional role playing games, and other games, means the profession or type of character (which then denotes that character’s skills in different areas, etc.)

Each character has a total of **10 points** to spend on 4 different attributes:

- Strength
- Intellect
- Wisdom
- Dexterity

ON A SMALLER DEVICE



Note that separate character professions/classes will result in a new activity being launched with the information displayed.

ON A LARGER DEVICE



Because larger devices have more screen real estate, the activity remains the same, but instead you use fragments to load/change only the right side of the screen to display the appropriate character information.

GENERAL REQUIREMENTS

You should ensure your program and assignment adhere to the following (In fact you could use the following, keeping the above instructions in mind also, as a check list):

- ☐ You must use fragments
- ☐ There must be a total of 10 points to spend for each character class
 - The user can navigate away without spending any, or spending all
 - The user cannot ever spend more than 10 points in total
- ☐ You must use the **RatingBar** widget for the star icons to denote the number of points spent on each attribute
- ☐ The app must display the “points left to spend” at the bottom (or the top – you can make small modifications to the interface.)
- ☐ The app must save the points spent on each attribute using **SharedPreferences**
- ☐ When the user navigates to and from different character classes, the spent points and related data must save and load the appropriate values, respectively
- ☐ You must create a basic test plan for **each** of the above features and fill it in
 - Use the example text plan on Canvas
 - Use Microsoft Excel (preferred), or Apache OpenOffice Calc, or LibreOffice Calc for the workbook with the Test Plan
 - If you use OpenOffice Calc or LibreOffice Calc, please generate a PDF and turn that in – do not turn in the native formats of these open source programs

DELIVERABLES

Zip your entire Android project, with the **test plan stored at the top level of the project**

Upload the entire zip file to Canvas on or before the due date to the appropriate project directory.