Project 2 (Due: Sunday, 3 April 2016, 11:59 pm)

For this project you will be writing functions for an interactive Farkle Game class. Specification:

1. A starter set of files can be found in the folder Project2 inside of our course pickup folder. The file Game.h is listed below. Your assignment is to implement the functions (in Game.cpp) that have not been completed.

2. Start with the output (<<) functions. The output function for the Die struct should output the information about a single Die; it will be called by the output function for the Game class. The output function for the Game class should display the information about the current state of the game.

3. Complete the Instructions function. This function will be called when the game starts and if the player enters ‘i’. The instructions should clearly explain how the game is played.

4. Implement the Roll function. This function should “randomly” choose the number of pips for each of the ‘numToRoll’ dice and insert them in the ‘rolledDice’ vector.

5. Complete the Message function.

6. Implement the SaveSelected function. This function should copy each of the dice marked as selected into the ‘selectedDice’ vector and each of the non-selected dice into the ‘remainingDice’ vector.

7. Implement the SaveScore function.
   a. If the value of ‘whoseTurn’ is HUMAN, the ‘turnScore’ should be added to the ‘humanScore’. If the value is AIA, the ‘turnScore’ should be added to the ‘aiaScore’.
   b. This function should determine if the current player is on the board.
   c. This function should determine if the next player’s turn is their last turn.
   d. This function should determine whether or not there is a winner.

8. Once you have completed all of the above functions, implement the AIAPlayer function. This function should implement your strategy for playing the game in a “winning manner”. This function may use any or all of the previously implemented functions as needed.

9. Fully document all of your files and functions. Modify the makefile by replacing all of the occurrences of “yln” with your last name so that you can submit your project by entering make submit at the command line.

Date Due: Sunday, 3 April 2016, 11:59 pm.
#ifndef GAME_H
#define GAME_H

#include <iostream>
#include <vector>
using namespace std;

#define MAX_DICE 6
#define POINTS_TO_WIN 5000
#define MIN_FIRST_SCORE 500

enum state_types { GO, HUMANWINS, AIAWINS, QUITTER }
enum player_type { NONE, HUMAN, AIA };

class Game
{
    public:
    Game (const string & T);
    ~Game ();
    void Init ();
    void Instructions (ostream & outs);
    void StartTurn ();
    friend ostream & operator << (ostream & outs, const Game & G);
    bool Enter (char selection, ostream & outs);
    bool TurnDone ();
    int AIAPlayer (ostream & outs);
    bool Done ();
    void Message (ostream & outs);

    struct Die
    {
        Die ();
        friend ostream & operator << (ostream & outs, const Die & D);
        int pips;
        bool selected;
        bool used;
    };
    private:
    void Roll ();
    int CalculateScore (vector <Die> & dice);
    void SaveSelected ();
    void SaveScore ();
    string title;
    vector <Die> rolledDice;
    vector <Die> remainingDice;
    vector <Die> selectedDice;
    vector <Die> savedDice;
    state_types gameState;
    player_type whoseTurn;
    bool turnDone;
    int numToRoll;
    int rolledScore;
    int subTotal;
    int turnTotal;
    bool humanOnBoard;
    bool aiaOnBoard;
    int humanScore;
    int aiaScore;
    bool lastTurn;
};

#ifndef GAME_H
#endif