**Project 2**

**Game Title:**

Snakes and Ladders

**Project Includes:**

Versions 4,5 & 6 of My Development

**Course:**

CSC 17A Fall 2017

**Section:**

47466

**Date Due:**

December 11, 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. In Project 2 I continued this idea but restructuring the game’s code to include the rest of the material from class this semester. Of course the biggest concept: classes.

**Summary:**

In Version 4 I convert version 3 to use classes instead of structures as was the task for project 2 overall. I added most concepts that come along with classes such as setting values, getting values, inheritance and overriding. The only concept left out was templates since I was again trying to get the base of the game down. I also left out binary files, although I had it in version 3 just to focus on the class portion of the project. In version 5, is where I included it once again. In version 5, I made the game a bit more continuous asking if there should be another game, if the answer is yes then the reset functions inside both player classes are called on to reset the positions on the board and then restarts the game. Binary and text files are added as well, this was done to check both components as well, before moving on. Improvements for the future could include better graphics, however possible. Creating a board or even importing some kind of graphics library would be best so it would not just run on the console.

**Checklist:**

Inheritance- line 16 player2.h

Classes- line 15 game.h

Polymorphism- line 16 player1.h

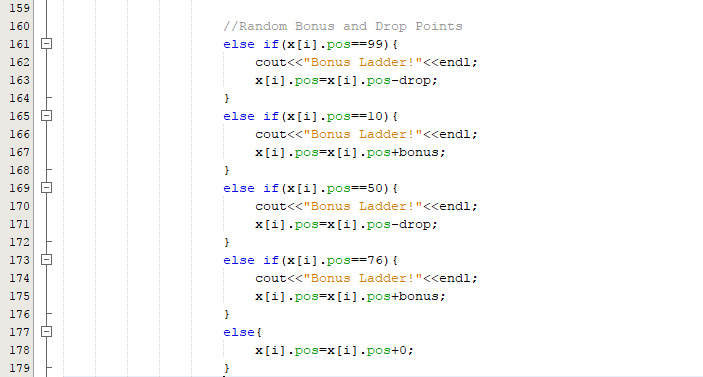
Templates- line 12 object.h

Binary Files- lines 143 -146 main.cpp

Text Files- lines 124-140 main.cpp

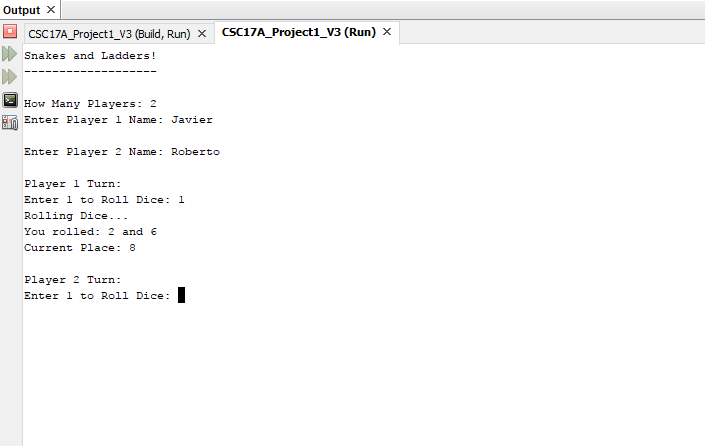
**Description:**

I kept in all versions the portion where 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. This is one of the advantages of having a digital version of the game. On a regular board that you play the ladders and snakes are set in place, so having some randomness to the game is an idea I had to keep the game fun, and give it more of a good restart appeal.



