7/30/2014

Project 2

Game

Trajon Felton
CIS-5

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 five animals which are the tiger, bear,wolf,man,dragon using the first letter of that animal as input.

Wolfs dominate bears and dragons
Bears dominate Tigers and Man
Tigers dominate Wolves and Man
Dragons conquer Tiger and Bear
Man conquers Dragon and wolf

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 535 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 5 days cummulative 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.

Doing the Project again for part 2, the problem that I ran across in general was initializing arrays with the right numbers, and the sorting functions developed by this project. I realized every time that I sort an array of names, that I would have to reinitialize the names in order to get the proper names of each data printed out into the array.

Project took 3 full days of coding non stop in order to code.
I knew what I wanted to include in this project. However the implementation of the functions and overloading was what caused the difficulty around this time.

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.

Flowchart can be found in the file. Too big to be seen on word.

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,4=Dragon,5=Wolf
            If the user selects an animal that dominates the other he wins
            Else if the user selects an animal that sucks to the other animal he loses
            Else the animals are the same, and the game results in a tie, not counting as a game.
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
                        Displays the average rate of success for computer and user
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 31
	Ai	Used to take in the random generated number from seed	Line 34
	aTig	Number of times the computer used the tiger	Line 46
	aBea	Number of times the computer used the bear	Line 47
	aWol	Number of times the computer used the wolf	Line 48
	aDra	Number of times the computer used the dragon	Line 49
	aMan	Number of times the computer used Man	Line 50
	uTig	Number of times the user used the tiger	Line 57
	uBea	Number of times the user used the bear	Line 58
	uWol	Number of times the user used the wolf	Line 59
	uDra	Number of times the user used the dragon	Line 60
	uMan	Number of times the user used Man	Line 61
	sWol2	Number of successful times the computer used the wolf	Line 51
	sBea2	Number of successful times the computer used the bear	Line 52
	sTig 2	Number of successful times the computer used the tiger	Line 53
	sDra2	Number of successful times the dragon was used by the computer	Line 54
	sMan2	Number of successful times Man was used by the compter	Line 55

	sWol	Number of successful times the bear was used by the user	Line 64
	sBea	Number of successful times the bear was used by the user	Line 63
	sTig	Number of successful times the tiger was used by the user	Line 62
	sDra	Number of successful times the dragon was used by the user	Line 65
	sMan	Number of successful times Man was used by the user	Line 66
Float	PerT	Calculates the percentage of success for tigers used by the user	Line 68
	perB	Calculates the percentage of success for bears used by the user	Line 69
	perW	Calculates the percentage of success for wolfs used by the user	Line 70
	perD	Calculates the percentage of success for dragons used by the user	Line 71
	perM	Calculates the percentage of success for Man used by the computer	Line 72
	perT2	Calculates the percentage of success for tigers used by the computer	Line 75
	perD2	Calculates the success rate of the use of dragon for the computer	Line 78
	perM2	Calculates the success rate of the use of man for the computer	Line 79
	perB2	Calculates the percentage of success for bears used by the computer	Line 76
	perW2	Calculates the percentage of success for wolfs used by the computer	Line 77
Unsigned int	Wins	Shows the number of wins	Line 85
	Losses	Shows the number of losses	Line 86
	Ties	Shows the number of ties	Line 87
	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.	Line 44
char	input	Used to get the animal the user wants to use. Only takes in the letters w, t, or b	Line 37
	cond	Used to get whether the user wants to play the game again	Line 36
String	Line	Used to output the title of the game from file	Line 42
ofstream	Output	Used to put results on the page	Line 40
ifstream	Title	Used to read in a file that displays the title of the game	Line 41
Unsigned short arrays[]	a[5]	Used to store the values of all the computer related uses of commands into an array	Line 343
	s[5]	Used to store the number of successful times the user used a command in the game	Line 346
	s2[5]	Used to store the number of successful times the computer used a command in the game	Line 344
	u[5]	Used to store the number of times the computer used a command in the game	Line 345
Float arrays[]	p2[5]	Used to store the number the success rate of input of the commands used by the computer	Line 364
	p[5]	Used to store the success rate of input of the commands used by the user	Line 370
String array[]	Names	Used to store the names of the command inputs	81

