PROJECT 1

<Pig Dice>

Link to Github Repository for proof of continuous development:

https://github.com/k7ndr4/C-Game-Project

CIS-5 #47589

Kendra Devlahovicht

October 21, 2024

Introduction:

Title: Pig Dice

Pig Dice is a die rolling game.

The goal is to get a certain number of points based on your chosen difficulty level, in a minimum number of turns. Turns will increase by 1 every time that you roll a '1'. You receive points by rolling the die. However, if you roll a '1', then your points will reset back to the last time you chose to manually restart a round.

Summary:

Project Size: ~155 lines.

Number of Variables: ~20 variables

This project contains concepts we have learned up until now, but without the use of any functions or arrays. It has potential to be extended onto next project, however the optimization of such code would likely lead to a minimal amount of lines. So consideration of switching games for project 2 is being considered.

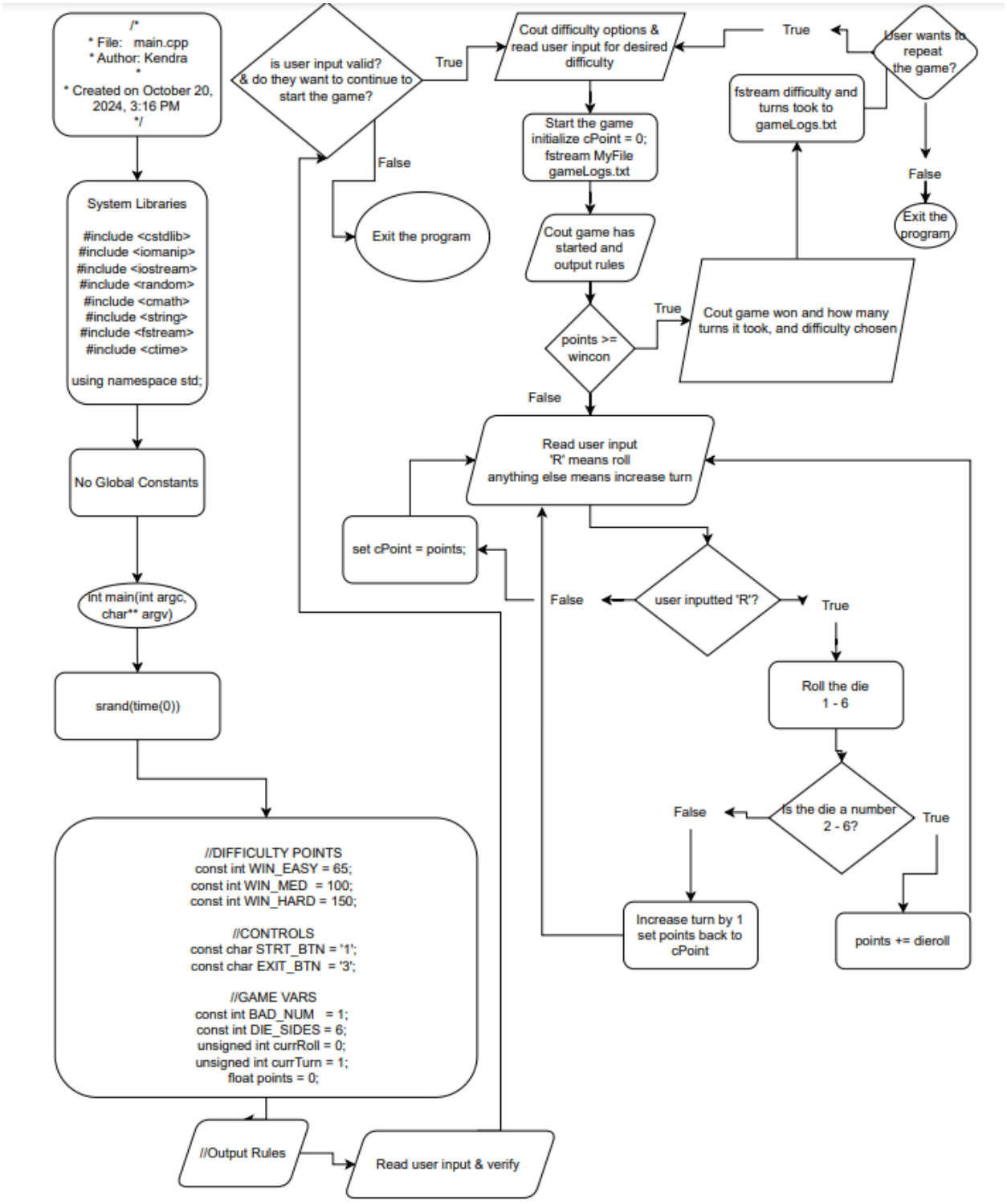
This project took me around a day and a half to complete. I am satisfied with what I could accomplish without the use of arrays/functions, but believe if I had chosen a card game that it would be immensely more difficult to complete without arrays/functions.