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 Game Class

Inherit Player 1 Class

Inherit Player 2 Class

Implement Classes

Include libraries in main cpp file

Include header files

Declare function prototype to pass array of structure and size

Set random number seed

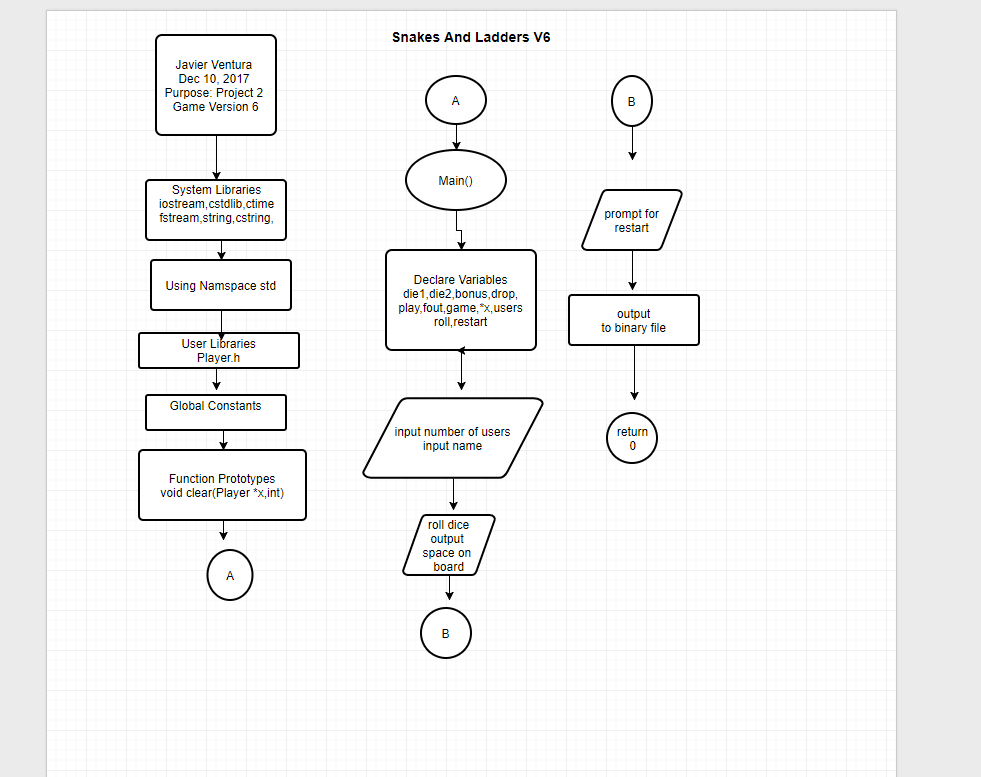
Declare and initialize variables for dice, array and fstream objects

Switch Statement for Menu

Output results to Files

End Program

**FlowChart:**

****

**Source Code (Main):**

/\*

\* File: main.cpp

\* Author: Javier Ventura

\* Purpose: CSC 17A Project 1

\* Includes: V6 is final version with all class concepts in classes,templates,files

\* inheritance and polymorphism

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

\*/

//Include Libraries, some also includes in respective header file implementations (.cpp files)

#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 <ctime> // Time Library

//Including Header Files

#include "game.h"

#include "player1.h"

#include "player2.h"

using namespace std;

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

//Declare Variables Here

// As long as play is true then game continues

bool play=true;

// Stream Object to write to binary file

fstream fout;

//Stream object to write to text file

ofstream data;

// Used to get response when game ends

char restart;

// Creating Object for Game Class

Game z;

// Creating Object for Player 1 Class

Player1 x;

//Creating Object for Player 2 Class

Player2 y;

//Choice for Menu

int choice;

//Game Starts Here

//Set username for Player 1

x.setName();

//Set username for player 2

y.setName();

//Print out the name of the game

z.printGame();

cout<<"1 - Play "<<endl;

cout<<"2 - Change Names "<<endl;

cout<<"3 - Quit Game "<<endl;

cout<<endl;

cout<<"Enter Choice: ";

cin>>choice;

cout<<endl;

switch(choice){

case 1:{

do{

cout<<endl;

x.setPos(); //Start Player 1 turn

//Set condition if player wins

if(x.getPos()>=100){

cout<<"Player 1 Wins!"<<endl;

cout<<endl;

x.setWins();

z.setGames();

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

cin>>restart;

if(restart=='Y'){

x.reset();

y.reset();

}

else{

play=false;

break;

}

}

//Start Player 2 turn

cout<<endl;

y.setPos();

//Set condition if player Wins

if(y.getPos()>=100){

cout<<"Player 2 Wins!"<<endl;

cout<<endl;

y.setWins();

z.setGames();

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

cin>>restart;

if(restart=='Y'){

x.reset();

y.reset();

