**Project 1**

**Game Title:**

Snakes and Ladders

**Project Includes:**

Versions 1,2,& 3 of My Development

**Course:**

CSC 17A Fall 2017

**Section:**

47466

**Date Due:**

November 5, 2017

**By:**

Javier Ventura

**Introduction:**

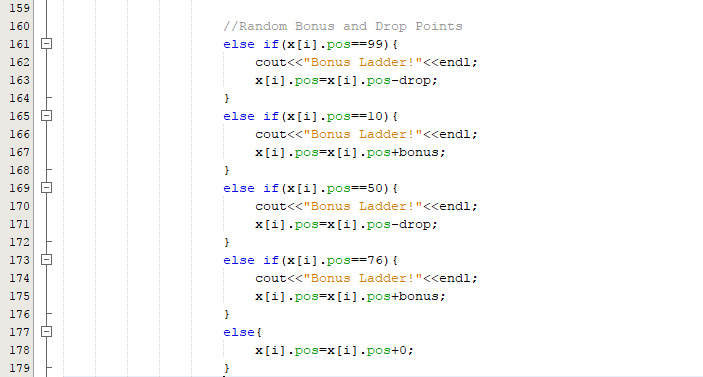
For Project 1, as a game I wanted to do in C++, I chose the popular board game of, Snakes and Ladders. This was a favorite of mine as a child. The game is not extremely complex compared to other games in terms of rules but the level of competition with family and friends often turned really intense because of the fun and excitement the game can bring. I wanted to be able to do a small interpretation of the game with what I know so far in C++ as well as use as much of the concepts from class as possible.

**Summary:**

The Final code in version 3 comes out to be 229 lines of code in the main cpp file and 22 lines in the header file, giving a total of 251 lines. I declared 14 variables in the main not including the 3 variable declarations inside a “for” loop. I also had 4 variables inside my structure for the players in the header file. I used a do-while loop to keep the game running and a switch statement. The problem was that the game just ran without the need of the player to do anything when rolling. The switch statement helped fix that with prompting the user to enter when they wanted to roll the dice. I used arrays, structures, array of structures, c-strings, functions with array of structures, and binary files. It was however, with the same structure, the array is dynamically allocated because the size is not specified until the user enters how many players will participate. After dynamically allocating memory for the array, I also pass it into many for loops and in a function I use to reset points. C-Strings are used for the name of the game and with the names and wins that I read to a binary file. Besides the constructs from class I also used a good amount of if-else if-else statements to maneuver around my code as well as set traps and bonuses to increase the amount of logic my game has with random number seeds using time. Putting concepts from CSC 5 and this course took a good amount of planning and much more to put in practice.

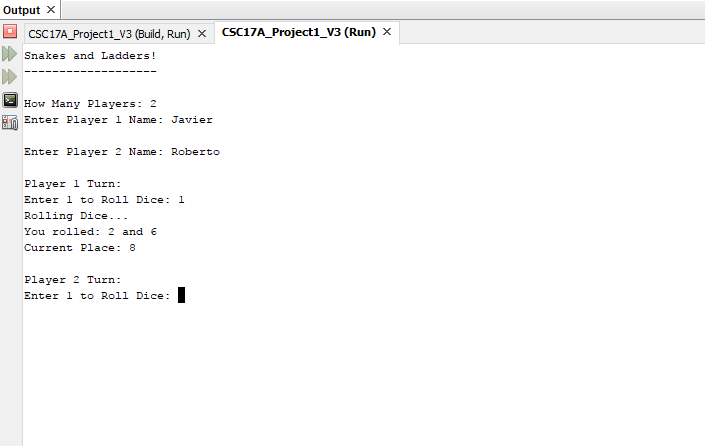
**Description:**

With this project I spanned three versions worth of time. However, the first version is not close to the game. My intention in version one was to get base of what the game is trying to do, roll dice and keep track of where the player is. I know in the game development industry they call this a “minimum viable product” or the basic idea of a game, something to test and ensure to the developers that they have an idea worth continuing. Version one is the same to me, before I continued on with the project using this idea, I wanted to make sure the base on its own would be fun and then adding to it would only make it even better. Version two is not much more as I only added a few elements and incorporating binary files. This was done because as I mentioned the rest of the constructs from class were mostly added all together as they relate closer together. In version 3 I add the array that allows multiple characters to play. I add the ladders that move forward and snakes that move back. As an extra twist of mine, I add random ladders and snakes, I label these variables as bonus and drop. These integer type number have a range of 1 to 10. If landed on the specified numbers you will move up or down any number in that range worth of spots. This keeps the game exciting and making sure someone cannot memorize what is going to happen.



