**GUESS THE NUMBER GAME REPORT**

**Introduction:**

The main aim of this project was to create a fun Number Guessing Game. Players are challenged to figure out a four-digit number that's picked randomly. To make the game even more enjoyable, it gives players hints in the form of 'circle' and 'x' symbols, nudging them in the right direction. Another goal was to use the pytest framework for testing the game automatically, making sure it works correctly. Overall, this project demonstrates how Test-Driven Development (TDD) can be a great way to build a reliable game by testing each part thoroughly.

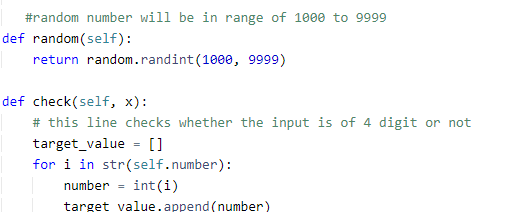
**Requirements:**

* Generates a random number of 4 digits
* I have added a function through which user can try for 15 times
* If player tried 15 wrong tries than it will display the correct random number
* After providing random number if you enter the correct number which is given, then you will know after how many attempts you have got the right number.
* It also Provides hints in the form of 'circle' and 'x' symbols.
* Display the number of attempts taken after the game ends.
* Allow the player to quit the game anytime.
* Ask player to play again

**Process:**

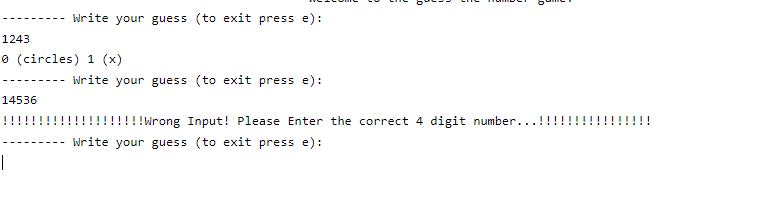
* Create a Git directory for the project and initialize a new Python file for the game implementation.
* I employed a Test-Driven Development (TDD) methodology to create the game. This method entails crafting tests for the intended features before generating the concrete code. Subsequently, the code is systematically developed and improved to fulfill these tests. To accomplish this, I leveraged the pytest framework to construct and execute automated unit tests.
* **Generate the Random Number:**

I started by creating a test to make sure that the random number generated has four digits. Here is the test function I wrote:



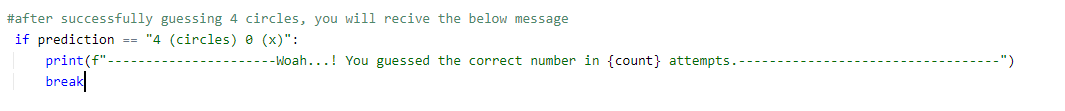
* **User Input Error:**

I have implemented a condition as well that will identify whether user that given input as per condition, else it will display a message which is shown below:



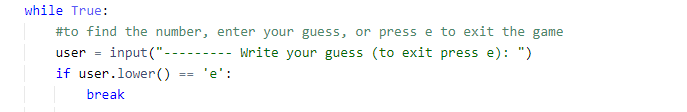
* **User Guess and Comparison:**

Write test cases to handle player input and validate the guesses. Implement the code to process the guesses.



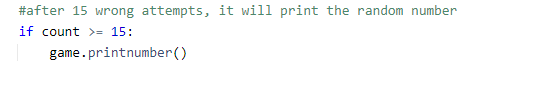
* **Game Loop and Quitting:**

I tested the game loop and quitting functionality. I have used an ‘ e’ key in order to exit from the game. Here's an example of the test for quitting:

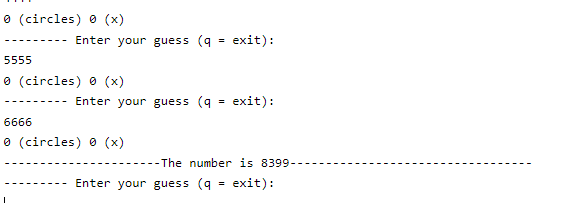


Displaying the random number after limited trials

I have also added a function in which the random number will be displayed after 15 wrong trials. After trial we can also enter the previous guess and it will tell how many attempts it took to guess the correct number.The pictures are shown below:

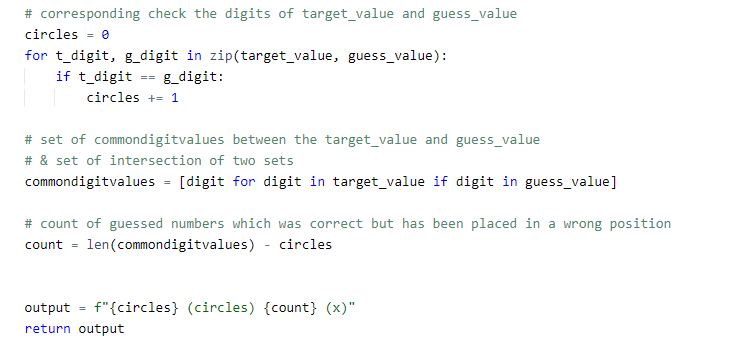


Output



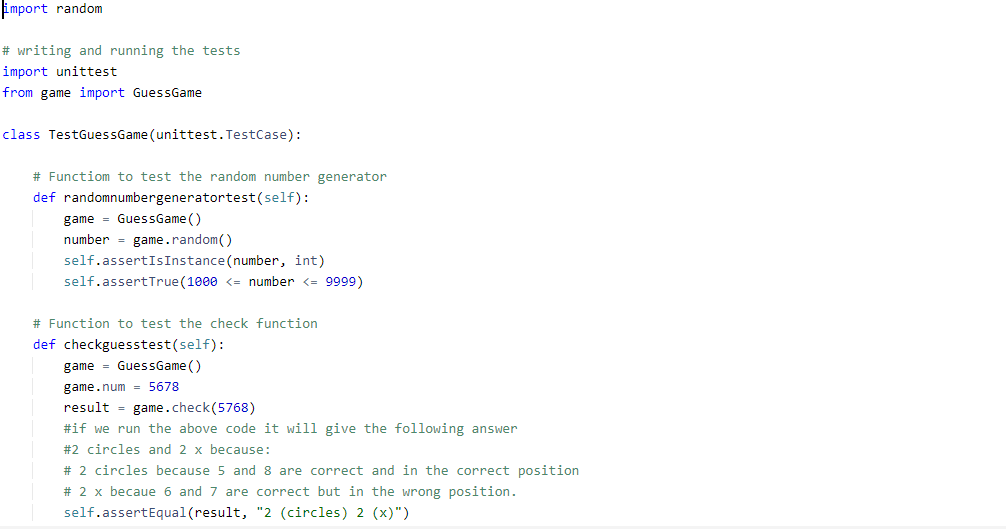
* **Main Game Logic:**

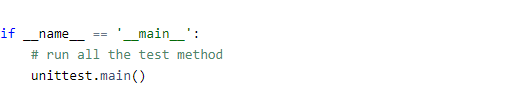
I tested the main game logic, including correct number guessing and win conditions. Here's an example:



* **Automated Testing Using the unittest Library:**

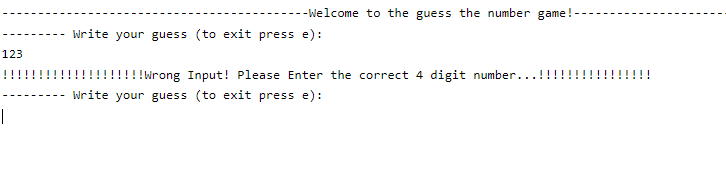
In software development, testing is a critical process to ensure the correctness and reliability of your code. One widely used testing framework in Python is unittest, which allows you to define and execute unit tests to verify the behaviorof your code.