}

else{

play=false;

break;

}

}

}while(play==true);

break;

}

case 2:{

//Set username for Player 1

x.setName();

//Set username for player 2

y.setName();

break;

}

case 3:{

//Writing to Text file

//Open File

data.open("stats.txt");

//Write to it Name of the Game

data<<"Snakes and Ladders"<<endl;

data<<"------------------"<<endl;

//Print Number of Overall Games

data<<"Number of Games: ";

data<<z.getGames()<<endl;

//Get Player 1 Name

data<<x.getName()<<endl;

//Get Number of wins for Player 1

data<<x.getWins()<<endl;

//Get Player 2 Name

data<<y.getName()<<endl;

//Get Number of wins for Player 2

data<<y.getWins()<<endl;

//Close File

data.close();

//Writing points to a binary file

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

fout.put(x.getWins());

fout.put(y.getWins());

break;

}

default:{

cout<<"Must Pick Valid Choice in Game Menu"<<endl;

cout<<endl;

}

}

return 0;

}

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

/\*

\* File: game.h

\* Author: Javier Ventura

\* Purpose: Specification of Game Class that will be passed down to other classes

\* Created on December 7, 2017, 4:28 PM

\*/

#ifndef GAME\_H

#define GAME\_H

#include <cstring>

class Game{

protected:

//Number of Games played in a session

int nGames;

//Name of the game in C string

char const \*game{"Snakes and Ladders!"};

public:

//Constructor

Game();

//Copy Constructor

Game(const Game &);

//Destructor Overriden by Inherited class destructors

~Game();

//Get Number of games overall

int getGames();

//Print Game for Intro

void printGame();

//Set Number of games played

void setGames();

};

#endif /\* GAME\_H \*/

**Source Code (Game.cpp):**

/\*

\* File: game.cpp

\* Author: Javier Ventura

\* Purpose: Implementation of the game class

\* Created on December 7, 2017, 5:03 PM

\*/

//Include Libraries

#include <iostream>

#include "game.h"

using namespace std;

//COnstructor

Game::Game(){

nGames=0;

}

//Destructor

Game::~Game(){

}

//Intro print game

void Game::printGame(){

cout<<game<<endl;

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

cout<<endl;

}

//Set Number of games played

void Game::setGames(){

nGames=nGames+1;

}

//Get number of games played for files

int Game::getGames(){

return nGames;

}

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

/\*

\* File: object.h

\* Author:Javier Ventura

\*

\* Created on December 10, 2017, 12:01 PM

\*/

#ifndef OBJECT\_H

#define OBJECT\_H

template <class T>

class Object{

private:

int num;

int \*x;

public:

Object();

void setX();

~Object();

};

#endif /\* OBJECT\_H \*/

**Source Code (object.cpp):**

/\*

\* File: object.cpp

\* Author: Javier Ventura

\*

\* Created on December 10, 2017, 12:03 PM

\*/

#include <iostream>

#include "object.h"

using namespace std;

template <class T>

Object<T>::Object(){

num=0;

}

template <class T>

void Object<T>::setX(){

cout<<"Enter Number of Players: ";

cin>>num;

x=new int[num];

}

template <class T>

Object<T>::~Object(){

delete x;

}

**Source Code (player1.h):**

/\*

\* File: player1.h

\* Author: Javier Ventura

\* Purpose: Specification for Player 1 object class inherited from Game class

\* Created on December 7, 2017, 4:27 PM

\*/

#ifndef PLAYER1\_H

#define PLAYER1\_H

#include "game.h"

#include <string>

using std::string;

class Player1:public Game{

private:

//Position on the board

int pos;

//Sum of both die rolled

int points;

//Username of player

string name;

//Number of wins in a session

int wins;

public:

//Constructor

Player1();

//Copy Constructor

Player1(const Player1 &obj);

//Set Position

void setPos();

//Get Position to check if they won

int getPos();

//Set Username for player

void setName();

//Set wins if position is greater than or equal to 100

void setWins();

//Get Wins for files

int getWins();

//Get Name for Files

string getName();

//Reset Position for new Game

void reset();

//Destructor

~Player1();

};