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

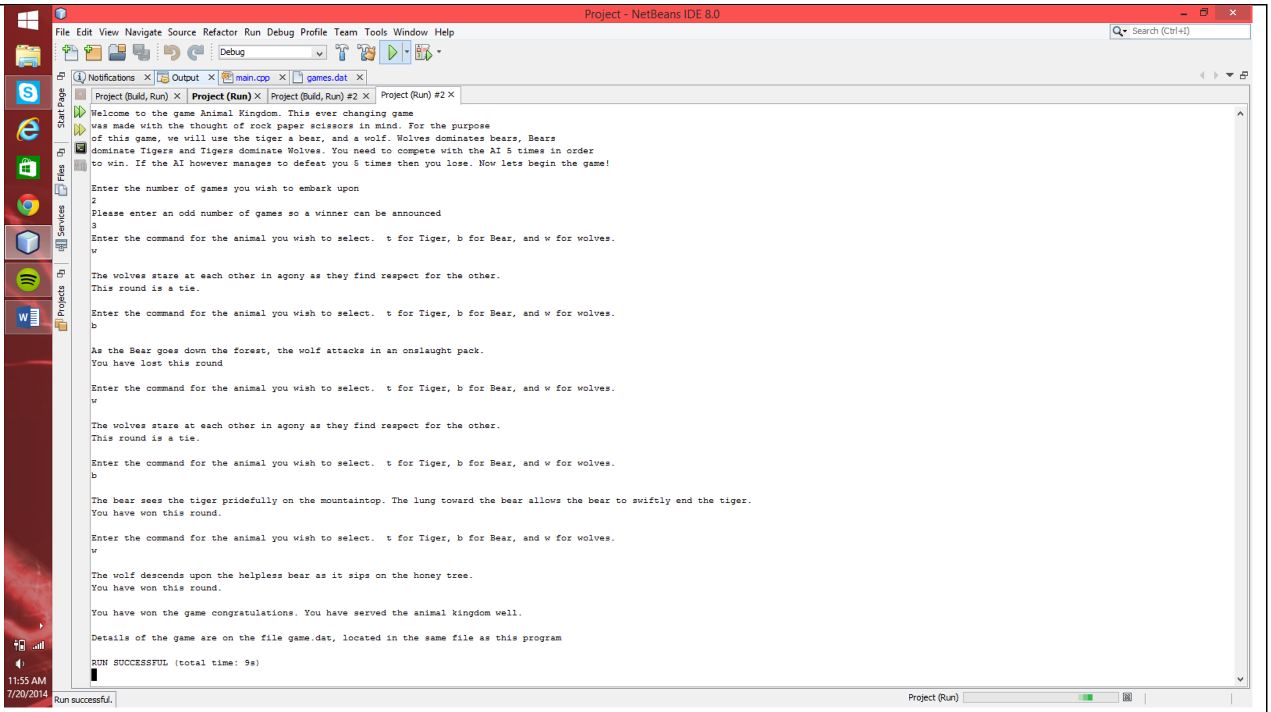
Functions

Void Prompt()	**Displays the prompt of the game, takes in no data types**	Line 22
Float perc(float,float)	**Used to calculate the percentage of success of a command**	Line 23
Float perc(unsigned short&, unsigned short&)	**Used to calculate the percentage of successs of a command. Takes in shorts, instead of floats**	Line 24
Void sort(unsinged short [], string [], int)	**Used to sort arrays of shorts with there names to display in file**	Line 25
Void sort(float arr[], string names[], short size)	**Used to sort arrays of floatswith the names to display in file**	Line 26
C++ Constructs		
Chapter 2	**Cin**	Line 99
	Cout	Line 90
	Input operator >>	Line 99
	Output operator <<	Line 90
	main function	Line 29
	// comment line	Line 28
	/* */ comment block	Line 1-5
Chapter 3	**Time()**	Line 32
	rand()	Line 35
	srand	Line 33
	Modulus operator %	Line 35
	Addition operator +	Line 102
	Assignment operator =	Line 37
	Multiplication operator *	Line 473
	Division operator /	Line 473
	setw()	Line 381
	setprecision()	Line 406
	showpoint	Line 406
	Fixed	Line 434
	getline	Line 89
	Prentices operator ()	Line 523
Chapter 4	**If-then statement**	Line 355
	If-then, else if, else statement	Line 216-260
	Boolean	Line 497
	Switch case	Line 114
	== conditional operator	Line 117
	> conditional operator	Line 355

