Ayan Nath

UID: 605910299

Obstacles:

For me, the biggest obstacle was writing the Score constructor. For me the hardest part was making letters turn into maybe if there were duplicates of those letters. For example, if the constructor was Score(level, spell), I was initially unable to correctly mark the second e as wrong. The similar situation occurred when in Score(daddy, added). These were both situations that I worked with a TA with, and I finally came to a solution where I made a copy of the correct word solution to a string, and edited the string as I progressed so there would be no overlap.

Test Cases (some inspired during discussion with TAs and looking at the suggested main)

//Testing the Piece methods

Piece b;

assert( b.getLetter() == NOTVALID ); //tests getLetter()

b = Piece( "A" ); //tests constructor

assert( b.getLetter() == B ); //tests getLetter()

assert( b.getLetterAsString() == "B" ); //tests getLetterAsString()

Move m;

b = m.getPiece( 0 ); //tests getPiece()

assert( b.getLetter() == NOTVALID ); //tests getLetter()

m.setPieces( "hello" ); //tests setPieces()

b= m.getPiece( 2 ); //tests getPiece()

assert( b.getLetter() == L ); //tests getLetter()

m.setPiece( 2, 'W' ); //tests setPiece

b = m.getPiece( 2 ); //tests getPiece()

assert( b.getLetter() == W ); //tests getLetter()

//Testing the Score methods

Score s;

assert( s.isExactMatch() == false ); //tests isExactMatch()

assert( s.getAnswer( 2 ) == WRONG ); //tests getAnswer()

m.setPieces( "hello" ); // tests setPieces()

Move theAnswer;

theAnswer.setPieces( "stank" ); //tests setPieces()

s = Score( m, theAnswer );//tests constructor

assert( s.isExactMatch() == false ); //tests isExactMatch()

assert( s.to\_string() == "\_\_\_\_\_" ); //tests to\_string()

theAnswer.setPieces( "hello" );//tests setPieces()

s = Score( m, theAnswer ); //tests constructor

assert( s.isExactMatch() == true ); //tests isExactMatch()

assert( s.to\_string() == "RRRRR" ); //tests to\_string()

//Testing words that have duplicate letters as well as the other methods from the project

//Also tests the board class methods

Board b;

assert( b.getCurrentRound() == 0 ); //tests getCurrentRound()

m.setPieces( "daddy" ); //tests setPieces()

theAnswer.setPieces( "added" );

s = Score( m, theAnswer ); //tests Score constructor

b.endRound( m, s ); //tests endRound()

assert( b.getCurrentRound() == 1 ); //tests getCurrentRound()

assert( b.getMoveForRound( 0 ).to\_string() == "DADDY" ); //tests getMoveForRound()

assert( b.getScoreForRound( 0 ).to\_string() == "MMRM\_" );//tests getScoreForRound()

//Testing the Wordle class methods

Wordle game( "added" );

assert( game.answer() == "added" );//tests answer()

assert( game.gameIsOver() == false );//tests gameIsOver()

m = game.play( "daddy" ); //tests play()

s = game.endRound( m );//tests endRound()

assert( s.to\_string() == " MMRM\_" ); //tests to\_string() and if game was ended collectly