

### Define the Problem:

The user is at the screen where his or her character has just contracted a disease on the frontier! He/she now has to choose what to do. Choosing to do nothing, means nothing will happen, however choosing to apply no medicine will also allow nothing to happen. The program needs to run a random number in the stamina slot against a random number in the medicine slot, all of which contribute to it's overall efficacy of healing the user.

It needs to have the proper variables input in the proper fashions in order to work properly. If a foul character, or a zero

### Test Case for Disease Contraction

input	input	output	output	output
stamina	medicine	stamina heal chance	Heal Chance	return
4	5	47	2	The medicine didn't really help
0	21	Game Over	Game Over	Output: "Game Over"
14	52	82	8	The medicine didn't really help
99	85	27	50	You are feeling better!
-12	10	ERROR	ERROR	ERROR
12	0	You will be unable to heal	You will be unable to heal	Oh no! You don't have any medicine!
VARCHAR	12	ERROR	ERROR	is not a number

The First case, 4 stamina and 5 medicine will be run into the takeMedicine constructor.

Case1: The formula would read something like this –  $4 * .5 = \text{healChanceStr}$ (which is 2), then we use the 2 and multiply it by 1 to make 2, due to the medicine being multiplied by 10 was far to strong. We also needed to just have medicine around, so having more than 1 item is important, but has the same effect as having 100 medicine. 2 would equal chanceToHeal, this is intended to something akin to you are being really weak because of your condition or previous travails. Then we use the math.random function and give it a ceiling of 100 and set it equal to num. Num will then be checked against chanceToHeal. The time I ran it just now it was 47, so then it would spit out "The medicine didn't really help."

Case 2: No stamina – You are always unable to heal without any stamina. You wouldn't be able to make it this far as deep within the character class, there's a note in there that when stamina hits zero, the game is set to over. – So you would see the game over screen.

Case 3: Again,  $14 * .5 = 7$ . Still a really dire situation, and you should probably rest up before you attempt the heal. 7+ some medicine of 1 makes 8. 8 is less than the random number of 82, therefore same as Case 1. "The medicine didn't really help."

Case 4: 99 stamina divided in half (because you're sick and for balancing) +1 so  $50 > 27$  (got lucky I guess) Then the output would read "You are feeling better!"

Case 5: -12 – If somehow there was an erroneous value passed into the service, this would probably crash the whole game because you've gone beyond the limit for allowable stamina. You'd see the game over screen.

Case 6: This would immediately output the exception – "Oh No! You don't have any medicine!" By selecting in the view that you'd like to try to heal yourself, but without medicine, it kicks you right back out and it won't make it to the else statement.

Case 7: This is handled in the view, but if you were to put an inappropriate letter in where a number should go, it would read – " X Is not a number"

The errors are handled in the error controller views!