CRAPS
Dice Game

Richard Rico

CSC 5

Summer 2022

TABLE OF CONTENTS

Introduction

In this project my main objective was to develop a game of craps. I did not want it to be any ordinary game of craps, but in the end I decided to still keep the original game if the player chooses to play it. The original game of craps is simple: You bet if you are going to either a 7 or 11 or bet you will roll a 2, 3, or 12. Now if you roll any other number beside that then that number becomes the point. If then after the point has been established and you roll a 7 you lose. But if you roll the point again one more time then you win. If you roll any other number you keep rolling until you get a 7 or the point. That is very simple but I wanted to do something extra in order to use all the various libraries. I then created “Story Mode”. This story mode lays out a story of your mother needing her hospital bills paid and you deciding to take out a loan. Now that loan has interest and interest is taken every round you play. Now in order to win the game you have to come up with enough money to pay the debt off and save your mother.

HOW CRAPS WORK

Objective

To end up with the most money. Simple right?

Rules

1. All craps games begin with a pass line bet. (Players start off with $1000). A craps player must choose whether they think the dice will land on a combined 7 or 11 to win ('pass the line') or lose by landing on a 2, 3 or 12.

2. The base dealers take all pass bets and add them to the craps table.

3. The shooter starts the craps game with the first roll of the dice, known as the comeout roll.

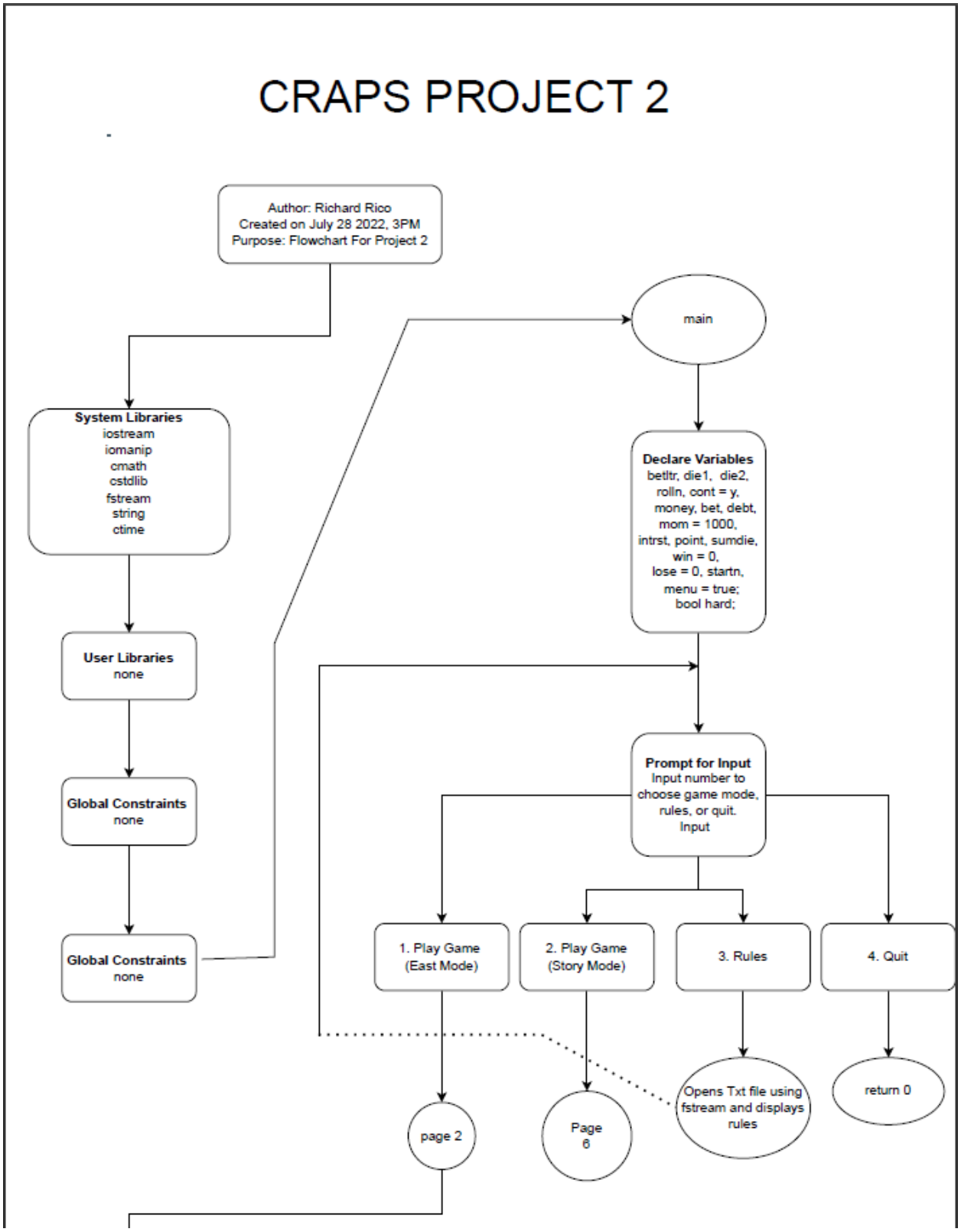
4.If the dice lands on 7 or 11, pass line bettors win their wager. Alternatively, if the dice lands on a combined 2, 3 or 12, don't pass bettors win. Any other numbers the dice land on establishes a 'point' on the craps table, and the game continues.

