Report

Jingjing Nie 304567417

1. Some of the notable obstacles I overcame were to figure out the relationships between each member of the class, and the logics of how this game should perform. Also, to properly include each member within one another is one of the challenges I faced.
2. Since this project involves random numbers, we need to “cheat” --- forcing the players to roll certain numbers in order to test the code.

For instance, to simulate a game with human winning as the final result (since human has won 4 rounds and computer has won 2 rounds):

Player p;

assert( p.getScore() == 0 );

p.setRound( 1 );

assert( p.getScore() == 0 );

assert( p.roll( 2 ) == 2 );

assert( p.getScore() == 0 );

assert( p.roll( 3 ) == 3 );

assert( p.getScore() == 0 );

assert( p.roll( 1 ) == 1);

assert( p.getScore() == 1 );

p.setRound( 3 );

assert( p.getScore() == 0 );

assert( p.roll( 3 ) == 3 );

assert( p.getScore() == 1 );

p.setRound( 5 );

assert( p.getScore() == 0 );

assert( p.roll( 5 ) == 5 );

assert( p.getScore() == 1 );

assert( p.roll( 5 ) == 5 );

assert( p.getScore() == 2);

assert( p.roll( 5 ) == 5 );

assert( p.getScore() == 3);

Bunco b;

b.setRound( 1 );

assert( b.determineRoundOutcome() == Bunco::NOTDECIDED );

b.computerPlay( 5 );

b.humanPlay( 3 );

assert( b.determineRoundOutcome() == Bunco::NOTDECIDED );

assert( !b.gameIsOver());

b.computerPlay( 1 );

b.humanPlay( 1 );

assert( b.determineRoundOutcome() == Bunco::NOTDECIDED );

assert( !b.gameIsOver());

b.computerPlay( 1 );

b.humanPlay( 2 );

assert( b.determineRoundOutcome() == Bunco::COMPUTERWON );

assert( !b.gameIsOver());

b.setRound( 2 );

assert( b.determineRoundOutcome() == Bunco::NOTDECIDED );

b.computerPlay( 1 );

b.humanPlay( 2 );

assert( b.determineRoundOutcome() == Bunco::HUMANWON );

assert( !b.gameIsOver());

b.setRound( 3 );

assert( b.determineRoundOutcome() == Bunco::NOTDECIDED );

b.computerPlay( 5 );

b.humanPlay( 1 );

assert( b.determineRoundOutcome() == Bunco::NOTDECIDED );

assert( !b.gameIsOver());

b.computerPlay( 4 );

b.humanPlay( 3 );

assert( b.determineRoundOutcome() == Bunco::HUMANWON );

assert( !b.gameIsOver());

b.setRound( 4 );

assert( b.determineRoundOutcome() == Bunco::NOTDECIDED );

b.computerPlay( 5 );

b.humanPlay( 3 );

assert( b.determineRoundOutcome() == Bunco::NOTDECIDED );

assert( !b.gameIsOver());

b.computerPlay( 4 );

b.humanPlay( 1 );

assert( b.determineRoundOutcome() == Bunco::COMPUTERWON );

assert( !b.gameIsOver());

b.setRound( 5 );

assert( b.determineRoundOutcome() == Bunco::NOTDECIDED );

b.computerPlay( 6 );

b.humanPlay( 3 );

assert( b.determineRoundOutcome() == Bunco::NOTDECIDED );

assert( !b.gameIsOver());

b.computerPlay( 5 );

b.humanPlay( 5 );

assert( b.determineRoundOutcome() == Bunco::NOTDECIDED );

assert( !b.gameIsOver());

b.computerPlay( 1 );

b.humanPlay( 5 );

assert( b.determineRoundOutcome() == Bunco::HUMANWON );

assert( !b.gameIsOver());

b.setRound( 6 );

assert( b.determineRoundOutcome() == Bunco::NOTDECIDED );

b.computerPlay( 5 );

b.humanPlay( 3 );

assert( b.determineRoundOutcome() == Bunco::NOTDECIDED );

assert( !b.gameIsOver());

b.computerPlay( 1 );

b.humanPlay( 1 );

assert( b.determineRoundOutcome() == Bunco::NOTDECIDED );

assert (!b.gameIsOver());

b.computerPlay( 1 );

b.humanPlay( 6 );

assert( b.determineRoundOutcome() == Bunco::HUMANWON );

assert( b.gameIsOver());

assert( b.determineGameOutcome() == Bunco::HUMANWONGAME);