7/21/2014

Trajon Felton

CIS-5

Project 1

Game

**Introduction**

Title: Animal Kingdom

This is a game that is based on the popular game rock-paper-scissors.

The Basic rules of the game is to select one of the three animals which are the tiger, bear, or wolf using the first letter of that animal as input.

Bears are superior to tigers, who are superior to wolfs, who are superior to bears.

If you select an animal that is superior toward another animal, then you have acquired a win against the AI.

If you and the ai select the same animal, then the round is a tie, and will not be counted in the game

Else, if the AI selects an animal superior to yours, then you have lost that round

After the game is played, results are printed onto the file games.dat located in the same file as the project.

This game shows a large amount of statistics based on the success rate of the animal you choose that acquires a win.

**Summary**

Project size: about 230 lines

The number of variables: about 27

The number of methods: 19

Since the concept of the project was really simple to outline. Originally what I had was about 60 lines of code. Since this needed more code to implement into this simple program what I then started to add were while loops, switches, and a counter to increment certain variables in order to collect data for the user to look at when done at the game. After a certain condition was met, on the console I outputted a neat display based on what the user and the AI selected. After that took around 100 lines, What I then added were more variables and the file output stream to display data on the game that made it more statistical above all else. After finishing all this, I was very satisfied with the result, especially since this was the first time I’ve used output on a program

Project took about 3 days to code.

This project was not very difficult for me since I knew exactly what I wanted to do.

The only real problem that I met with was making sure to assign each counter equal to **zero** before putting them in the loop to increment.

When using this project again in the near future I plan to increase the anima size to 5 as well as create an animal that dominates all the other animals, however can only be used once.

**Description**

The main point of the program is inputting a letter, based on the animal you want. Based on a random instance run through the system, will display a result of either a tie, a win, or a loss.

Psudeo Code

Start Program

Declare Variables

Open the file games.dat

Output directions for the game

Get the number of games

If games is not an odd number, prompt for number of games again.

Get user input for the animal they wish to select

If user dosent select a letter that is related to the animals prompt again

Generate Statements based on user input

For this case, 1 = wolf, 2 =tiger, 3 =bear

If the user selects a tiger verses a wolf he wins

If the user selects a wolf vs a bear he wins

If the user selects a bear vs a tiger he wins

Else If the user selects a wolf agaisnt a wolf, it ends in a tie

Else if the user selects a tiger against a tiger, it ends in a tie

Else if the user selects a bear against a bear, it ends in a tie

Since it is a tie, the game won’t count/ it’s nullified.

Else the user selects a bear vs a wolf, he gets hunted by a pack and loses.

Else the user selects a tiger vs a bear, he gets manhandled in the jungle and loses

Else the user selects a wolf vs a tiger, the king of the jungle dominates the pack like simba.

Display the winner

Calculations for percentage success

Display information on console that details can be found on file

In the file:

Based on the game itself:

Displays results of the user,

Display the data for the user

Displays results of the computer

Prompt user to play again, Noting if they choose yes, file will be wiped clean

Get input from the choice.

If yes, loop again

Else exit the loop and end the program.

