

# Testing

---

## Examples - Regression Testing :

---

### Example 1

#### Objective:

The goal of this test code is to validate the game logic of our Hangman game. This might involve testing whether the game correctly determines if a guessed letter is in the word and updates the game state consequently.

#### Test Code:

```
[TestMethod]
public void HangmanGame_GuessCorrectLetter()
{
    // Arrange
    var NewGame = new HangmanGame("Scotland");

    // Act
    var result = NewGame.GuessLetter('S');

    // Assert
    Assert.IsTrue(result);

    //Assert.AreEqual(" S _ _ _ _ _ ", NewGame.CurrentState()); not finished.
}
```

#### Explanation of the Test:

The test simulates a scenario where a player guesses the letter "S" correctly in the word "Scotland." It sets up a new Hangman game, invokes the `GuessLetter` method with the correct guess, and then checks the result of the guess using the `Assert.IsTrue` method.

#### Importance of Testing:

This test is important because it verifies the core game logic related to letter guessing in the Hangman game.

#### Limitations:

One limitation of these tests is that they cover only the basic game logic.

---

### Example 2

#### Objective:

The specific test scenario focuses on regression testing for verifying if the game correctly detects when the entire word has been guessed. It simulates a series of correct guesses until the entire word is revealed.

### Test Code:

```
[TestMethod]
public void HangmanGam_WordFound()
{
    var NewGame = new HangmanGame("Scotland");

    NewGame.GuessLetter('s');
    NewGame.GuessLetter('c');
    NewGame.GuessLetter('o');
    NewGame.GuessLetter('t');
    NewGame.GuessLetter('l');
    NewGame.GuessLetter('a');
    NewGame.GuessLetter('n');
    NewGame.GuessLetter('d');

    var WordGuessed = NewGame.IsWordGuessed();

    Assert.IsTrue(wordGuessed);
}
```

### Explanation of the Test:

The test, `HangmanGame_TestDeRegression_DevinetteDuMot`, simulates a scenario where the player guesses the letters of the word "Scotland" correctly. Each correct letter guessed are using the `GuessLetter` method. After the entire word has been correctly guessed, the test checks the value returned by `IsWordGuessed` to ensure that it returns `true`, indicating that the word has been successfully guessed.

### Importance of Testing:

This test is important because it verifies the core game logic related to letter guessing in the Hangman game.

---

These two pieces of code demonstrate that the game itself works. It detects when a letter is correct and when the word is found. Now, we should add tests for the user interface: the number of players, advancing to the next player when the current player guesses incorrectly, etc.