This project taught me that the data we use/create can be used in various ways as long as we can keep track of it's uses & check for errors consistently.

Description:

The main purpose of this game is to highlight my proficiency with C++ thus far by using concepts right up until before arrays/functions. It can be used as a form of entertainment for those wanting to see fast they can beat the game as well.

Flow Chart:


/*
* File: main.cpp
* Author: Kendra
*
* Created on October 20, 2024, 3:16 PM
*/
System Libraries
#include <cstdlib>
#include <iomanip>
#include <iostream>
#include <random>
#include <cmath>
#include <string>
#include <fstream>
#include <ctime>
using namespace std;
No Global Constants
int main(int argc, char** argv)
srand(time(0))
//DIFFICULTY POINTS
const int WIN_EASY = 65;
const int WIN_MED = 100;
const int WIN_HARD = 150;
//CONTROLS
const char STRT_BTN = '1';
const char EXIT_BTN = '3';
//GAME VARS
const int BAD_NUM = 1;
const int DIE_SIDES = 6;
unsigned int currRoll = 0;
unsigned int currTurn = 1;
float points = 0;
//Output Rules
Read user input & verify
is user input valid? & do they want to continue to start the game?
True
False
Exit the program
Cout difficulty options & read user input for desired difficulty
Start the game initialize cPoint = 0; fstream MyFile gameLogs.txt
Cout game has started and output rules
points >= wincon
True
Cout game won and how many turns it took, and difficulty chosen
False
Read user input
'R' means roll
anything else means increase turn
user inputted 'R'?
True
False
Roll the die
1 - 6
Is the die a number 2 - 6?
True
False
points += dieroll
Increase turn by 1
set points back to cPoint
set cPoint = points;
fstream difficulty and turns took to gameLogs.txt
User wants to repeat the game?
True
False
Exit the program

Pseudo Code:

Include Libraries & use namespace std;

Int main

Initialize game constants such as control bindings, difficulty point win conditions, & number of sides on the die

Initialize other game variables such as current amount of points, current turn

Initialize an input to keep the game going or not

While(input!= exit){

Cout the rules, & take/verify user input on whether to start the game or to exit the program

//if exit

Exit the program

//if start

Cout to choose a difficulty

Take/verify user input on which difficulty chosen

Store chosen difficulty in a char and corresponding difficulty in a string to output later

Cout difficulty chosen and that game has started

Fstream to a gameLogs.txt that the game has started

Save integer for last checkpoint = 0

do{

cout to roll the die enter ‘R’

//if user enters ‘R’

use math library to randomly roll the die 1-6

cout die roll.

//if die roll == 1

Reset points until last manual reset

Turn++

//if die roll != 1

Points += die roll

//if user enters anything else

Turn++;

Cout their decision and continue with loop

} while(points < winCondition)

Double check if user has points >= winCondition

//if yes

Cout victory, chosen difficulty and amount of turns it took to get there

//if no

Exit program, means something is wrong

Fstream to a gameLogs.txt the number of turns it took and the difficulty used

} loop will repeat to start the game over

Major Variables:

Type	Variable Name	Description	Location
Const Integer	WIN_EASY	Point win con for easy difficulty	Main()
	WIN_MED	Point win con for medium difficulty	
	WIN_HARD	Point win con for hard difficulty	
	BAD_NUM	The number a die has to land on to end the turn	
	DIE_SIDES	Number of sides on the die	
Unsigned Integer	currRoll	The current number the die landed on	
	currTurn	Current turn the user is on	
Integer	cPoint	Last checkpoint (last time user chose to manually end the turn)	While(input!=EXIT_BTN)
Float	points	Keeps track of the users points	
Char	diff	Users chosen difficulty	
	roll	Users decision to roll or end the turn	
Bool	beatGame	Verifies if the user has beaten the game or not	
ofstream	MyFile	Keeps a log of the game files	

