Returning to the Guess My Number Game

The Guess My Number game combines many of the concepts you learned in this chapter. But, more importantly, it represents the first full game that you can use to show off to your friends, family, and members of the opposite sex.

Planning the Program

To plan the game, I wrote some pseudocode first:

```
pick a random number
while the player hasn't guessed the number
  let the player guess
congratulate the player
```

This isn't a bad first pass, but it's missing some important elements. First, the program needs to tell the player if the guess is too high, or too low. Second, the program should keep track of how many guesses the player has made and then tell the player this number at the end of the game.

HINT It's okay if your first program plan isn't complete. Start planning with the major ideas first, then fill in the gaps until it feels done.

Okay, here's a refinement of my algorithm:

```
welcome the player to the game and explain it
pick a random number between 1 and 100
ask the player for a guess
set the number of guesses to 1
  while the player's guess does not equal the number
    if the guess is greater than the number
       tell the player to guess lower
    otherwise
      tell the player to guess higher
    get a new guess from the player
    increase the number of guesses by 1
congratulate the player on guessing the number
let the player know how many guesses it took
```

Now I feel ready to write the program. Take a look over the next few sections and see how directly pseudocode can be translated into Python.

Creating the Initial Comment Block

Like all good programs, this one begins with a block of comments, describing the program's purpose and identifying the author:

```
# Guess My Number
#
# The computer picks a random number between 1 and 100
# The player tries to guess it and the computer lets
# the player know if the guess is too high, too low
# or right on the money
#
# Michael Dawson - 1/8/03
```

Importing the `random` Module

To be fun, the program needs to generate a random number. So, I imported the `random` module:

```
import random
```

Explaining the Game

The game is simple, but a little explanation wouldn't hurt:

```
print "\tWelcome to 'Guess My Number'!"
print "\nI'm thinking of a number between 1 and 100."
print "Try to guess it in as few attempts as possible.\n"
```

Setting the Initial Values

Next, I set all the variables to their initial values:

```
# set the initial values
the_number = random.randrange(100) + 1
guess = int(raw_input("Take a guess: "))
tries = 1
```

`the_number` represents the number the player has to guess. I assign it a random integer from `1` to `100` with a call to `random.randrange()`. Next, `raw_input()` gets the player's first guess. `int()` converts the guess to an integer. I assign this number to `guess`. I assign `tries`, which represents the number of guesses so far, the value `1`.

Creating a Guessing Loop

This is the core of the program. The loop executes as long as the player hasn't correctly guessed the computer's number. During the loop, the player's guess is compared to the computer's number. If the guess is higher than the number, `Lower. . .` is printed; otherwise, `Higher. . .` is printed. The player enters the next guess, and the number of guesses counter is incremented.

```
# guessing loop
while (guess != the_number):
    if (guess > the_number):
        print "Lower..."
    else:
        print "Higher..."

    guess = int(raw_input("Take a guess: "))
    tries += 1
```

Congratulating the Player

When the player guesses the number, `guess` is equal to `the_number`, which means that the loop condition, `guess != the_number`, is false and the loop ends. At that point, the player needs to be congratulated:

```
print "You guessed it! The number was", the_number
print "And it only took you", tries, "tries!\n"
```

The computer tells the player what the secret number was and how many tries it took the player to guess it.

Waiting for the Player to Quit

As always, the last line waits patiently for the player to press the Enter key:

```
raw_input("\n\nPress the enter key to exit.")
```