Documentation of Variable, Constructs and Libraries

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Major Variables | | | | | | |
|  | | | | | | |
| Type | Variable Name | | | Description | | Location |
| Unsigned Short | Seed | | | Used to generate a seed for the random function | | Line 27 |
|  | Ai | | | Used to take in the random generated number from seed | | Line 31 |
|  | aTig | | | Number of times the computer used the tiger | | Line 39 |
|  | aBea | | | Number of times the computer used the bear | | Line 40 |
|  | aWol | | | Number of times the computer used the wolf | | Line 41 |
|  | uTig | | | Number of times the userused the tiger | | Line 46 |
|  | uBea | | | Number of times the user used the bear | | Line 47 |
|  | uWol | | | Number of times the user used the wolf | | Line 48 |
|  | sWol2 | | | Number of successful times the computer used the wolf | | Line 42 |
|  | sBea2 | | | Number of successful times the computer used the bear | | Line 43 |
|  | sTig 2 | | | Number of successful times the computer used the tiger | | Line 44 |
|  | sWol | | | Number of successful times the bear was used by the user | | Line 51 |
|  | sBea | | | Number of successful times the bear was used by the user | | Line 50 |
|  | sTig | | | Number of successful times the tiger was used by the user | | Line 49 |
| Float | perT | | | Calculates the percentage of success for tigers used by the user | | Line 53 |
|  | perB | | | Calculates the percentage of success for bears used by the user | | Line 54 |
|  | perW | | | Calculates the percentage of success for wolfs used by the user | | Line 55 |
|  | perT2 | | | Calculates the percentage of success for tigers used by the computer | | Line 57 |
|  | perB2 | | | Calculates the percentage of success for bears used by the computer | | Line 58 |
|  | perW | | | Calculates the percentage of success for wolfs used by the computer | | Line 59 |
| Unsigned int | wins | | | Shows the number of wins | | Line 63 |
|  | losses | | | Shows the number of losses | | Line 64 |
|  | ties | | | Shows the number of ties | | Line 65 |
|  | i | | | Increments the loop based on the number of games | | Line 78 |
|  | games | | | Used to get the number of games, without including ties, and only  takes in odd integers | | Line 34 |
|  | total | | | Gets the total number of games played, including ties. | | 37 |
| char | input | | | Used to get the animal the user wants to use. Only takes in the letters w, t, or b | | Line 33 |
|  | cond | | | Used to get whether the user wants to play the game again | | Line 32 |
| Ofstream | Output | | | Used to put results on the page | | Line 36 |
| SYSTEM LIBRARIES | | | | | | |
| iostream | | **Used for outputting and inputing into the counsole** | | | | Line 8 |
| cstdlib | | **Used for the rand() and srand() operator to get random results** | | | | Line 9 |
| Ctime | | **used for the time function to generate the number of seconds since 1970** | | | | Line 10 |
| Fstream | | **Used to output data onto the file games.dat** | | | | Line 11 |
| Iomanip | | **Used to get spacing onto the file** | | | | Line 12 |
| C++ Constructs | | | | | | |
| Chapter 2 | | | **cin** | | Line 73 | |
|  | | | **cout** | | Line 66 | |
|  | | | **Input operator >>** | | Line 75 | |
|  | | | **Output operator <<** | | Line 66 | |
|  | | | **main function** | | Line 25 | |
|  | | | **// comment line** | | Line 26 | |
|  | | | **/\* \*/ comment block** | | Line 1-5 | |
| Chapter 3 | | | **Time()** | | Line 28 | |
|  | | | **rand()** | | Line 31 | |
|  | | | **srand** | | Line 29 | |
|  | | | **Modulus operator %** | | Line 31 | |
|  | | | **Addition operator +** | | Line 31 | |
|  | | | **Assignment operator =** | | Line 37 | |
|  | | | **Multiplication operator \*** | | Line 192 | |
|  | | | **Division operator /** | | Line 192 | |
|  | | | **setw()** | | Line 201 | |
|  | | | **setprecision()** | | Line 214 | |
|  | | | **showpoint** | | Line 214 | |
|  | | | **Fixed** | | Line 214 | |
|  | | | **Prentices operator ()** | | Line 192 | |
| Chapter 4 | | | **If-then statement** | | Line 188-190 | |
|  | | | **If-then, else if, else statement** | | Line 124-149 | |
|  | | | **Switch case** | | Line 90 | |
|  | | | **== conditional operator** | | Line 93 | |
|  | | | **> conditional operator** | | Line 188 | |
|  | | | **! conditional operator** | | Line 84 | |
|  | | | **break** | | Line 122 | |
|  | | | **&& conditional operator** | | Line 84 | |
|  | | | **|| conditional operator** | | Line 231 | |
| Chapter 5 | | | **Do-While statement** | | Line 66 | |
|  | | | **While statement** | | Line 75 | |
|  | | | **For statement** | | Line 79 | |
|  | | | **Open file member fstream** | | Line 61 | |
|  | | | **Close file member fstream** | | Line 228 | |
|  | | | **++ increment operator** | | Line 180 | |
|  | | | **--decrement operator** | | Line 178 | |

Sample Runs

|  |  |
| --- | --- |
| Sample Run 1 |  |
| Sample Run 2 |  |
| Sample File Run 1 |  |

**Program**

/\*

\* File: main.cpp

\* Author: Trajon Fetlon

\* Created on July 10, 2014, 1:31 PM

\*/

//System Libraries

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <fstream>

#include <iomanip>

using namespace std;

//User Libraries

//Global Constants

//Function Prototypes

//Execution Begins Here

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

//Declare Variables

unsigned short seed; //Needed to generate random results

seed = time(0); //Set seed equal to time based on seconds from 1970

srand(seed); //places the number in the random generator

unsigned short ai; //Computer commands for the game

ai = rand() % 3 + 1; //Needed for the computer to select 3 random commands

char input; //User input based on the first letter of the animal

unsigned int games; //Number of games a player may wish to embark upon

ofstream output; //Used to output results and data onto a seperate file

unsigned int total = 0; //Number of games played total, including ties

//Ai related data

unsigned short aTig = 0; //Number of times the computer used the tiger

unsigned short aBea = 0; //Number of times the computer used the bear

unsigned short aWol = 0; //Number of times the computer used the wolf

unsigned short sWol2 = 0; //Number of successful times the computer used wolf

unsigned short sBea2 = 0; //Number of successful times the computer used the bear

unsigned short sTig2 = 0; //Number of successful times the computer used the tiger