	! conditional operator	Line 107
	Break	Line 163
	&& conditional operator	Line 107
	|| conditional operator	Line 447
Chapter 5	**Do-While statement**	Line 88
	While statement	Line 89
	For statement	Line 101
	Open file member fstream	Line 83
	Close file member fstream	Line 443
	++ increment operator	Line 502
	--decrement operator	Line 132
Chapter 6	**Void**	Line 22
	Value returning function	Line 23
	Call by reference	Line 24
	return	Line 491
	Overload functions	Line 24-25
Chapter 7	**Call by array**	Line 26
	Array initalization	Line 437
	2D Array initialization	Line 371
	For Loop array initialization	Line 372-375
	Bubble sort	Line 496-513

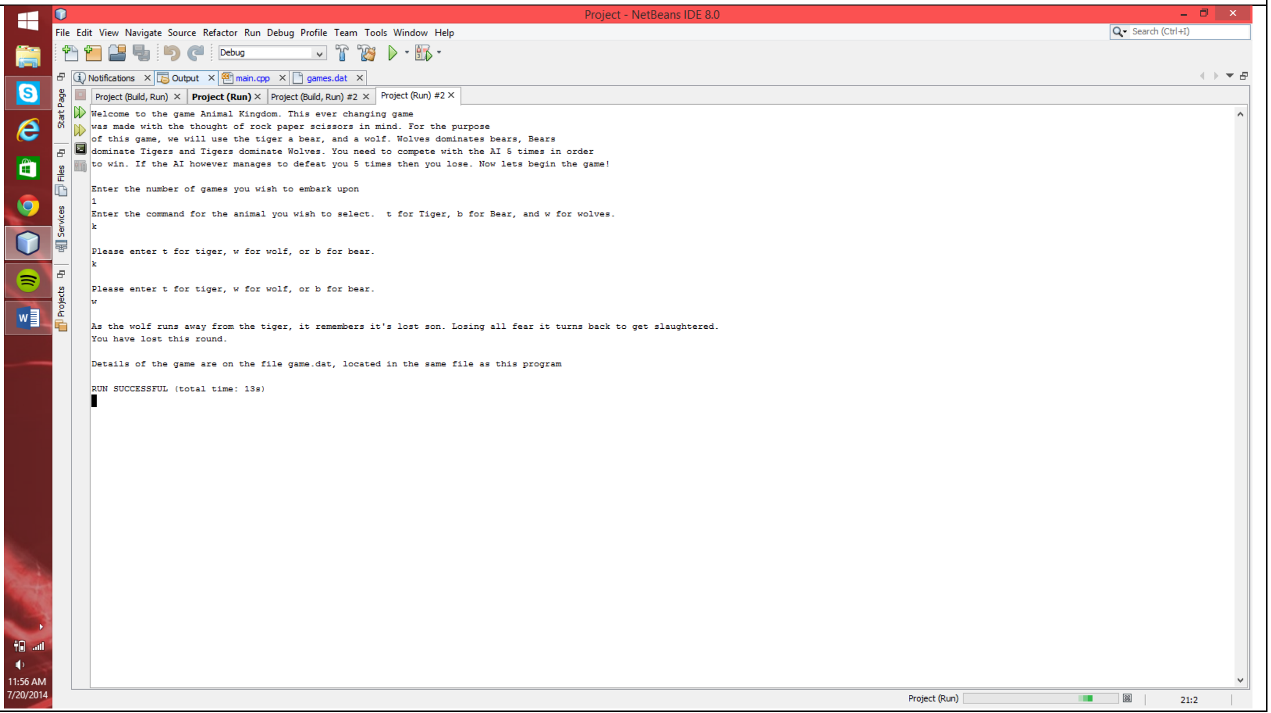
Sample Runs

Sample Run 1



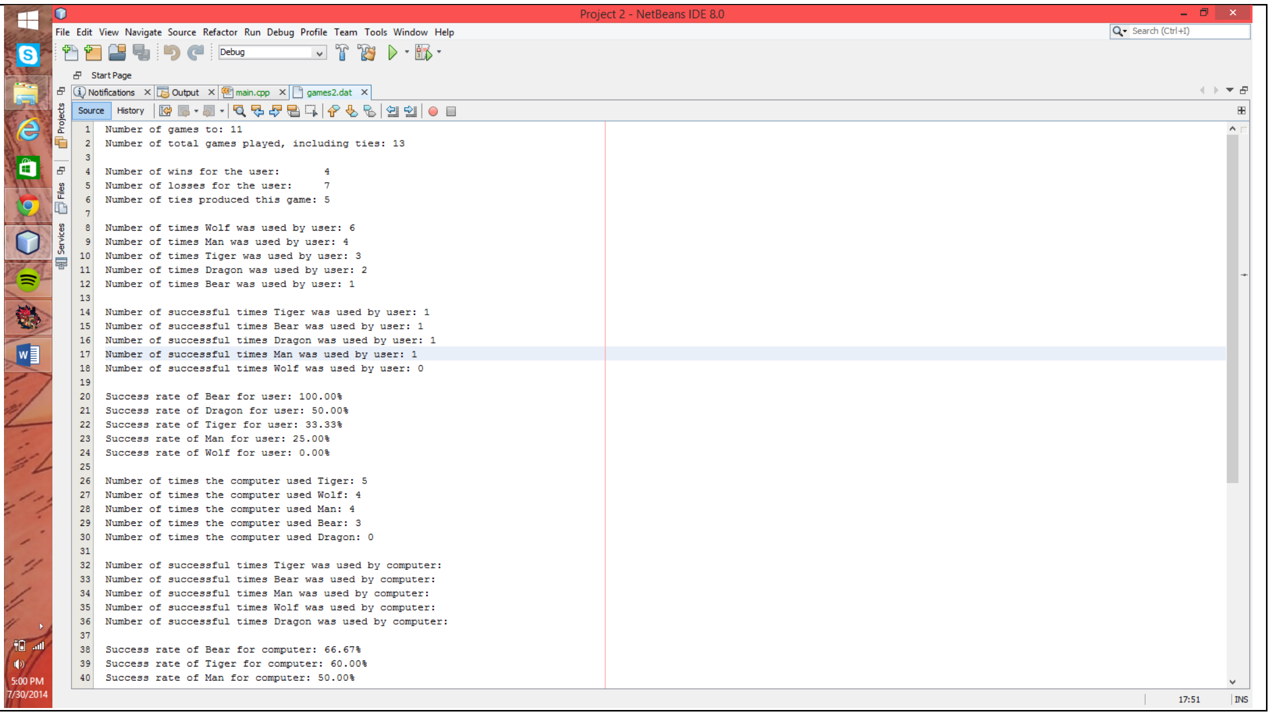
```
Welcome to the game Animal Kingdom. This ever changing game
was made with the thought of rock paper scissors in mind. For the purpose
of this game, we will use the tiger a bear, and a wolf. Wolves dominates bears, Bears
dominate Tigers and Tigers dominate Wolves. You need to compete with the AI 5 times in order
to win. If the AI however manages to defeat you 5 times then you lose. Now lets begin the game!

Enter the number of games you wish to embark upon
2
Please enter an odd number of games so a winner can be announced
3
Enter the command for the animal you wish to select.  t for Tiger, b for Bear, and w for wolves.
w

The wolves stare at each other in agony as they find respect for the other.
This round is a tie.

Enter the command for the animal you wish to select.  t for Tiger, b for Bear, and w for wolves.
b

As the Bear goes down the forest, the wolf attacks in an onslaught pack.
You have lost this round

Enter the command for the animal you wish to select.  t for Tiger, b for Bear, and w for wolves.
