COMP 4004 Individual Assignment 3
Due **December 8th 2023 11:59PM**

You are given the posted web-based implementation of the game Crazy Eights.
You do NOT need to understand all the details of this implementation in order to do this assignment.
But it is a crucial industrial skill to be able to "understand enough" to be able to test if not improve code.
You should play the game, identify the small bug I introduced in it, and correct it in file **GameLogic.java**
Then you are to develop Selenium tests (in file **AcceptanceTest.java**) corresponding to those specified in the posted grid to fill. **Please note that the posted code does include the pom file to use.**

Description of the basic card game: (also watch https://www.youtube.com/watch?v=iDQjn3k76Mw)
The game consists of several rounds, each round concluding by computing the score of *each* player for that round and updating their game scores accordingly (so that everyone sees everyone's scores). In a round, five cards are initially dealt to each player. The remaining cards of the deck are placed face down at the center of the table as the *stock pile*. The top card of this stock pile is then turned face up to start the game as the first card in the discard pile. This top card cannot be an 8 and, if this is the case, that 8 is put back randomly in the deck and a new top card is picked.

In a round, on their turn, a player discards a card by matching rank **or** suit with the top card of the discard pile. Alternatively, a player can play any 8, which allows that player to then select the suit that the next player is to play. That next player must then match the named suit or play another 8. If a player is unable to play, that player draws, one at a time, up to 3 cards from the stock pile. A card that is drawn and that can be played **must** be played (after which that turn ends). If still unable to play after drawing 3 cards, that player ends their turn. At the start of their turn, a player can choose to draw even if they have a valid card to play.

A *round* is over as soon as one player plays their last card. In this case, that player scores 0 for the round. Each other player scores the total value of the cards in their hand as follows: an 8 is worth 50 points, kings, queens and jacks are worth 10 points, and all other cards are worth their face value. The scores for the current round are added to those of the previous rounds. The game is over once any of the players reaches 100 points. The winner is the player with the lowest score once the game is over. All players are then notified about who is the winner.

Alternatively, a round is over once the stock pile is exhausted and no player can play another card. That is, even if the stock pile is exhausted, players continue their turns until no player can play another card. In this case, all players score the total value of the cards left in their hand.

Additional Special Behavior Cards:
a) *Queens*
Playing a Queen causes the next player to miss their turn.
b) *Aces*
Playing an Ace reverses the direction of play (i.e., modifies who is the next player to play).
c) *Twos*
Playing a Two forces the next player to draw two cards and then play their turn **unless** they can immediately play legally 2 cards (ie the first one matches rank or suit of the Two and then the second card matches rank or suit of the first card just played).If a player plays 2 cards in response to a 2 (ie does not draw), their turn ends immediately after.
Playing a Two on a Two forces the next player to play legally 4 cards or immediately draw 4 cards.

Order of players' turns in a round:
The order in which the players join a game determines their initial sequence of play. That is, the first player to have joined plays first in the first round, followed by the second player to have joined, etc. In the second round, it is the second player to have joined the game who starts that second round, followed by the third player to have joined, etc.

User Interface
The web-based interface allows each player to see their cards, the top card of the discard pile, and the score of all players. The interface also indicates whose turn it is. Finally, at the end of the game, the winner is identified in each player's browsers.

Card naming convention of the test plan
The test plan refers to cards as follows: S for spades, C for Clubs, D for diamonds and H for hearts.
So 8H refers to the eight of hearts.

The test plan refers to 1 for Aces and 10 for tens. The game instead uses A and T respectively.
Also note that the test plan systematically uses 4 players.

Requirements:

1) Each test case should be named **testRow**<a row number from the grid>. So for row 25, the corresponding test case is testRow25()
2) In order to facilitate correction, except for the test case of PART 3, the rigging of each test case, that is, setting up the initial 5 cards of each of the 4 players, should not be in the test case but in a separated procedure called **rigTestRow**<a row number>. So for row 25, procedure testRow25() must call rigTestRow25().
3) When the test plan states something like “Player4 can't play and must draw; draws 2C”, it is sufficient to assert that 2C is drawn (ie becomes part of the hand of that player). In other words you need not assert that the clicking on any card of the hand of that player is disabled prior to drawing 2C. Also, when a player draws a card and must play it, it is sufficient to assert that that card is now the top card and no longer in the hand of the player.
4) You are to make a video that shows you identifying yourself, downloading the game from your new **PRIVATE** repository, running all 3 parts of my acceptance test suite **in one shot**. Thus you must create and invite the TAs and myself to your new repo for A3. And do submit a *properly-named* filled grid to Brightspace.

Hints:

1) You should take advantage of the @BeforeEach and @AfterEach annotations to have a setup and teardown shared by all your test cases.
2) You absolutely want to listen to Abhai’s video on the assignment at https://youtu.be/Hrh2P-PV2eE. The documentation links are in the video’s description.
3) A timeout of 20 second is recommended to be able to follow a bit what’s going on as the whole test suite executes. Each of your 4 web browsers will need to set this up. Please be aware that there is **no** Chrome web driver for Chrome version 117 and up.
4) In a test case, after playing cards, you need to assert for each player they all have the same top card and turnId (ie the id of the player’s whose turn it is). You also need to assert the direction of play. And for some test cases you should assert that the draw button is enabled.
5) Do not be surprised that many test cases will be very similar except for the cards played.