Section 0: Initial Design

```
● Prompt user for input:
    ○ # of vampires, n
    ○ Seed number
● Int lifeSteal = [2] #position 1 = vampire, position 2 = roll
● Create a 1d array, lives, size n, (number of vampires), each item in array is constant int 3 to begin.
● While Loop (condition: while true)
    ○ aliveCount = 0 #number of vampires still alive
    ○ For i in range(len(lives)){
        ■ If lives[i] > 0:
            ● aliveCount++
    ○ If aliveCount == 1:
        ■ Do end stuff
        ■ break
    ○ For i in range(len(lives)):
        ■ If lives[i]  == 0:
            ● Continue;
        ■ Else:
            ● Roll dice
            ● If midnight:
                ○ left(i,n) right(i,n) sparkle/ressurect
            ● If diceroll < lifeSteal[1]:
                ○ lifeSteal = [i,diceroll]
    ○ Lives[lifeSteal[0]] -= 1;
● }
```

Section 1: Revised Design

- Prompt user for input, accounting for errors:
    - # of vampires
    - Seed number
- Create a 1d array, lives, size n, (number of vampires), each item in the array is constant int 3 to begin.
- While Loop (condition: while true)
    - Int lifeSteal = [2] (array that holds the lowest roll of the current round, slot 1 = vampire location, slot 2 = roll
    - aliveCount = 0 #number of vampires still alive
    - Loop through # of vampires in play{
        - If their life count is greater than 0:
            - aliveCount++
    - If number of vampires alive is only 1, (aliveCount == 1)
        - Loop through lives array to find the life above 0
            - Print that names[iterator] wins the garlic games!
            - break
        - **Break *(THIS WILL END THE GAME)***
    - Print the round number
    - For the amount of vampires in play:
        - If their life count is equal to 0:
            - Do nothing and continue the for loop;
        - Else:
            - Roll dice
            - Print correct text for the vampire roll
            - If roll is midnight:
                - left(i,n) right(i,n) sparkle/resurrect
            - If diceroll is lower than current stored # in array lifeSteal
                - lifeSteal array will be changed to store new vampire position and roll #
    - At the end of the round, go to array lives and decrease the life corresponding to the second element in lifeSteal by 1
    - Print Correct message depending on roll

- Increment round counter
- }