#endif /\* PLAYER1\_H \*/

**Source Code (player1.cpp):**

/\*

\* File: player1.cpp

\* Author: Javier Ventura

\* Purpose: Implementation of player 1 class

\* Created on December 7, 2017, 5:03 PM

\*/

#include <string>

#include <iostream>

#include <cstdlib>

#include <ctime>

#include "game.h"

#include "player1.h"

using namespace std;

//Constructor

Player1::Player1(){

wins=0;

pos=0;

points=0;

}

void Player1::setPos(){

//Setting Random Number Seed

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

// 1 of two Die range is from 1 to 6

int die1=rand()%6+1;

// 1 of two Die range is from 1 to 6

int die2=rand()%6+1;

// Press 1 to roll, interface for rolling

int roll;

// Possible bonus points to move forward 1 to 10

int bonus=rand()%10+1;

// Possible random drop from 1 to 10

int drop=rand()%10+1;

//Start Player 1 Turn

cout<<"Player 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;

points=die1+die2;

pos=pos+points;

//Set Traps and Ladder Jumps

if(pos==4){

cout<<"Ladder!"<<endl;

pos=14;

}

else if(pos==9){

cout<<"Ladder!"<<endl;

pos=31;

}

else if(pos==13){

cout<<"Ladder!"<<endl;

pos=74;

}

else if(pos==66){

cout<<"Ladder!"<<endl;

pos=78;

}

else if(pos==18){

cout<<"Ladder!"<<endl;

pos=37;

}

else if(pos==28){

cout<<"Ladder!"<<endl;

pos=84;

}

else if(pos==21){

cout<<"Ladder!"<<endl;

pos=42;

}

else if(pos==51){

cout<<"Ladder!"<<endl;

pos=67;

}

else if(pos==71){

cout<<"Ladder!"<<endl;

pos=91;

}

else if(pos==80){

cout<<"Ladder!"<<endl;

pos=97;

}

else if(pos==32){

cout<<"Ladder!"<<endl;

pos=49;

}

else if(pos==17){

cout<<"Snake!"<<endl;

pos=7;

}

else if(pos==64){

cout<<"Snake!"<<endl;

pos=60;

}

else if(pos==62){

cout<<"Snake!"<<endl;

pos=19;

}

else if(pos==87){

cout<<"Snake!"<<endl;

pos=24;

}

else if(pos==54){

cout<<"Snake!"<<endl;

pos=34;

}

else if(pos==98){

cout<<"Snake!"<<endl;

pos=79;

}

else if(pos==95){

cout<<"Snake!"<<endl;

pos=75;

}

else if(pos==93){

cout<<"Snake!"<<endl;

pos=53;

}

else if(pos==55){

cout<<"Snake!"<<endl;

pos=3;

}

else if(pos==29){

cout<<"Snake!"<<endl;

pos=6;

}

//Random Bonus and Drop Points

else if(pos==99){

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

pos=pos-drop;

}

else if(pos==10){

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

pos=pos+bonus;

}

else if(pos==50){

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

pos=pos-drop;

}

else if(pos==76){

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

pos=pos+bonus;

}

else{

pos=pos+0;

}

cout<<"Current Place: "<<pos<<endl;

cout<<endl;

break;

}

default:{

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

}

}

}

//Get Position to check if they won

int Player1::getPos(){

return pos;

}

//Set Username

void Player1::setName(){

cout<<"Enter desired username for Player 1: ";

cin>>name;

}

//Position >= 100 set wins

void Player1::setWins(){

wins=wins+1;

}

//Get Wins for files

int Player1::getWins(){

return wins;

}

//Reset Position for New Game

void Player1::reset(){

pos=0;

}

//Get Name for Files

string Player1::getName(){

return name;

}

//Destructor

Player1::~Player1(){

}

**Source Code (player2.h):**

/\*

\* File: player2.h

\* Author: Javier Ventura

\* Purpose: Specification for Player 2 object class inherited from Game class

\* Created on December 7, 2017, 4:27 PM

\*/

#ifndef PLAYER2\_H

#define PLAYER2\_H

#include "game.h"

#include <string>

using std::string;

class Player2:public Game{

private:

//Position on the board

int pos;

//Sum of both die rolled

int points;

//Username of player

string name;