//User related data

unsigned short uTig = 0; //Number of times the user used the tiger

unsigned short uBea = 0; //Number of times the user used the bear

unsigned short uWol = 0; //Number of times the user used the wolf

unsigned short sTig = 0; //Number of successful times the tiger was used by user

unsigned short sBea = 0; //Number of successful times the bear was used by user

unsigned short sWol = 0; //Number of successful times the wolf was used by the user

//Percentage of success

float perT = 0; //Used to calculate the percentage of success for tigers, user related

float perB = 0; //Used to calculate the percentage of success for Bears, user related

float perW = 0; //Used to calculate the percentage of success for wolf, user related

//AI percentage of success

float perT2 = 0; //Used to calculate the percentage of success for tiger, computer related

float perB2 = 0; //Used to calculate the percentage of success for bear, computer related

float perW2 = 0; //Used to calculate the percentage of success for wolf, computer related

//Open the file games.dat

output.open("games.dat");

//Output directions for the game

unsigned int wins = 0; //Shows the number of wins

unsigned int losses = 0; //Shows the number of losses

unsigned int ties = 0; //Shows the number of ties

cout << "Welcome to the game Animal Kingdom. This ever changing game" << endl;

cout << "was made with the thought of rock paper scissors in mind. For the purpose" << endl;

cout << "of this game, we will use the tiger a bear, and a wolf. Wolves dominates bears, Bears" << endl;

cout << "dominate Tigers and Tigers dominate Wolves. You need to compete with the AI 5 times in order" << endl;

//Get user input for the animal they wish to select

cout << "to win. If the AI however manages to defeat you 5 times then you lose. Now lets begin the game!" << endl << endl;

cout << "Enter the number of games you wish to embark upon" << endl;

cin >> games;

while(games % 2 == 0){

cout << "Please enter an odd number of games so a winner can be announced" << endl;

cin >> games;

}

for(unsigned int i = 0; i < games; i++){

ai = rand() % 3 + 1; //Recalculated in the loop in order to reset the generator

cout << "Enter the command for the animal you wish to select. t for Tiger, b for Bear, and w for wolves." << endl;

cin >> input;

cout << endl;

while(input != 't' && input != 'b' && input != 'w'){

cout << "Please enter t for tiger, w for wolf, or b for bear." << endl;

cin >> input;

cout << endl;

}

//Generate Statements based on user input

//For this case, 1 = wolf, 2 =tiger, 3 =bear

switch(ai){

case 1:{

if(input == 't'){ //If the user selects a tiger verses a wolf he wins

cout << "The tiger swallows the wolf in its last effort to survive." << endl;

cout << "You have won this round." << endl << endl;

wins++; //Number of wisn the user has

uTig++; //Number of times the user used the tiger

aWol++; //Number of times the wolf was used by computer total

total++; //Total number of games played regardless of ties

sTig++; //Successful nubmer of tiems the wolf was used by the computer

}

else if(input == 'w'){ //If the user selects a wolf agaisnt a wolf, it ends in a tie

cout << "The wolves stare at each other "

"in agony as they find respect for the other." << endl;

cout << "This round is a tie." << endl << endl;

ties++; //Number of ties in the game

uWol++; //Number of times the user used the wolf

i--; //Decrements the counter for a tie, in order to have a winner result

aWol++; //Total number of times teh wolf was used

total++; //Total number of games played regardless of ties

}

else{ //If the user selects a bear vs a wolf, he gets hunted by a pack and loses.

cout << "As the Bear goes down the forest, the wolf attacks in an onslaught pack." << endl;

cout << "You have lost this round" << endl << endl;

losses++; //Increments the total number of losses

aWol++; //Total number of tiems the wolf was used

total++; //Increments the total number of games regardless of total

sWol2++; //Successful times the wolf was used by computer

uBea++;

}

break;

}

case 2:{ //If the number is 2, it is a tiger

if(input == 't'){

cout << "The tiger comes back to his home, to find another tiger staring at him in confusion." << endl;

cout << "This round is a tie." << endl << endl;

ties++; //Increments the number of ties in this game

i--; //Decrements the counter due to the tie.

