Project 1

<Rock Paper Scissors>

CIS-5 44187

Name: Seth Tyler

Date: 4/17/17

Introduction

Title: Rock Paper Scissors

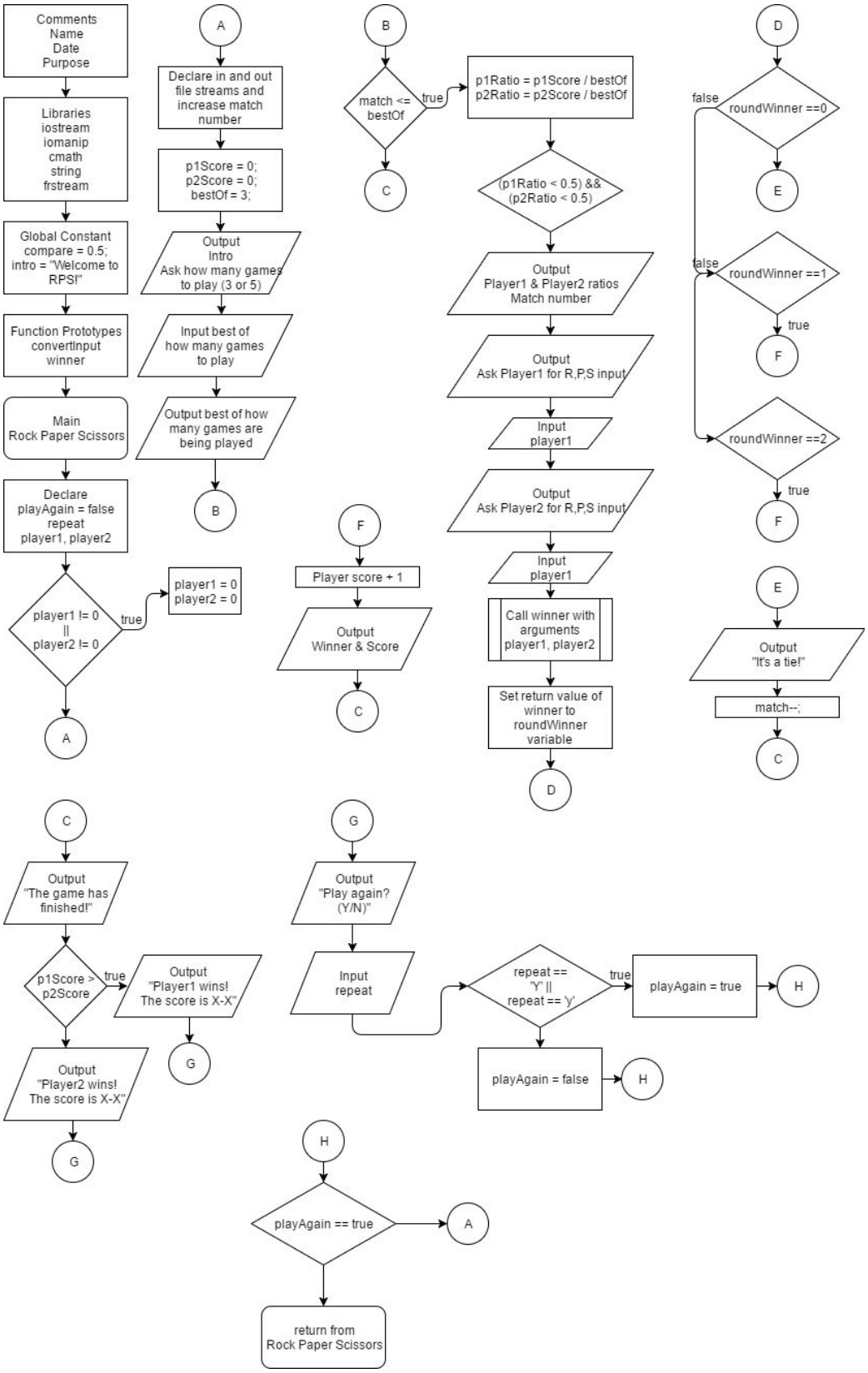
This game is based on the classic physical game of Rock, Paper, Scissors. In person, the game is played as both players pound a fist on the other hand 3 times as they say ‘Rock, Paper, Scissors’, then ‘shoot’, to which they both reveal a hand sign simultaneously depicting a rock, paper, or scissors. Rock breaks scissors, paper covers rock, and scissors cut paper. The game is typically played in a best 2 out of 3 format, to which the player who wins twice wins the set.

