Project 7 Report

1. Some of the problems I faced were with general syntax (in the beginning) for the class functions. My biggest issue was later, where I realized my player.ccp was written incorrectly and wasn’t resetting scores, which took me a long time to figure out (before reading the FAQ I had thought I had to fix it in Board, not Player). After figuring out that I was just not correctly allocating values to mScore in my Player::roll(), it wasn’t too bad fixing it. I also had an issue with the final assert, which I’m not entirely sure how I fixed because I just remember looking at the code for issues and then it just magically worked. I must have changed something small while I was moving things around.
2. I used first the test code available and then tweaked it a little just to make sure multiple scenarios worked. I also played the game myself.

#include <iostream>

#include <string>

#include <cassert>

#include "Die.h"

#include "Board.h"

#include "Player.h"

#include "Pig.h"

int main()

{

using namespace std;

using namespace cs31;

Die d;

Player p;

Board board, b1;

Pig game;

// test code for Die

for (int i = 1; i <= 100; i++)

{

d.roll();

int value = d.getValue();

assert(value >= 1 && value <= 6);

}

// test code for Player

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

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

assert(p.roll(4) == 4);

assert(p.getScore() == 4);

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

assert(p.getScore() == 9);

p.endTurn();

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

assert(p.getTotal() == 9);

assert(p.roll(4) == 4);

assert(p.getScore() == 4);

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

assert(p.getScore() == 7);

assert(p.roll(6) == 6);

assert(p.getScore() == 13);

p.endTurn();

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

assert(p.getTotal() == 22);

assert(p.roll(4) == 4);

assert(p.getScore() == 4);

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

assert(p.getScore() == 9);

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

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

assert(p.getTotal() == 22);

// test code for Board

assert(board.getComputerTotal() == 0);

assert(board.getHumanTotal() == 0);

assert(board.getRolledValue() == 0);

assert(board.getComputerScore() == 0);

assert(board.getHumanScore() == 0);

assert(board.isHumanWinner() == false);

assert(board.isHumanTurn() == true);

assert(board.isGameOver() == false);

board.setComputerTurn();

board.setComputerTotal(2);

board.setComputerScore(10);

board.setHumanTurn();

board.setHumanScore(3);

board.setHumanTotal(14);

board.setRolledValue(7);

assert(board.getComputerTotal() == 2);

assert(board.getHumanTotal() == 14);

assert(board.getRolledValue() == 7);

assert(board.getComputerScore() == 10);

assert(board.getHumanScore() == 3);

assert(board.isHumanWinner() == false);

assert(board.isHumanTurn() == true);

assert(board.isGameOver() == false);

// test code for b1 - board's should accept values passed to them

b1.setComputerTurn();

b1.setComputerTotal(22);

b1.setComputerScore(23);

b1.setHumanTurn();

// no changes allowed if it isn't your turn...

b1.setComputerTotal(50);

b1.setComputerScore(50);

assert(b1.getComputerTotal() == 22);

assert(b1.getComputerScore() == 23);

b1.setHumanScore(25);

b1.setHumanTotal(26);

b1.setRolledValue(28);

b1.setRolledValue(250);

b1.setGameOver(true);

b1.markComputerAsWinner();

assert(b1.getComputerTotal() == 22);

assert(b1.getHumanTotal() == 26);

assert(b1.getRolledValue() == 250);

assert(b1.getComputerScore() == 23);

assert(b1.getHumanScore() == 25);

assert(b1.isHumanWinner() == false);

assert(b1.isHumanTurn() == true);

assert(b1.isGameOver() == true);

// no board changes allowed if the game is over...

b1.setComputerTurn();

b1.setComputerTotal(50);

b1.setComputerScore(50);

b1.setRolledValue(50);

assert(b1.getComputerTotal() == 22);

assert(b1.getComputerScore() == 23);

assert(b1.getRolledValue() == 250);

// test code for Game, just changed from original to test Computer Win

assert(game.isGameOver() == false);

assert(game.determineGameOutcome() == Pig::GAMEOUTCOME::GAMENOTOVER);

game.humanPlay(5);

game.humanEndTurn();

game.computerPlay(30);

game.computerEndTurn();

assert(game.isGameOver() == true);

assert(game.determineGameOutcome() == Pig::GAMEOUTCOME::COMPUTERWONGAME);

cout << "all tests passed!" << endl;

return 0;

}