w

The wolves stare at each other in agony as they find respect for the other.
This round is a tie.

Enter the command for the animal you wish to select.  t for Tiger, b for Bear, and w for wolves.
b

The bear sees the tiger pridefully on the mountaintop. The lung toward the bear allows the bear to swiftly end the tiger.
You have won this round.

Enter the command for the animal you wish to select.  t for Tiger, b for Bear, and w for wolves.
w

The wolf descends upon the helpless bear as it sips on the honey tree.
You have won this round.

You have won the game congratulations. You have served the animal kingdom well.

Details of the game are on the file game.dat, located in the same file as this program

RUN SUCCESSFUL (total time: 9s)
```

Sample Run 2



```
Welcome to the game Animal Kingdom. This ever changing game
was made with the thought of rock paper scissors in mind. For the purpose
of this game, we will use the tiger a bear, and a wolf. Wolves dominates bears, Bears
dominate Tigers and Tigers dominate Wolves. You need to compete with the AI 5 times in order
to win. If the AI however manages to defeat you 5 times then you lose. Now lets begin the game!

Enter the number of games you wish to embark upon
1
Enter the command for the animal you wish to select.  t for Tiger, b for Bear, and w for wolves.
k

Please enter t for tiger, w for wolf, or b for bear.
k

Please enter t for tiger, w for wolf, or b for bear.
w

As the wolf runs away from the tiger, it remembers it's lost son. Losing all fear it turns back to get slaughtered.
You have lost this round.

Details of the game are on the file game.dat, located in the same file as this program

RUN SUCCESSFUL (total time: 13s)
```