Summary

Project Size: 200 lines (including blank lines)
Number of Variables: About 15-20
Number of functions: 3

This project aims to include all sections covered from chapter 1 - 5 in the book. Total production time probably added to around 20 hours of work, including flow charts and the write up.

Flow Chart


Comments
Name
Date
Purpose
Libraries
iostream
iomanip
cmath
string
frstream
Global Constant
compare = 0.5;
intro = "Welcome to RPS!"
Function Prototypes
convertInput
winner
Main
Rock Paper Scissors
Declare
playAgain = false
repeat
player1, player2
player1 != 0
||
player2 != 0
true
player1 = 0
player2 = 0
A
A
Declare in and out file streams and increase match number
p1Score = 0;
p2Score = 0;
bestOf = 3;
Output
Intro
Ask how many games to play (3 or 5)
Input best of how many games to play
Output best of how many games are being played
B
B
match <= bestOf
true
C
p1Ratio = p1Score / bestOf
p2Ratio = p2Score / bestOf
(p1Ratio < 0.5) &&
(p2Ratio < 0.5)
Output
Player1 & Player2 ratios
Match number
Output
Ask Player1 for R,P,S input
Input
player1
Output
Ask Player2 for R,P,S input
Input
player1
Call winner with arguments
player1, player2
Set return value of winner to roundWinner variable
D
F
Player score + 1
Output
Winner & Score
C
D
false
roundWinner ==0
E
false
roundWinner ==1
true
F
roundWinner ==2
true
F
E
Output
"It's a tie!"
match--;
C
C
Output
"The game has finished!"
p1Score > p2Score
true
Output
"Player1 wins!
The score is X-X"
G
Output
"Player2 wins!
The score is X-X"
G
G
Output
"Play again?
(Y/N)"
Input
repeat
repeat ==
'Y' ||
repeat == 'y'
true
playAgain = true
H
playAgain = false
H
H
playAgain == true
A
return from
Rock Paper Scissors

Pseudo Code

```
/*
 * File:   main.cpp
 * Author: Seth Tyler
 * Created on April 16, 2017, 6:06 PM
 * Purpose:  Rock paper scissors
 */

//System Libraries

// Input/Output Library
// Format Library
// Math library
// String library
// File stream library

//Namespace std of system libraries

//User Libraries

//Global Constants
//Float comparison
//Intro string

//Function Prototypes
// Convert input function, changes char to int and outputs accordingly
// Determine winner function, determines which player won via rock, paper or
scissors

// Main -> Executable code begins here

    // Declare variables
    // Such as playAgain variable and player inputs

    // While loop that resets scores to 0

    // Play the game by looping until playAgain is false

        // Read the number of matches played from a text file, increment it and
output the number of matches played.

        // Declare variables
        // Player 1's score, Player2's score, best of x games

        // Introduction
        // Output a welcome to the game and ask how many games to play (best of
3 or 5)
```

```
        // Determine how many games will be played
        // Read user input and set bestOf to number of games according to input

        // Output a confirmation of how many games are being played

        // Loop until matches have finished
            // Check if a player has won

            // Declare ratio variables. If one becomes 0.5 or greater, that
player wins.

            // Check if a player has the best of 3 or best of 5

               // Output the ratios

                // Output the match numbers

                // Inputs

                // Tell player1 to input, receive input. Tell player2 to input.
Receive input.



                // Determine the winner
                // Call the winner function and write to roundWinner variables

                // Output the winner
                // Use roundWinner variable to determine the winner. If 0, it's
a tie. If 1, player 1 wins. If 2, player 2 wins.
                // Increase the score accordingly or decrease match number if
it's a tie.


        // Conclude
        // Output who the winner is

        // Ask the user if they want to play again.
        // Read input

        // If input is Y/y, set playAgain to true and repeat the game.

// Close files and exit stage right!
```