If the player lands in any of those spots then the new spot is current spot plus or minus the bonus or drop. For Loops are found on lines 61,215,236. Binary Files on line 216. Do while loop starts at 60 and ends at 212. If statements span lines 74 to 178. Switch statement starts at line 66 and ends at line 196. Function that gets a structure passed in is on line 225. All major variables are on lines 28-41. Dynamic Memory Allocation on line 51. Random Number seed on line 25. The libraries are from lines 10 to 14 and the rest are in the header file.

To test the code I played with my brother as shown.



**PseudoCode:**

Create Header File

Initialize structure Player

Declare and initialize variables in structure

Include libraries in main cpp file

Include header file

Declare function prototype to pass array of structure and size

Set random number seed

Declare and initialize variables for dice, array and fstream objects

Prompt for number of players

Use do while loop to keep game going

Use for loop to go alternate players

Use switch statement to roll dice

Use if statements to incorporate snakes and ladders

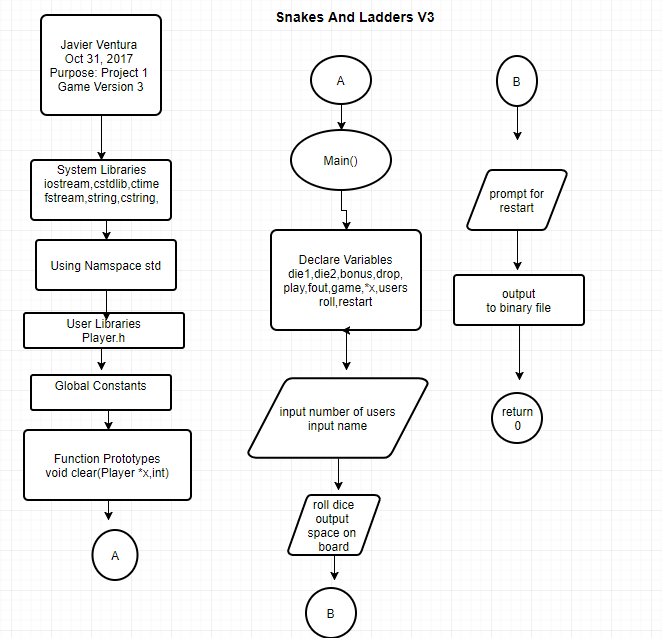
Use if statement to track when there is a winner

Use if statement to prompt for restart and call function to reset

Use for loop to output results to binary file

end program

**FlowChart:**

****

**Source Code (Main):**

/\*

\* File: main.cpp

\* Author: Javier Ventura

\* Purpose: CSC 17A Project 1

\* Includes Binary File Concepts, V3 will add turn-based play to use

\* structures with arrays and functions as well as memory allocation,C Strings

\* Created on October 31, 2017, 6:10 PM

\*/

#include <cstdlib> // C Standard Library

#include <iostream> // Input and Output Library

#include <fstream> // Input and Output to Files

#include <string.h> // String Library

#include <cstring> // C String Library

#include "Player.h" // Include Player Header File with Player Structure

using namespace std;

//Function Prototypes

void clear(Player \*x,int);

int main(int argc, char\*\* argv) {

//Random Number Seed

srand(static\_cast<unsigned int>(time(0)));

//Declare Variables

int points=0; // Points are added up by dice rolled,max of 12 per roll

int pos=0; // Position on Board out of 100

string name; // Name of Player

int die1=rand()%6+1; // 1 of two Die range is from 1 to 6

int die2=rand()%6+1; // 1 of two Die range is from 1 to 6

int bonus=rand()%10+1; // Possible bonus points to move forward 1 to 10

int drop=rand()%10+1; // Possible random drop from 1 to 10

bool play=true; // As long as play is true then game continues

fstream fout; // Stream Object to write to binary file

char const \*game{"Snakes and Ladders!"}; //Name of the game in C string

Player \*x; // Pointer to array for structure

int users; // Number of Users

int roll; // Used for switch statement in turns

char restart; // Used to get response when game ends

cout<<game<<endl;

cout<<"-------------------"<<endl;

cout<<endl;

cout<<"How Many Players: ";

cin>>users;

x=new Player[users];

for(int i=0;i<users;i++){

cout<<"Enter Player "<<i+1<<" Name: ";

cin>>x[i].name;

cout<<endl;

}

do{

for(int i=0;i<users;i++){

cout<<"Player "<<i+1<<" Turn: "<<endl;

cout<<"Enter 1 to Roll Dice: ";

cin>>roll;

switch(roll){

case 1:{

cout<<"Rolling Dice..."<<endl;

cout<<"You rolled: "<<die1<<" and "<<die2<<endl;

x[i].points=die1+die2;

x[i].pos=x[i].pos+x[i].points;

