Use Case: Play game

1. Player enters 0, 1 or 2 arguments to set the number of tanks, and game mode (optional). assume the first argument is an integer and as # of tanks, 2nd argument is of cheating and it should be case insensitive. If user enters 0 argument, default to N=5 tanks.
2. The program randomly places the tanks on a 10 x 10 grid board with all cells. If Player provide second argument indicating cheating, the tanks will be represented in the grid system as letters, otherwise they will be represented as '~'
3. System speaks prompt:
    a. Enter your move:
4. Player enters a move in the form of <Letter><Number> and is told if it hits or misses.
5. The program shows player the remaining structural integrity of their fortress and a map of what is known about the game-board.
    a. X indicates a tank is hit
    b. ' ' (space) indicates a miss.
    c. Player is shown how much damage is suffered for each enemy shot.
6. Continue step 3 and step 4 until either
    a. User has destroyed all the tanks on the grid system
       or
    b. User's fortress is destroyed by enemy tanks.
7. If user wins by destroyed all the tanks, system prints out "Congratulations, you win!". If user loses with fortress being destroyed, system prints out "You lost, better luck next time."

Variation#1

1.1 In step 1, player enters invalid # of arguments.
1.2 The program prints "Please enter valid number of arguments".
1.3 System exits with failure

Variation#2

2.1 In step 2, not all tanks specified in first argument can be placed on the board.
2.2 System exits with failure

Variation#3

3.1 In step 4, player enters a invalid move.
3.2 The program prints "Invalid target. Please enter a coordinate such as D10".
3.3 Continue with step 3.

Variation#4

4.1 In step 1, player enters invalid type of arguments.
4.2 The program crashes