Reference

Textbook (Savitch & Gaddis)

Variables

Type	Name	Description
Integer	convertInput	A function that uses a switch to convert the user's input (char) and returns 1, 2, 3 for rock, paper, and scissors respectively
	conversion	An integer variable that converts the char argument of the function convertInput to an integer
	winner	A function that uses if-else-if statements to determine who the winner of the round is based on two inputs. Returns 0 for tie, 1 for player1, 2 for player2.
	player1	The first argument of function "winner" for player1's input
	player2	The second argument of function "winner" for player1's input
	main	Main
	matches	The number of matches played by the user
	bestOf	The number of games to be played, or the 'best of X' games.
	match	The current iteration of the loop that cycles through each of the rounds.

	roundWinner	An integer that stores the result of the returned value by the winner function with arguments player1 and player2.
String	intro	A string that welcomes the user.
Bool	playAgain	A variable that determines if the do-while loop should continue running.
Char	repeat	A variable that stores a char inputted by the user. Input ‘Y’ or ‘y’ will tell playAgain to be true and continue the loop.
	input	The char argument of function convertInput
	player1	User input for player1
	player2	User input for player2
Float	compare	The constant comparison value that the program uses to determine if a player has won
	p1Score	The current amount of wins player1 has.
	p2Score	The current amount of wins player2 has.
	p1Ratio	The amount of wins player 1 has divided by the bestOf value.
	p2Ratio	The amount of wins player 2 has divided by the bestOf value.

