1. I created a public method “boolean updateState (char ch)” which takes a char guess and updates the state of the game. The return type is Boolean and indicates whether the character is present in the originSecretWord.

Tests created:

* testUpdateCorrectState ()

This test asks to update the state with a correct guess. It should correctly update the state and return true.

* testUpdateInCorrectState ()

This test asks to update the state with an incorrect guess. It should not make changes in the state and return false.

* testUpdateRepeatedCorrectState ()

This test asks to update the state with a repeated, correct guess. It should not make changes in the state and return true.

1. I used extract superclass and pull up method patterns to refactor NormalHungman and EvilHungman classes. The reason behind my decision is that they implement the same interface, and some of the methods are exactly the same. Hence, the methods can be defined in a common abstract superclass (pull up method). The common superclass also contains the fields that both of the classes contained. Note, that some of the fields and methods were essentially the same but had different names. I modified some of the names such that they are the same.