//Set Traps and Ladder Jumps

if(x[i].pos==4){

cout<<"Ladder!"<<endl;

x[i].pos=14;

}

else if(x[i].pos==9){

cout<<"Ladder!"<<endl;

x[i].pos=31;

}

else if(x[i].pos==13){

cout<<"Ladder!"<<endl;

x[i].pos=74;

}

else if(x[i].pos==66){

cout<<"Ladder!"<<endl;

x[i].pos=78;

}

else if(x[i].pos==18){

cout<<"Ladder!"<<endl;

x[i].pos=37;

}

else if(x[i].pos==28){

cout<<"Ladder!"<<endl;

x[i].pos=84;

}

else if(x[i].pos==21){

cout<<"Ladder!"<<endl;

x[i].pos=42;

}

else if(x[i].pos==51){

cout<<"Ladder!"<<endl;

x[i].pos=67;

}

else if(x[i].pos==71){

cout<<"Ladder!"<<endl;

x[i].pos=91;

}

else if(x[i].pos==80){

cout<<"Ladder!"<<endl;

x[i].pos=97;

}

else if(x[i].pos==32){

cout<<"Ladder!"<<endl;

x[i].pos=49;

}

else if(x[i].pos==17){

cout<<"Snake!"<<endl;

x[i].pos=7;

}

else if(x[i].pos==64){

cout<<"Snake!"<<endl;

x[i].pos=60;

}

else if(x[i].pos==62){

cout<<"Snake!"<<endl;

x[i].pos=19;

}

else if(x[i].pos==87){

cout<<"Snake!"<<endl;

x[i].pos=24;

}

else if(x[i].pos==54){

cout<<"Snake!"<<endl;

x[i].pos=34;

}

else if(x[i].pos==98){

cout<<"Snake!"<<endl;

x[i].pos=79;

}

else if(x[i].pos==95){

cout<<"Snake!"<<endl;

x[i].pos=75;

}

else if(x[i].pos==93){

cout<<"Snake!"<<endl;

x[i].pos=53;

}

else if(x[i].pos==55){

cout<<"Snake!"<<endl;

x[i].pos=3;

}

else if(x[i].pos==29){

cout<<"Snake!"<<endl;

x[i].pos=6;

}

//Random Bonus and Drop Points

else if(x[i].pos==99){

cout<<"Bonus Ladder!"<<endl;

x[i].pos=x[i].pos-drop;

}

else if(x[i].pos==10){

cout<<"Bonus Ladder!"<<endl;

x[i].pos=x[i].pos+bonus;

}

else if(x[i].pos==50){

cout<<"Bonus Ladder!"<<endl;

x[i].pos=x[i].pos-drop;

}

else if(x[i].pos==76){

cout<<"Bonus Ladder!"<<endl;

x[i].pos=x[i].pos+bonus;

}

else{

x[i].pos=x[i].pos+0;

}

cout<<"Current Place: "<<x[i].pos<<endl;

cout<<endl;

if(x[i].pos>=100){

cout<<"You win!"<<endl;

cout<<endl;

x[i].wins++;

cout<<"Current wins for Player "<<i+1<<": "<<x[i].wins<<endl;

cout<<endl;

break;

}

break;

}

default:{

cout<<"You Must Enter 1 to Roll Dice"<<endl;

}

}

if(x[i].pos>=100){

cout<<"Restart and Play Again? Y for Yes and N for No "<<endl;

cin>>restart;

if(restart=='Y'){

clear(x,users);

cout<<endl;

}

else{

cout<<"Quitting Game... :("<<endl;

return 0;

}

}

}

}while(play=true);

//Writing points to a binary file

for(int i=0;i<users;i++){

fout.open("Stats.dat",ios::out|ios::binary);

fout.put(\*game);

fout.put(x[i].wins);

}

return 0;

}

//Functions with Structures

void clear(Player \*x,int n){

for(int i=0;i<n;i++){

x[i].pos=0;

}

}

**Source Code (Player.h):**

/\*

\* File: Player.h

\* Author: Javier Ventura

\* Purpose: CSC 17A Project 1 Structure Header File

\* Created on October 31, 2017, 6:19 PM

\*/

#ifndef PLAYER\_H

#define PLAYER\_H

#include <string>

using std::string;

struct Player{

int pos=0; //Position of player on the board

string name; //Name of Player

int points=0; //Points the player gets from dice

int wins=0; //The wins a player has in a session

};

#endif /\* PLAYER\_H \*/