Program

```
/*
 * File:   main.cpp
 * Author: Seth Tyler
 * Created on April 16, 2017, 6:06 PM
 * Purpose:  Rock paper scissors
 */

//System Libraries
#include <iostream>  //Input - Output Library
#include <iomanip>
#include <cmath>
#include <string>
#include <fstream>

using namespace std; //Name-space under which system libraries exist

//User Libraries

//Global Constants
const float compare = 0.5; // the number that is used to determine if the
matches are finished
const string intro = "Welcome to Rock, Paper, Scissors!\n";

//Function Prototypes

int convertInput(char input) {

    int conversion = static_cast<int>(input);

    switch (conversion) {
        case 82: // if the input is R
        case 114: // if the input is r
            return 1;
            break;
        case 80: // if the input is S
        case 112: // if the input is s
            return 2;
            break;
        case 83: // if the input is P
        case 115: // if the input is p
            return 3;
            break;
        default: // if the user input is anything else
            return 1; // default to rock
            break;
    }
```

```
}

int winner(int player1, int player2) {
    // Calculate the winner
    // 1 = rock, 2 = paper, 3 = scissors

    // Convert (char) R, P, S to (integer) 1, 2, 3 respectively
    player1 = convertInput(player1);
    player2 = convertInput(player2);

    if (player1 == 1) {
        if (player2 == 1) { // If player2 chooses rock and player1 chooses rock
            return 0;
        } else if (player2 == 2) { // If player2 chooses paper and player1
chooses rock
            return 2;
        } else if (player2 == 3) { // If player2 chooses scissors and player1
chooses rock
            return 1;
        }
    } else if (player1 == 2) {
        if (player2 == 1) { // If player2 chooses rock and player1 chooses
paper
            return 1;
        } else if (player2 == 2) { // If player2 chooses paper and player1
chooses paper
            return 0;
        } else if (player2 == 3) { // If player2 chooses scissors and player1
chooses paper
            return 2;
        }
    } else if (player1 == 3) {
        if (player2 == 1) { // If player2 chooses rock and player1 chooses
scissors
            return 2;
        } else if (player2 == 2) { // If player2 chooses paper and player1
chooses scissors
            return 1;
        } else if (player2 == 3) { // If player2 chooses scissors and player1
chooses scissors
            return 0;
        }
    }
}

//Execution begins here
int main(int argc, char** argv) {
```

```
    // Declare variables

    bool playAgain = false; // if true, the loop will continue
    char repeat; // user input for settings playAgain to true or false
    char player1, player2; // user inputs for rock, paper or scissors

    while (player1 != 0 || player2 != 0) {
        player1 = 0;
        player2 = 0;
    }

    // Play the game
    do {

        // Read the number of matches played from a text file, increment it and
output the number of matches played.
        int matches;
        {
            std::ifstream in( "matchesPlayed.txt" );
            in >> matches;
         }
         matches++;
         {
            std::ofstream out( "matchesPlayed.txt" );
            out << matches;
         }

         cout << (matches - 1) << " matches have been played.\n";

        // Declare variables
        float p1Score = 0; // player1's score
        float p2Score = 0; // player2's score
        int bestOf = 3; // best out of how many matches will be played (3 or 5)

        // Introduction
        cout << intro;
        cout << "Rock beats scissors, paper beats rock, and scissors beats
paper.\n";
        cout << "Best of how many games will be played? (3 or 5): ";

        // Determine how many games will be played
        cin >> bestOf; // user inputs if they want a best of 3 or best of 5
        bestOf = (bestOf == 5 ? 5 : 3); // if the input is 5 then it is a best
of 5, otherwise it is a best of 3

        cout << endl << "A best of " << bestOf << " match! Let's begin.\n";
```

```
        // Loop until matches have finished
        for (int match=1; match <= bestOf; match = match + 1) { // create a
loop for the number of matches
            // Check if a player has won

            // if the ratio becomes > 0.5, they win.
            float p1Ratio = p1Score / bestOf;
            float p2Ratio = p2Score / bestOf;

            if ((p1Ratio < 0.5) && (p2Ratio < 0.5)) { // check if a player has
the best of 3 or best of 5

                cout << setprecision(2) << fixed;
                cout << "The ratio for player 1 is " << p1Ratio << endl
                        << "The ratio for player 2 is " << p2Ratio << endl;
                cout << setprecision(0) << fixed;

                cout << "MATCH " << match;

                // Inputs

                cout << endl << "Player 1, enter R for Rock, P for Paper, and S
for Scissors: ";
                cin >> player1;
                cout << endl << "Player 2, enter R for Rock, P for Paper, and S
for Scissors: ";
                cin >> player2;
                cout << endl << "Let's determine the results...\n";



                // Determine the winner
                int roundWinner = winner(player1, player2);

                // Output the winner
                switch (roundWinner) {
                    case 0:
                        match--;
                        cout << "It was a tie!\n";
                        break;
                    case 1:
                        p1Score++;
                        cout << "Player 1 wins! The score is " << abs(p1Score)
<< "-" << abs(p2Score) << endl;
                        break;
                    case 2:
                        p2Score++;
```

```
                        cout << "Player 2 wins! The score is " << abs(p2Score)
<< "-" << abs(p1Score) << endl;
                        break;
                }
            } else {
                break;
            }
        }

        // Conclude
        cout << "The game has finished!\n";
        if (p1Score > p2Score ) {
            cout << "Player 1 takes it with a score of " << p1Score << "-" <<
p2Score << "!\n";
        } else {
            cout << "Player 2 takes it with a score of " << p2Score << "-" <<
p1Score << "!\n";
        }

        // Ask the user if they want to play again.
        cout << "Play again? (Y/N)\n";
        cin >> repeat;
        if (repeat == 'Y' || repeat == 'y') {
            playAgain = true;
        } else {
            playAgain = false;
        }
    } while (playAgain == true);

    return 0;
}
```

Complexities Explained

Ratio

The ratio is a value that the program compares to determine the winner. The for loop will continue to run if and only if both players' wins each divided by the bestOf value are less than 0.5. If the value is greater, (the value cannot be exactly 0.5 since the program is using integers and the denominator is an odd number) the for loop will break and this will tell the program that a player has beaten the other. It will then take the highest ratio to determine who won the set.