Program:

```
/*
 * File:   main.cpp
 * Author: Kendra
 *
 * Created on October 20, 2024, 3:16 PM
 */

#include <cstdlib>
#include <iomanip>
#include <iostream>
#include <random>
#include <cmath>
#include <string>
#include <fstream>
#include <ctime>

using namespace std;

int main(int argc, char** argv) {
   srand(time(0));

   //DIFFICULTY POINTS
   const int WIN_EASY = 65; //NUM OF POINTS TO WIN THE GAME (easy)
   const int WIN_MED  = 100; //NUM OF POINTS TO WIN THE GAME (medium)
   const int WIN_HARD = 150; //NUM OF POINTS TO WIN THE GAME (hard)

   //CONTROLS
```

```
  const char STRT_BTN = '1';
  const char EXIT_BTN  = '3';

  //GAME VARS
  const int BAD_NUM   = 1; //THE SIDE OF THE DIE THAT WILL RESULT IN ENDING YOUR TURN
  const int DIE_SIDES = 6; //AMOUNT OF SIDES ON THE DIE
  unsigned int currRoll = 0; //THE CURRENT SIDE OF THE DIE THAT HAS BEEN ROLLED 'i.e: 5'
  unsigned int currTurn = 1; //THE CURRENT NUMBER OF TURNS THE PLAYER IS ON
  float points = 0;

  //EXPLAIN RULES
  char input;
  while(input != EXIT_BTN){
     cout << "Welcome to the game of Pig Dice!\n" <<
          "To play the game, you roll a die as many times as you'd like to try and reach \n\n" <<
          "EASY: " << WIN_EASY << " points\nMEDIUM: " << WIN_MED << " points\nHARD: " << WIN_HARD << " points\n\nin a minimum amount of turns.\n" <<
          "The catch is, if you roll a " << BAD_NUM << " your points will revert back to the amount of points you had since the last time you chose to end a turn manually, and the amount of turns will increase by 1.\n" <<
          "You may end your turn at any time to save the number of points you have, and try to minimize the amount of turns until you reach the desired amount of points.\n" <<
          "Type " << STRT_BTN <<  " to start the game, or type " << EXIT_BTN << " to exit the program.\n\n";

     //CAPTURE USER INPUT & START THE GAME IF REQUESTED
       cin >> input;
  //   if(input == EXIT_BTN){
  //       cout << "Exiting game...\n";
  //       exit(0);
```

```
//   }else if(input == STRT_BTN){
//      //DO NOTHING, CONTINUE ON WITH PROGRAM
//   }else{
//      cout << "Exiting game...\n";
//      exit(0);
//   }

   //CHOOSE DIFFICULTY
   char diff;
   string diffStr;
   int currCon;
   cout << "Please choose your difficulty...\n" <<
         "Enter 'E' for Easy   (" << WIN_EASY <<  " points)\n" <<
         "Enter 'M' for Medium (" << WIN_MED  <<  " points)\n" <<
         "Enter 'H' for Hard   (" << WIN_HARD <<  " points)\n";
   cin >> diff;

   switch(diff){
      case 'E':
         currCon = WIN_EASY;
         diffStr = "EASY";
         break;

      case 'M':
         currCon = WIN_MED;
         diffStr = "MEDIUM";
         break;

      case 'H':
```

```
      currCon = WIN_HARD;
      diffStr = "HARD";
      break;

   default :
      currCon = WIN_EASY;
      diffStr = "EASY";
      break;
}



//START THE GAME & CREATE A TEXT FILE
ofstream MyFile("gameLogs.txt");
MyFile << "\nGame Session Started\n";

cout << "\nSTARTING GAME on " <<  diffStr << " DIFFICULTY\n\n";

//GAME LOOP
char roll;
int cPoint = 0;
do{
   cout << "Roll the die by inputting 'R', or input anything else to stop your turn.\n";
   cin >> roll;

//IF THE PLAYER CHOOSES TO CONTINUE ROLLING THE DIE
   if(roll == 'R'){

      //ROLL THE DIE & OUTPUT THE RESULT
      currRoll = rand()%DIE_SIDES+1;
```

```
        cout << "You rolled a " << currRoll << '\n';


        //IF THE DIE ROLL LANDED ON '1' THEN THE PLAYER LOSES THEIR POINTS AND THE TURN
# INCREASES
        if(currRoll == 1){
          points = cPoint;
          cout << "Your points have been reverted to " << cPoint << " points. You are currently on
turn: " << currTurn++ << '\n';
          cout << "\nTURN: " << currTurn << '\n';
        }else{
          points += float(currRoll);
          cout << "Your current amount of points are: " << points << '\n';
        }
    //IF THE PLAYER CHOOSES TO MANUALLY END THE TURN
      }else{
        cout << fixed << setprecision(0) <<
            "You have chosen to end your turn. It is currently turn " << currTurn++ <<
            "\nYou have " << points << " amount of points.\n\n";
        cPoint = points;
      }
    }while(points < currCon);


    //ENSURE THAT THE WHILE LOOP BROKE BECAUSE PLAYER REACHED WIN CONDITION
    bool beatGame = points >= currCon ? true : false;
    if(beatGame){
      cout << "Congratulations! You have beat the game on " << diffStr << " difficulty, in " << currTurn
<< " turns!\n\n";
    }


    //RECORD DATA IN FILE
```

```
    const int DATANUM = 2;

    for(int i = 0; i < DATANUM; ++i){
      string output;

      if(i == 0) output = "Difficulty: " + diffStr + "\n";
      else if(i == 1) output = "Amt of Turns: " + to_string(currTurn) + "\n";
      else output = "---------\n\n";

      MyFile << output;
    }

    MyFile.close();
  }

  return 0;
}
```