//Number of wins in a session

int wins;

public:

//constructor

Player2();

//Copy Constructor

Player2(const Player2 &);

//Set Position on Board

void setPos();

//Get Position to check where player is

int getPos();

//Set Desired Username

void setName();

//Set Number of Wins when player position reaches 100

void setWins();

//Get Number of Wins for Statistics

int getWins();

//Reset Position for new Game

void reset();

//Get Name for file

string getName();

//Destructor

~Player2();

};

#endif /\* PLAYER2\_H \*/

**Source Code (player2.cpp):**

/\*

\* File: player2.cpp

\* Author: Javier Ventura

\* Purpose: Implementation of player 2 class

\* Created on December 7, 2017, 5:03 PM

\*/

#include <string>

#include <iostream>

#include "game.h"

#include "player2.h"

using namespace std;

//Constructor

Player2::Player2(){

wins=0;

pos=0;

points=0;

}

void Player2::setPos(){

//Set Random Number Seed

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

// 1 of two Die range is from 1 to 6

int die1=rand()%6+1;

// 1 of two Die range is from 1 to 6

int die2=rand()%6+1;

// Press 1 to roll, interface for rolling

int roll;

// Possible bonus points to move forward 1 to 10

int bonus=rand()%10+1;

// Possible random drop from 1 to 10

int drop=rand()%10+1;

cout<<"Player 2 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;

points=die1+die2;

pos=pos+points;

//Set Traps and Ladder Jumps

if(pos==4){

cout<<"Ladder!"<<endl;

pos=14;

}

else if(pos==9){

cout<<"Ladder!"<<endl;

pos=31;

}

else if(pos==13){

cout<<"Ladder!"<<endl;

pos=74;

}

else if(pos==66){

cout<<"Ladder!"<<endl;

pos=78;

}

else if(pos==18){

cout<<"Ladder!"<<endl;

pos=37;

}

else if(pos==28){

cout<<"Ladder!"<<endl;

pos=84;

}

else if(pos==21){

cout<<"Ladder!"<<endl;

pos=42;

}

else if(pos==51){

cout<<"Ladder!"<<endl;

pos=67;

}

else if(pos==71){

cout<<"Ladder!"<<endl;

pos=91;

}

else if(pos==80){

cout<<"Ladder!"<<endl;

pos=97;

}

else if(pos==32){

cout<<"Ladder!"<<endl;

pos=49;

}

else if(pos==17){

cout<<"Snake!"<<endl;

pos=7;

}

else if(pos==64){

cout<<"Snake!"<<endl;

pos=60;

}

else if(pos==62){

cout<<"Snake!"<<endl;

pos=19;

}

else if(pos==87){

cout<<"Snake!"<<endl;

pos=24;

}

else if(pos==54){

cout<<"Snake!"<<endl;

pos=34;

}

else if(pos==98){

cout<<"Snake!"<<endl;

pos=79;

}

else if(pos==95){

cout<<"Snake!"<<endl;

pos=75;

}

else if(pos==93){

cout<<"Snake!"<<endl;

pos=53;

}

else if(pos==55){

cout<<"Snake!"<<endl;

pos=3;

}

else if(pos==29){

cout<<"Snake!"<<endl;

pos=6;

}

//Random Bonus and Drop Points

else if(pos==99){

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

pos=pos-drop;

}

else if(pos==10){

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

pos=pos+bonus;

}

else if(pos==50){

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

pos=pos-drop;

}

else if(pos==76){

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

pos=pos+bonus;

}

else{

pos=pos+0;

}

cout<<"Current Place: "<<pos<<endl;

cout<<endl;

break;

}

default:{

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

}

}

}

//Get Position to check where Player is on the board

int Player2::getPos(){

return pos;

}

//Setting player Username

void Player2::setName(){

cout<<"Enter desired username for Player 2: ";

cin>>name;

}

//Set Wins if player reaches spot 100

void Player2::setWins(){

wins=wins+1;

}

//Reset the position for new game

void Player2::reset(){

pos=0;

}

//Get Number of Wins for stats

int Player2::getWins(){

return wins;

}

//Get the Players Name for Stats File

string Player2::getName(){

return name;

}

//Destructor

Player2::~Player2(){

}