Sample File Run 1



```
Number of games to: 11
Number of total games played, including ties: 13

Number of wins for the user:       4
Number of losses for the user:     7
Number of ties produced this game: 5

Number of times Wolf was used by user: 6
Number of times Man was used by user: 4
Number of times Tiger was used by user: 3
Number of times Dragon was used by user: 2
Number of times Bear was used by user: 1

Number of successful times Tiger was used by user: 1
Number of successful times Bear was used by user: 1
Number of successful times Dragon was used by user: 1
Number of successful times Man was used by user: 1
Number of successful times Wolf was used by user: 0

Success rate of Bear for user: 100.00%
Success rate of Dragon for user: 50.00%
Success rate of Tiger for user: 33.33%
Success rate of Man for user: 25.00%
Success rate of Wolf for user: 0.00%

Number of times the computer used Tiger: 5
Number of times the computer used Wolf: 4
Number of times the computer used Man: 4
Number of times the computer used Bear: 3
Number of times the computer used Dragon: 0

Number of successful times Tiger was used by computer:
Number of successful times Bear was used by computer:
Number of successful times Man was used by computer:
Number of successful times Wolf was used by computer:
Number of successful times Dragon was used by computer:

Success rate of Bear for computer: 66.67%
Success rate of Tiger for computer: 60.00%
Success rate of Man for computer: 50.00%
```

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
void prompt();
float perc(float,float);
float perc(unsigned short&,unsigned short&);
void sort(unsigned short arr[],string names[],int size);
void sort(float arr[],string names[],short size);

//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() % 5 + 1; //Needed for the computer to select 5 random commands
    char cond; //Used to get the condition to play again
    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
    ifstream title;
```

```
string line;
title.open("title.dat");
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 aDra = 0; //Number of times the computer used the dragon
unsigned short aMan = 0; //Number of times the computer used Man
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
unsigned short sDra2 = 0; //Number of successful times the computer used the dragon
unsigned short sMan2 = 0; //Number of successful times the computer used man
//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 uDra = 0; //Number of times the user used the dragon
unsigned short uMan = 0; //Number of times the user used Man
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
unsigned short sDra = 0; //Number of successful times the user used the dragon
unsigned short sMan = 0; //Number of successful times the user used man
//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
float perD = 0; //Used to calculate the percentage of success for dragon, user related
float perM = 0; //Used to calculate the percentage of success for man, user related
float uPer[5] = {perT, perB, perW, perD, perM };
//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
float perD2 = 0; //Used to calculate the percentage of success for dragon, computer related
float perM2 = 0; //Used to calculate the percentage of success for man, computer related
float aPer2[5] = { perT2, perB2, perW2, perD2, perM2 };
string names[5] =  {"Tiger", "Bear", "Wolf", "Dragon", "Man" };
//Open the file games.dat
output.open("games2.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
do{
while(getline(title,line)){
   cout << line << '\n';
}
cout << endl;
prompt();
cout << "Enter the number of plays up to. Must be an odd number" << endl;
//Get the number of games
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() % 5 + 1; //Recalculated in the loop in order to reset the generator
//Get user input for the animal they wish to select
cout << "Enter the command for the animal you wish to select.  t for Tiger, b for Bear,w for wolves, d for Dragon, and m for Man" << endl;
cin >> input;
cout << endl;
while(input != 't' && input != 'b' && input != 'w' && input != 'd' && input != 'm'){
    cout << "Please enter t for tiger, w for wolf, b for bear, d for dragon, and m for man" << endl;
    cin >> input;
    cout << endl;
}
//Generate Statements based on user input
//For this case, 1 = wolf, 2 =tiger, 3 =bear, 4 = dragon, 5 = Tiger
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 wins 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 number of times 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(input == 'd'){ //If the user selects a dragon, then there in different domains, and can't do
anything
    cout << "The dragon sores toward the sky as the wolf howls into the moon in agony" << endl;
    cout << "You lose this round" << endl << endl;
    losses++; //Number of losses in the game
    uDra++; //Number of times the user used the dragon
    sWol2++; //Number of successful times the computer used the wolf
    aWol++; //Number of times the computer used the artifical wolf
    total++; //Total number of games
  }
  else if(input == 'm'){
    cout << "Man turns wolf into a household pet." << endl;
    cout << "You win this round" << endl;
    wins++; //Number of losses in the game
    uMan++; //Number of times the user used man
    sMan++; //Number of successful man used in the  game by the user
    total++; //Number of total games
    aWol++; //Number of times the wolf was used by the computer
  }
  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 times the wolf was used
        total++; //Increments the total number of games regardless of total
        sWol2++; //Successful times the wolf was used by computer
        uBea++; //Number of times the user used the bear
    }
    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++; //Number 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 if(input == 'd'){ //If the user selects a dragon, then there in different domains, and can't do
    anything
                cout << "Crouching tiger, Hidden dragon. In this case, tiger overklls the dragon." << endl;
                cout << "You have won this round" << endl << endl;
                wins++; //Number of ties int eh game
                uDra++; //Number of times the user used the dragon
                sDra++; //Decrements the number of games, since it is a tie.
                aTig++; //Number of times the computer used the artifical wolf
                total++; //Total number of games
    }
            else if(input == 'm'){
                cout << "Tiger has taken revenge for the advancements of man" << endl;
                cout << "You have lost this round" << endl;
                losses++; //Number of losses in the game
                uMan++; //Number of times the user used man
                sTig2++; //Number of successful times the computer used the tiger
                total++; //Number of total games
                aTig++; //Number of times the Bear was used by the computer
    }
            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
        uWol++; //Number of tiems the user used the wolf
      }
      else if(input == 'd'){ //If the user selects a dragon, then there in different domains, and can't do anything
        cout << "Dragon saw the bear in the forest and made an example of why you should start wild fires" << endl;
        cout << "You have won this round." << endl << endl;
```

