Play game use case

I/ Information

1. Name: User play game
2. Actor: User (player)
3. Precondition: The program is running right
4. Trigger: User starts the game
5. Post-condition: The user finishes the game, either by winning or losing

II/ Steps

1. The user enters in the command line the number of tokimon, fokimon, and game mode. If the command line doesn't specific any field, the default value will be used for that field (the default value is 10 for tokimon, 5 for pokemon, and false for cheat mode.)
2. The system will create a grid of 10x10 cells, label the row from A ->J and column from 1 -> 10 with the visibility is unvisited, and populate in the grid with tokimons and fokimons randomly.
3. The system will print "Please enter the initial position for trainer (Eg: B.5): " to ask trainer for initial position
4. The user will enter the position for trainer's initial position
5. The system will validate the chosen position
6. The system will create a new trainer and put him into the initial position
7. The system will update the game grid with the new trainer
8. The system check if trainer face a fokimon at the initial position
9. The system will prompt "Please enter either W, A, S, or D to move up, left, down, or right from current position or enter 2 to choose using spell: "
10. The user will enter his choice
11. The system will validate the user choice
12. The system will process the chosen action
    a. If user choice is 'A' or 'D' or 'S' or 'W' (case insensitive):
        1.1. The system validate the user input
        1.2. If user enter 'A', move trainer to the left 1 unit
            If user enter 'D', move trainer to the right 1 unit
            If user enter 'W', move trainer up 1 unit
            If user enter 'S', move trainer down 1 unit
    b. If user choose '2' (use spell):
        2.1. The system prompt
        "Please enter 1, 2, 3 to choose option for spell:
            1. Jump the player to another grid location.
            2. Randomly reveal the location of one of the Tokimons.
            3. Randomly eliminate one of the Fokimons."
        2.2. The user enter the choice
        2.3. The system validate the user input
        2.4. The system process the input according to user choice
            If user enter '1', trainer will be moved to another random position
            If user enter '2', a tokimon will be chosen randomly, and system will print "Reveal: There is a tokimon at position "
            If user enter '3', a fokimon will be chosen randomly and delete
13. The system will update the game grid according to the user choice for that turn

14. The system check whether the user win or lose at that turn
    a. If the trainer collect all the tokimons, the system
        1.1. Display the complete grid
        1.2. Print “Hurray! Trainer has collected all the tokimons. You are winning!!"
        1.3. The user will exit the system
    b. If the container face a fokimon, the system
        1.1. Display the complete grid
        1.2. Print "Oops! Trainer faces a fokimon. You are losing!!"
        1.3. The user will exit the system
15. The system will display the status of grid.
16. The system will display the number of tokimons remained, fokimons remain and number of spells remained.
17. Continue from step 9 to step 16

Variation #1 (Step 1):

E1. Invalid argument
1. In step 1, user enter more than 3 arguments in command line
2. The program prints “Error: Please enter from 0 to 3 arguments”
3. The system exists with failure

E2. Invalid argument (2)
1. In step 1, any argument in command line does not start with “--numToki=“ or “--numFoki=” or “--cheat”
2. The program prints “Error: Invalid argument”
3. The system exists with failure

E3. Invalid tokimon/ pokemon number
1. In step 1, the number of tokimon or fokimon in command line is less than 5
2. The program prints "Error: The number of fokimon and tokimon has to be at least 5!"
3. The system exists with failure

E4. Invalid tokimon/ pokemon number (2)
1. In step 1, the sum of number of tokimon or fokimon in command line is more than 100
2. The program prints "Error: The number of both tokimon and tokimon has to be at most 100!"
3. The system exists with failure

Variation #2:

A1. Cheat mode
1. In step 2, if the command “--cheat” is in the argument, all the cell visibility will be assigned visited

Variation #5:

A1. Invalid initial position
1. In step 5, the entered input is invalid
2. The system prints "Invalid initial position. Let’s try again!"
3. Go to step 3

Variation # 8:

A1. Trainer meet a fokimon

1. If the trainer meet tokimon at his initial position
2. The system
    1. Display the complete grid
    2. Print "Oops! Trainer faces a fokimon. You are losing!!"
    3. The user will exit the system

Variation #11:

A1. Invalid input

1. The input is neither ‘A’, ‘S’, ‘D’, ‘W’, or ‘2’
2. The system prints "Invalid input. Let’s try again!"
3. Go to step 9

A2. No spell left

1. The user input is “2” but there is no spell left
2. The system prints "The trainer has used all spell. Let’s try again!"
3. Go to step 9

Variation #12 - A:

A1. Move out of grid

1. In step 12, point 1.1, the user input make trainer out of the grid
2. The system will print "Invalid move. The trainer will be out of grid. Let’s try again!"
3. Go to step 9

Variation #12 - B:

A1. Invalid spell choice

1. In step 12, point 2.1, the user input is neither “1”, “2”, or “3”
2. The system will print "Invalid choice for spell. Let’s try again!"
3. Go to step 12, point 2.1