Project Overview

Each student shall **incrementally design, code, and test** their software application based on the requirements stated for a project increment (aka spiral). The first three project increments are defined by the instructor. After this, the student follows their plan (developed after the third increment) to complete their project.

As part of this first increment, the student shall select the game they would like to design and implement. The choices are:

- Cosmic Wipeout, a dice game
- Tunk, a card game

The rules for each game have been posted in Canvas.

Artifacts for Increments #1

- The total number of hours it took the student to complete the increment.
  - Hint: It will benefit you if you take the time to keep track of the hours you spent designing, coding, and testing each increment. Splitting up your total hours into these three buckets will help you produce a better project plan that you develop after the third increment.
- In one zip file:
  - A dia file that contains a class diagram.
  - A dia file that contains a state machine diagram that describes the human-computer interactions.
  - Java source code files.

Grading for Increment #1

The rubrics used to assess your artifacts are**:**

- **Code**: all evaluation criteria are used.
- **Design**: only the ***Modeling techniques***, ***Design artifacts***, and ***Implementation*** criteria are used.

Scope of Increment #1

Mandatory requirements are stated using *shall*, while optional requirements are stated using *should*. The requirements are in no particular order, but they are numbered for easy reference.

1. Your game shall be played by one human player.

2. While the game is being played:
   2.1. The software application shall exit only after the human player has won the game.

3. For each human player's turn, the:
   3.1. **Cosmic Wipeout game** shall:
      3.1.1. Allow the player to roll all five dice.
      3.1.2. When the roll results in a score using only the rules listed below, add the appropriate turn score to the human players total score.
         3.1.2.1. Produce a turn score only for rule 2.a (i.e., each 5 or 10 rolled) and 2.b (i.e., three 5's or three 10's rolled). However, do not include a flaming sun as described in rule 2.b in this increment (i.e., do not score for a pair of 5's or a pair of 10's along with a flaming sun).
      3.1.3. When the roll results in no score, the player's turn has ended and their total score is not changed.

3.1.4. Play continues for this one human player until they have won the game.

3.2. **Tunk game** shall:

3.2.1. Allow the player to deal seven cards. (Note: in this first increment, shuffling the deck of cards is not necessary.)

3.2.2. The player will immediately call "tunk" indicating the end of this hand. (For this increment, you do not implement any portion of requirement 6.)

3.2.3. The player will add up all of their card values, ignoring the possibility of having matching sets. This hand total is added to the human player's total score. (Note: in this first increment we are ignoring matching sets and are not dealing with the multi-player aspects of how *other* players may play on a tunker's matched set.)

3.2.4. Play continues for this one human player until they have lost the game.

Hint: Your user interface for this increment has no need to prompt the user for any information. The entire game may be played without intervention from the user (i.e., human player).