aTig++; //Number of times the tiger was used total

total++; //Increments the total number of games regardless of the tie

uTig++; //Number of times user used the tiger

}

else if(input == 'w'){

cout << "As the wolf runs away from the tiger, it remembers it's lost son. Losing all fear it turns back to get slaughtered." << endl;

cout << "You have lost this round." << endl << endl;

losses++; //Increments the number of losses user related

aTig++; //Nubmer of times the tiger was used by the computer total

sTig2++; //Number of successful times the tiger was used by computer

uWol++; //Number of times user used the wolf

}

else{

cout << "The bear sees the tiger pridefully on the mountaintop. The lung toward the bear allows the bear to swiftly end the tiger." << endl;

cout << "You have won this round." << endl << endl;

wins++; //Increments the number of wins, user related

aTig++; //AI's tiger counter

total++; //Increments the total number of games, regardless of ties

sBea++; //User successful Bear counter

uBea++; //Number of times user used the bear

}

break;

}

case 3:{ //If the number is 3, it is a bear

if(input == 't'){

cout << "The tiger oversteps his bounds on bear territory, and learns who is king of the tree." << endl;

cout << "You have lost this round." << endl << endl;

losses++; //Increments the number of losses

aBea++; //Number of times the bear was used by the computer total

total++; //Number of games play total, regardless of tes

sBea2++; //Number of successful times the bear was used by the computer

uTig++; //Number of times the user used the tiger

}

else if(input == 'w'){

cout << "The wolf descends upon the helpless bear as it sips on the honey tree." << endl;

cout << "You have won this round." << endl << endl;

wins++; //Increments the number of wins

aBea++; //Ai's use of bear counter

total++; //Total number of games played

sWol++; //User's successful use of tiger

uTig++; //Number of tiems the user used the tiger

}

else{

cout << "The bears walk around the tree waiting for a chance to lick the tree." << endl;

cout << "This round is a tie." << endl << endl;

ties++; //Increments the number of ties

i--; //Decrements the counter of games, since this is a tie

aBea++; //Number of times the bear was used total

total++; //Increments the toatl number of games regardless of ties

uBea++; // Number of times the user used the bear

}

break;

}

}

}

//Display the winner

if(wins > losses){

cout << "You have won the game congratulations. You have served the animal kingdom well." << endl << endl;

}

//Calculations for percentage success

perT2 = (1.0 \* sTig2/aTig) \* 100; //Calculates the success rate for computer's use of tiger

perW2 = (1.0 \* sWol2/aWol) \* 100; //Calculates the success rate for computer's use of wolf

perB2 = (1.0 \* sBea2/aBea) \* 100;//Calculates the success rate for computer's use of bear

perT = (1.0 \* sTig/uTig) \* 100; //Calculates the success rate for the users use of tiger

perW = (1.0 \* sWol/uWol) \* 100; //Calculates the success rate for the users use of wolf

perB = (1.0 \* sBea/uBea) \* 100; //Calculates the success rate for the users use of bear

//Display information on console

cout << "Details of the game are on the file game.dat, located in the same file as this program" << endl;

output << "Number of games to: " << setw(21) << games << endl;

output << "Number of total games played, including ties: " << total << endl << endl;

//Displays results of the user

output << "Number of wins for the user: " << setw(6) << wins << endl;

output << "Number of losses for the user: " << setw(4) << losses << endl;

output << "Number of ties produced this game: " << setw(6) << ties << endl << endl;

//Display the data for the user

output << "Number of times the tiger was used by user: " << uTig << endl;

output << "Number of times the wolf was used by user: " << uWol << endl;

output << "Number of times the bear was used by user: " << uBea << endl << endl;

output << "Number of successful times the tiger was used by user: " << sTig << endl;

output << "Number of successful times the wolf was used by user: " << sWol << endl;

output << "Number of successful times the bear was used by user: " << sBea << endl << endl;

output << "Success rate of Tiger for computer: " << setprecision(2) << fixed << showpoint << perT << "%" << endl;

output << "Success rate of Bear for computer: " << setprecision(2) << fixed << showpoint << perB << "%" << endl;

output << "Success rate of Wolf for computer: " << setprecision(2) << fixed << showpoint << perW << "%" << endl << endl;

//Displays results of the computer

output << "Number of times the computer used the tiger: " << setw(5) << aTig << endl;

output << "Number of times the computer used the wolf: " << setw(6) << aWol << endl;

output << "Number of times the computer used the bear: " << setw(6) << aBea << endl << endl;

output << "Number of successful times the tiger was used by computer: " << sTig2 << endl;

output << "Number of successful times the wolf was used by computer: " << sWol2 << endl;

output << "Number of successful times the bear was used by computer: " << sBea2 << endl << endl;

output << "Success rate of Tiger for computer: " << setprecision(2) << fixed << showpoint << perT2 << "%" << endl;

output << "Success rate of Bear for computer: " << setprecision(2) << fixed << showpoint << perB2 << "%" << endl;

output << "Success rate of Wolf for computer: " << setprecision(2) << fixed << showpoint << perW2 << "%" << endl << endl;

output.close();

return 0;

}