**Project 1: PokemonGO!**

Pokemon Go! (feel free to download the app if you want to experience it) is a game with a lot of possible things. One instance of gameplay goes as follows, and is what we will be coding. This assignment should highlight your mastery of **conditionals, soliciting input, delivering information, FOR/WHILE loops,** and the **GoTo** function.

When the game opens up, there is a random number of pokemon around you (typically 0-6). You have the option to engage with any one of those pokemon. If there are no pokemon, you can choose to move your location and search for pokemon again. The pokemon you select to engage with has a *(displayed)* random strength level, and moves in a random pattern. You select a pokeball of either strength 1, 2, or 3 (weakest to strongest). The pokeball you select may make the pokemon you are attempting to catch easier/more likely to be caught, but it may not. You can only throw as many pokeballs as you have in your bag. You then throw the pokeball and hope that it (a) hits the Pokemon and (b) is able to keep the pokemon trapped. Part (a) can happen and Part (b) fail, but part (b) cannot happen if part (a) fails, which it does sometimes. If you capture the Pokemon, you will be told the Pokemon’s name and given the chance to store the pokemon, or to release them. Once the interaction is over [either you decided to stop trying, or you were successful in catching the pokemon, or you ran out of pokeballs and had to stop], you are taken to your location map again and there is again a random number of pokemon around you. At this home screen, there could also be a PokeStop (0-3) – when the user interacts with the PokeStop, they win a random number (1-8) of a specific type of Pokeball. The game play can continue until the user decides to close the game, or until you have completely run out of pokeballs, or if you have run out of space in which to store the Pokemon you decide to keep [note: you are not allowed to interact with a pokemon if you have no more storage space].

For this assignment, assume:

* Every time you “resurface” at the home/location screen, there should be a new random assortment of Pokemon for you to interact with
* You begin the game with a random number of pokeballs
  + Strength 1: 10-20
    - Can possibly capture within a range of (50-250)
  + Strength 2: 2-12
    - Can possibly capture within a range of (50-400)
  + Strength 3: 0 – 8
    - Can possibly capture within a range of (50-600)
* You have 10 storage slots available
* You should give the player the chance to choose a throwing position between 0-10. The pokemon should be located at a position in this range
* Each Pokemon is either stationary or moving, which impacts their chance of being “hit”
  + If the Pokemon is active, the player must choose a number +/- 1 of the pokemon’s position
  + If the Pokemon is stable, the player must choose a number +/- 3 of the pokemon’s posiiton
* Each Pokemon has a random strength assignment between (50-800)
* Pokemon within a ball’s range have a 80% chance of being caught
* Pokemon outside of a ball’s range by 1 range have a 55% chance of being caught
* Pokemon outside of a ball’s range by 2 ranges have a 20% chance of being caught
* Pokemon outside of a ball’s range by 3 ranges have a 5% chance of being caught
* Users can stop/exit/quit whenever they wish, and should have to back out of each “stage” of the game in order to exit the game completely
* There are at least 8 possible Pokemon to catch, and each type can be any strength

When a player exits the game completely, they should be given a play report of their time. This report should tell them how many of each type of ball they have remaining, how many Pokemon they caught, how many they kept, and should list the names of all of them.

Coding Note: Use input validation techniques for every inputbox/msgbox. User’s should not be able to break the game. You should also include a sufficient amount of comments so that someone reading your code could follow the logic.

**Submission:** You will need to upload a photo of your final pseudo code, as well as your .xlsm file, and a PDF file of your code.