```
            wins++; //Number of wins in the game

            uDra++; //Number of times the user used the dragon

            sDra++; //Number of successful times the user used dragon

            aBea++; //Number of times the computer used the artifical bear

            total++; //Total number of games

    }

         else if(input == 'm'){

            cout << "As man tried to crouch in order to save himself, the bear pounced him, and devoured his legs" << endl;

            cout << "You lose this round" << endl;

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

            uMan++; //Number of times the user used man

            sBea2++; //Number of successful bear used in the  game by the computer

            total++; //Number of total games

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

    }

         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;

      }


    case 4:{ //If the number is 4 it is a dragon
```

```
if(input == 't'){
cout << "The tiger was completely devastated by the dragon" << endl;
cout << "You have won this round" << endl;
    wins++; //Number of losses in the game
    sDra2++; //Number of times the computer successfully used the dragon
    uTig++; //Number of times the user used a tiger
    total++; //Total number of games
}
else if(input == 'w'){
    cout << "The dragon war of 2014. The wolf's came in the thousands. Many were lost, however the dragon was taken down" << endl;
    cout << "You have won this round" << endl;
    wins++; //Number of losses in the game
    uWol++; //Number of times the user used the wolf
    sDra2++; //Number of dragons that the computer used successfully
    total++; //Total number of games
}
else if(input == 'b'){
    cout << "The bear went completely ballistic on the dragon, as he burned his tree, and the dragon slapped him." << endl;
    cout << "You have lost this round" << endl;
    losses++; //Number of losses in the game
    uBea++; //Number of bears that have been used by the user
    sBea++; //Number of successful times the computer has used the bear
    total++; //Total number of games in the computer
}
else if(input == 'd'){
    ties++; //Number of ties in the game
    i--; //Decrements the number of games since this did not count
```

```
        uDra++; //Number of times the user has used the dragon
        total++; //Total number of games in the system
      }
      else{
        wins++; //Total number of wins in the game
        uMan++; //Total number of times the user has used man
        sMan++; //Total numver of successful times man was used by the user
        total++; //Total number of games
      }
      aDra++; //Number of times the computer has used the dragon
      break;
  }
  case 5:{
    if(input == 't'){
      cout << "Man has stumbled upon the the tiger to late as it bites off his head from the neck" <<
endl;
      cout << "You have won this round" << endl;
      wins++; //Number of wins in the game
      uTig++; //Number of tigers used by the user
      sTig++; //Number of successful times the user used the tiger
      total++; //Total number of games including ties
    }
    else if(input == 'b'){
      cout << "Mas has seen the bear move onto the honey tree. And running at full speed was foolish,
as the bear tiraded through a bush to chase" << endl;
      cout << "You have won this round" << endl;
      wins++; //Number of wins in the game
      uBea++; //Number of bears used by the user
      sBea++; //Number of successful times in the bear
```

