COMPSCI 119

Lab #4

Hunt the Hurkle

Professor William T. Verts

Back in the early days of small computers, one of the first small games that was written was called "Hunt the Hurkle" (not to be confused with "Hunt the Wumpus" which was a similar small game). A hurkle is a creature that hides at random in one cell of a large two-dimensional grid. The player has a small number of guesses to find the hurkle.

Part #1: The Overall Design

The basic outline of your code must be:

```
# Your name

def GetANumber(Message):
    ?????                          # Your code goes here
    return Result                  # The value returned from GetANumber

def Main()
    ?????                          # Your game play code goes here
    return
```

Part #2: The `GetANumber` function

The first part of the program is to complete the function called **`GetANumber`**. This function displays the message in its parameter list asking the player for an integer between 0 and 999, gets that number from the player, but asks repeatedly for another number if the number entered was less than 0 or greater than 999. This function *must not crash* if the player enters something that is not an integer or not a number (use the **`try-except`** statement to filter out bad inputs). Once the player enters an acceptable value, return that value as the **`Result`** variable.

Part #3: The Hurkle's Position

The next part of the program needs to *select the position*, at random, of the hurkle in the grid. The grid is 1000×1000, so you need to pick the X position of the hurkle at random from 0 to 999, and the Y position at random from 0 to 999 as well. Location <0,0> is the upper left corner of the grid, and <999,999> is the lower right corner of the grid. You'll need to add the statement **`import random`** at the start of the program (see page 261 of the Companion for a list of random-number functions).

Part #2: Game Play

The final part is to play the game. The game asks the player for their guess of the X coordinate by calling **`GetANumber("Enter X from 0 to 999")`** and then asks for the Y coordinate guess by calling **`GetANumber("Enter Y from 0 to 999")`**.

If the guess is incorrect, the hurkle will tell the player to "Go northwest", "go south", "go southeast", "go west", etc., for all eight possible directions. If the player's guess is correct, the hurkle says "You caught me!" and the game ends.

The player has *no more than ten* guesses to find the hurkle. If the player is incorrect all ten times, the hurkle says "I got away!", prints the actual location of where it was, and the game ends.

Recommendation for Testing

When testing your program, I recommend that you print the hurkle's position at the start of the program ("cheat mode"). By doing so you can tell if your directions to "go north", "go southwest", etc., all work correctly. Once you are certain that all eight directions are correctly handled, only then should you *remove the print* of the hurkle's position to make game play fair and correct.

Turning in the Program

Make sure that your name is in a comment at the top of the program. If your name is not present your program will receive zero credit.

Do not turn in your program until it works completely; a non-working program will also receive zero points.

When your program is complete and has been *thoroughly tested* submit it through the on-line form as Lab #4.