CPSC x150 Mobile Device Software Development **FA19**
Homework 2 – 35 points

You are highly recommended to start early on this project since you need to set up and become familiar with a new IDE. Ask questions on Piazza! Don't wait until the last minute!

Assignment Objectives:

- Use the Android Studio device emulator
- Individually create a basic project that builds upon activities and layouts and includes use of menus, dialogs and touch.
- Implement a basic artificial intelligence strategy.

Delivery Instructions:

- This assignment is due by 11PM on **Monday, September 23, 2019** and will not be accepted late.
- Upload a .zip file containing the following to appropriate assignment on Canvas:
  - Your Android project directory containing all source code.
  - A `lastname-readme.txt` containing your name, CID number, and directions on how to load and run your program.
  - Your Android project should be titled `lastnameHW2`.
  - A screenshot of your gameplay activity in .jpg or .png format for competition.

Approximate Grading Distribution:

- **Test Cases** (20 points): Checking to see if your program functions efficiently and correctly on an Android Emulator simulating a Google Pixel 2 device running Oreo 8.0. Code should compile cleanly (e.g., without errors or warnings). All code should be written in Java (not Kotlin). Project should meet specifications.
- **Proper Documentation** (5 points): Code should have consistent and proper indentation. Functions should have appropriate comments (including pre/post comments). Any code inspired by zyBooks or the official Android tutorials should have clear citations within your code comments (both in the header and where the code is used). Your name, CID, and email address should appear in a header in each file.
- **User Interface / User Experience** (5 points): App has a nicely-defined user interface as per guidelines discussed in class and/or readings.
- **Game Strategy** (5 points): How well does the artificial intelligence work for the computer player

Collaboration Policy:

- For this assignment, you should work on developing your code yourself.
- You may consult the following resources:
  - zyBooks Chapters 1 – 4 (and the associated sample code)
  - https://developer.android.com/guide (including Code Labs)
  - https://docs.oracle.com/javase/8/docs/api
- You may use Piazza for any of the following:
  - Asking conceptual questions about how your code functions (or should function)
  - Asking high-level questions related to debugging or errors (but clearly indicate in your post what you've already done to attempt to debug and/or trouble-shoot)
  - Referencing code from zyBooks, Android's Developer's Guide, or the Java API.
- As a student in a 4000/6000-level course, you should have ample experience in debugging object-oriented code. Expecting extensive one-on-one assistance from anyone else (course instructor, teaching assistant, classmate both in-person or online) is not reasonable.

Part 1: Pig Dice: The Background

Pig Dice is a strategy game between two players who compete against each other by rolling dice and trying to reach a pre-determined value. The basic algorithm for this game is:

- Determine what the winning score value will be.
- Take turns rolling a die
    - If the die rolls a value of 2 through 6, the amount is added to the round's running total. The player who just rolled 2 through 6 can either:
        - Roll again
        - Bank that amount to the player's overall score and switch turns to the other player.
    - If the die rolls a 1, the player loses the round's accumulating total and control will switch to the other player.
    - If any one of the player's running total exceeds the winning score amount, the game ends.

More details on Pig Dice can be found at: https://www.whatdowedoallday.com/pig-dice-game/

Part 2: Mobile Pig Dice

You will create an Android version of Pig Dice where a human player will play against "The Banker" (e.g., a personified version of a strategic algorithm). There is some room for creativity in this assignment as well as extra credit.

Specifications:

- You will need at least two activities (and subsequent layouts)
    - On launch, you will display an "attractive" welcome screen that contains at least two buttons: *Start Game* and *About Pig Dice*
        - *Start Game* will launch a new game of Pig Dice and you will use a dialog box to get the target score.
        - *About Pig Dice* will give directions on how to play the game.
    - During a game a pig dice, you need to attractively display the following at a minimum:
        - A leaderboard that shows the human player's cumulative score and the banker's cumulative score as well as the target score.
        - The die
        - Indicator of which round and whose turn it is (e.g., *Round 1 – Banker's turn)*
        - The current banked value for the round/turn.
        - Buttons to bank or roll again.
    - Once a player wins, a dialog should display congratulating the winner and display two options: *quit* or *play again*.
    - Your computer player should have a reasonable strategy for determining to bank or roll again. You should justify your strategy in your `lastname-readme.txt` file.
- Any graphics used should be original or else permitted for use (typically with modification) royalty-free.
    - Please include sources for images as the last section of *About Pig Dice* called *Credits*. You should also include citations for these images in your `lastname-readme`.txt file

Extra Credit Opportunities:

- Implement an additional leaderboard that keeps track of matches won for the human player and banker player.
- A prompt to enter in the human player's name and subsequently displaying that in the leaderboard.
- Implement a spinner that prompts for various skill levels for the computer player:
    - Random where the algorithm should randomly choose banking or rolling.
    - Average where the algorithm should act how you planned out in the basic version of Pig Die
    - Cut Throat where you come up with an advanced algorithm that is more likely to win over the average player.
    - Your strategies for your skill level algorithms should be documented within your lastname-readme.txt file.

Sample Screenshots:

- Please monitor Canvas for sample screenshots of one possible solution.

Note, various choices related to graphical design (e.g., font selection, color choices, and layouts) are up to you. Just make sure you leverage good design practices as discussed in class.

Additional References:
https://zyBooks.com
https://developer.android.com/studio/install
https://developer.android.com/studio/projects/create-project

Hints:

The following links may provide some helpful guidance in trouble-shooting as you work through your app design
https://unsplash.com/license