```
         total++; //Total number of games including ties
      }
      else if(input == 'w'){
         cout << "The mighty wolf they call it. Now he is the slave of man" << endl;
         cout << "You have lost this round" << endl;
         losses++; //Number of losses in the game
         uWol++; //Number of wolves used by the user
         sMan2++; //Number of successful times used by the computer
         total++; //Number of games total including ties
      }
      else if(input == 'd'){
        cout << "The dragon was no match for the might of man!" << endl;
        cout << "You have lost this round" << endl;
        losses++; //Number of losses in the game
        uDra++; //Number of dragons used by the user
        sMan2++; //Number of successful times the man uses by the computer
        total++; //Total number of games in the game including ties
      }
      else{
         cout << "Nerds battle it out on animal kingdom to see the supreme ruler of all" << endl;
         cout << "This round is a tie." << endl;
         ties++; //Number of ties in the game
         uMan++; //Number of times  used by the man
         i--; //Decrement the number of games since it is a tie
         total++; //Total number of games in the game
      }
       aMan++; //Number of times the computer has used man
       break;
    }
    }
    }
    unsigned short a[5] = { aTig, aBea, aWol, aDra, aMan}; //Array of the amounts computer uses things
    unsigned short s2[5] = { sTig2, sBea2, sWol2, sDra2, sMan2}; //Array of successful times the computer has used commands
    unsigned short u[5] = { uTig, uBea, uWol, uDra, uMan }; //Array of number of times each command was used by the user
    unsigned short s[5] = { sTig, sBea, sWol, sDra, sMan }; //Array of number of times each command was used by the computer
    //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 = perc(sTig2,aTig); //Calculates the success rate for computer's use of tiger
    perW2 = perc(sWol2,aWol); //Calculates the success rate for computer's use of wolf
    perB2 = perc(sBea2,aBea);//Calculates the success rate for computer's use of bear
    perD2 = perc(sDra2,aDra); //Calculates the success rate for computer's use of dragon
    perM2 = perc(sMan2,aMan);//Calculates the success rate for computer's use of man
    float p2[5] = {perT2, perB2, perW2, perD2, perM2 };
    perT =  perc(sTig,uTig); //Calculates the success rate for the users use of tiger
    perW =  perc(sWol,uWol); //Calculates the success rate for the users use of wolf
    perB =  perc(sBea,uBea); //Calculates the success rate for the users use of bear
    perD =  perc(sDra,uDra); //Calculates the success rate for the users use of dragon
    perM =  perc(sMan,uMan); //Calculates the success rate for the users use of man
    float p[5] = {perT, perB, perW, perD, perM };
    float totalP[2][5];
    for(int c = 0; c < 5; c++){
        totalP[0][c] = p[c];
```

```
      totalP[1][c] = p2[c];
    }
    //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: " <<  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(7) << wins << endl;
    output << "Number of losses for the user: " << setw(5) << losses << endl;
    output << "Number of ties produced this game: " << ties << endl << endl;
     //Display the data for the user
    sort(u,names,5); //Sort the data by greatest to least
    //Displays number of times used throughout the game
    for(int k = 0; k < 5; k++){
      output << "Number of times " << names[k] <<  " was used by user: " << u[k] << endl;
    }
    //Reset the list, so it displays the right names
    names[0] ="Tiger", names[1] = "Bear", names[2] = "Wolf", names[3] = "Dragon", names[4] = "Man";
    //Sort again, this time successful array
    sort(s,names,5);
    output << endl;
    //Displays the number of successful times the computer used a command
    for(int m = 0; m < 5; m++){
       output << "Number of successful times " << names[m] << " was used by user: " << s[m] << endl;
    }
    output << endl;
    //Rest the name list again
    names[0] ="Tiger", names[1] = "Bear", names[2] = "Wolf", names[3] = "Dragon", names[4] = "Man";
    //Sort again, this time the percentages for success

    sort(p,names,5);

    //Displays the success rate of using some of the items

    for(int n = 0; n < 5; n++){

        output << "Success rate of "  << names[n] << " for user: " << setprecision(2) << fixed << showpoint << p[n] << "%" << endl;

    }

    output << endl;

    //Displays results of the computer

    //Reset the name list again

    names[0] ="Tiger", names[1] = "Bear", names[2] = "Wolf", names[3] = "Dragon", names[4] = "Man";