****

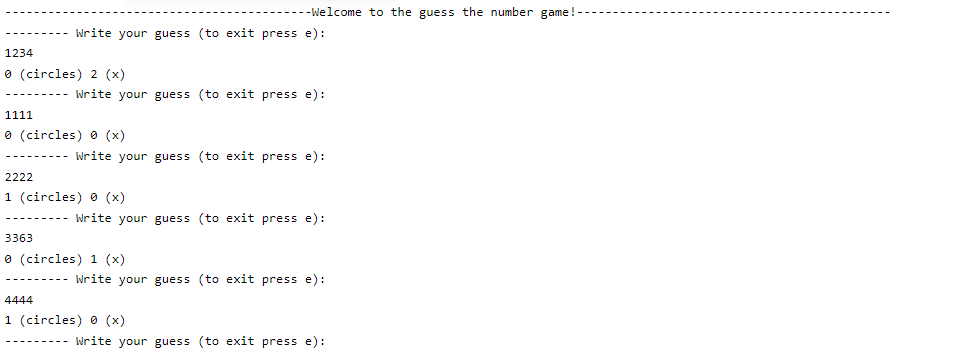
****

* **Screenshot of Output:**

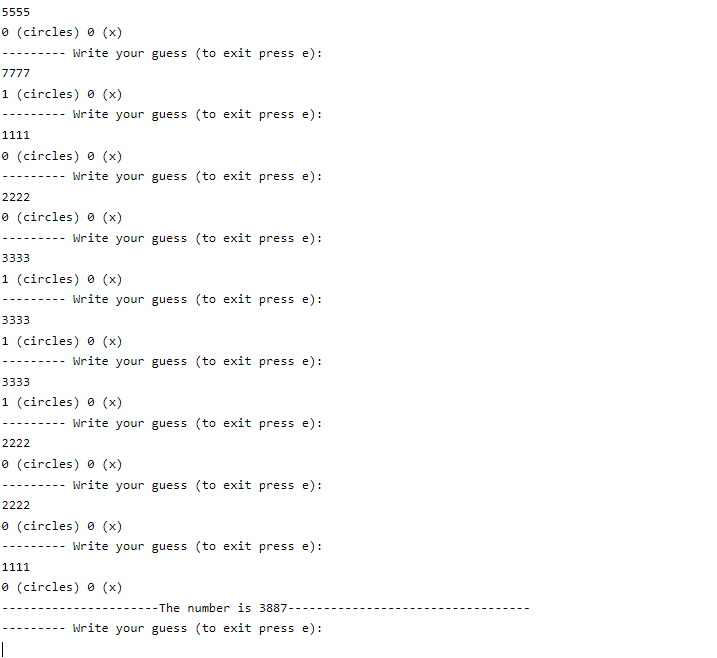
First, if we provide an input less than 4 digits it will provide a message



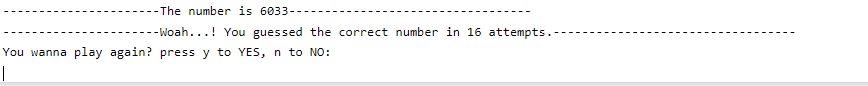
Later we will give a correct input which is of 4 digits and as per the number it will provide circles and x :



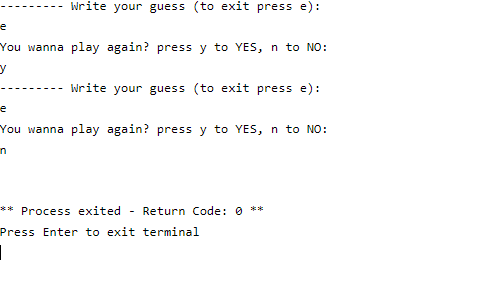
After 15 wrong trials it will automatically display a random number



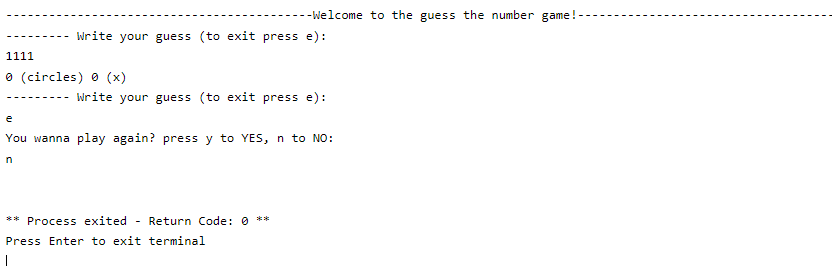
After providing the random number it will ask to enter guess and when we successfully provide the guessed number it will shows a message with attempts which is given below



Later, it will also ask you to play again and if you want to play again press y and if you don’t press n



It has also a function through which user can exit the game by pressing e



**Conclusion:**

I have created a game named “guessing the number” using a approach called Test Driven Development (TDD). This method helped me to create a fully functional game step by step and the tool that I have used is known as “unittest” that helped me to test the game. I have learnt how much TDD is important for the development process. I have also used pytest framework to make automated tests in order to run the different parts of the game.

**GitHub Link**:

**https://github.com/deepak364/Assignment2-guess-the-number/blob/main/main.py**