    //Sort the number of times the user used a command in general

    sort(a,names,5);

    //Displays the number of times the computer used a command

    for(int p = 0; p < 5; p++){

        output << "Number of times the computer used " << names[p] << ": " << a[p] << endl;

    }

    output << endl;

    //Reset the names array again

    names[0] ="Tiger", names[1] = "Bear", names[2] = "Wolf", names[3] = "Dragon", names[4] = "Man";

    //Sort again, this time the success rate for the computer

    sort(s2,names,5);

    //Display the successful times the computer uses a command for s2

    for(int q = 0; q < 5; q++){

        output << "Number of successful times " <<  names[q] <<  " was used by computer: " << endl;

    }

    output << endl;

    //Reset the array names again

    names[0] ="Tiger", names[1] = "Bear", names[2] = "Wolf", names[3] = "Dragon", names[4] = "Man";
    //Sort, this time the success rate of the computers choices
    sort(p2,names,5);
    //Displays the success rate of the computers choices
    for(int r = 0; r < 5; r++){
        output << "Success rate of " <<  names[r] << " for computer: " << setprecision(2) << fixed << showpoint << p2[r] << "%" << endl;
    }
    output << endl;
    names[0] ="Tiger", names[1] = "Bear", names[2] = "Wolf", names[3] = "Dragon", names[4] = "Man";
    for(int r = 0; r < 5; r++){
        float num = (totalP[0][r] + totalP[1][r]) / 2;
        output << "Average success rate of both players for " << names[r] << " is " << num << "%" << endl;
    }
    output.close();
    //Display prompt to play again
    cout << "Again? *Note this will wipe out any data in file from previous game*. Enter y for yes" << endl;
    //Input condition
    cin >> cond;
    }while(cond == 'Y' || cond == 'y');
    //Exit Stage Right
    return 0;
}

/*Function sole purpose is to print out the rules of the game*/
void prompt(){
    cout << "Welcome to the game Animal Kingdom. This ever changing game was made" << endl;
    cout << "with the thought of rock paper scissors in mind. For the purpose of this" << endl;
    cout << "game, we will now modify the game to use the tiger a bear, a wolf, a dragon, " << endl;
    cout << "and a man." << endl << endl;
    cout << "Wolves dominates bears and dragons" << endl;
    cout << "Bears dominate Tigers and Man" << endl;
    cout << "Tigers dominate Wolves and Man, " << endl;
    cout << "Dragons conquer Tiger and Bear  " << endl;
    cout << "Man conquers Dragon and Wolf" << endl;
    cout << "You need to compete with the AI 5 times in order to win." << endl;
    cout << "Now lets begin the game!" << endl << endl;
}

/*Function percentage, takes in two float variables
 and returns a float to main that is the percentage of the games
 played.*/

float perc(float suc, float total){
    float num = 0;
        num = (1.0 * suc/total) * 100;
        if(total == 0){
            return total;
        }
        else{
            return num;
        }
}

/*Function perc same as above except for integer variables
 to return a float variable*/
float perc(unsigned short& suc, unsigned short& total){
    float num = 0;
    num = ((1.0 * suc)/total) * 100;
```

```
   if(total == 0){
      return total;
   }
   else{
      return num;
   }
}
/*Function used to sort out an array of numbers, and a parallel array of names
 in order to print out from least to greatest*/
void sort(unsigned short arr[], string names[],int size){
   bool swap;
   unsigned short temp;
   string nameSw;
   do{
      swap = false;
      for(int k = 0; k < (size - 1); k++){
         if(arr[k] < arr[k + 1]){
            temp = arr[k];
            nameSw = names[k];
            arr[k] = arr[k + 1];
            arr[k+1] = temp;
            names[k] = names[k+1];
            names[k+1] = nameSw;
            swap = true;
         }
      }
   }while(swap);
}
/*Overloaded function that takes in floats instead of shorts
```

```
 and uses pointer notation in order to increment the function*/
void sort(float arr[],string names[],short size){
    bool swap;
    float temp;
    string nameSw;
    do{
        swap = false;
        for(int k = 0; k < (size - 1); k++){
            if(*(arr + k) < *(arr + k + 1)){
                temp = *(arr + k);
                nameSw = names[k];
                *(arr + k) = *(arr + k + 1);
                *(arr+k+1) = temp;
                *(names + k) = *(names + k + 1);
                *(names + k+1) = nameSw;
                swap = true;
            }
        }
    }while(swap);
}
```