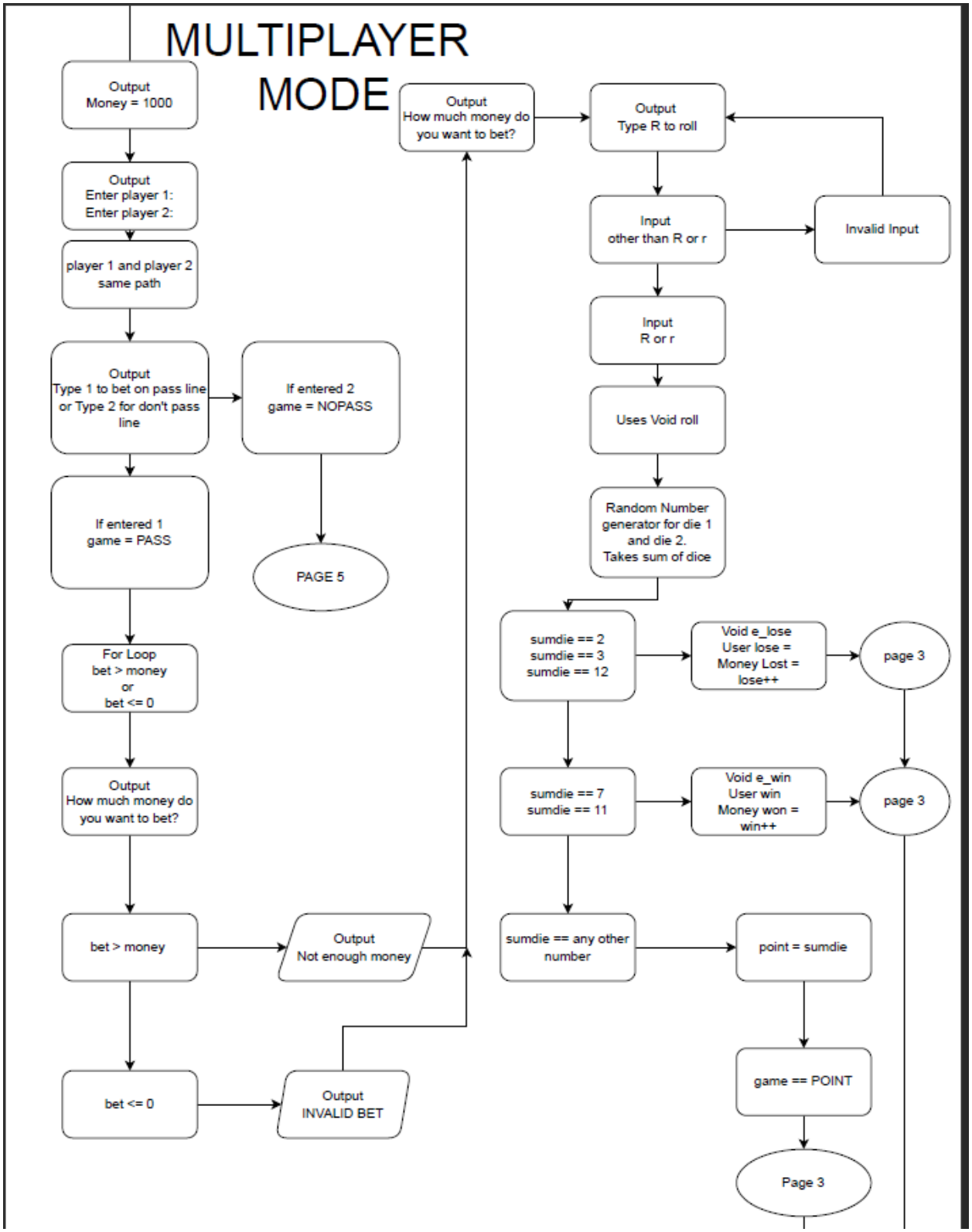
5. Once a point (4, 5, 6, 7, 8, 9, 10) is set on the craps table, craps players can bet on the dice landing the point or landing on 7.

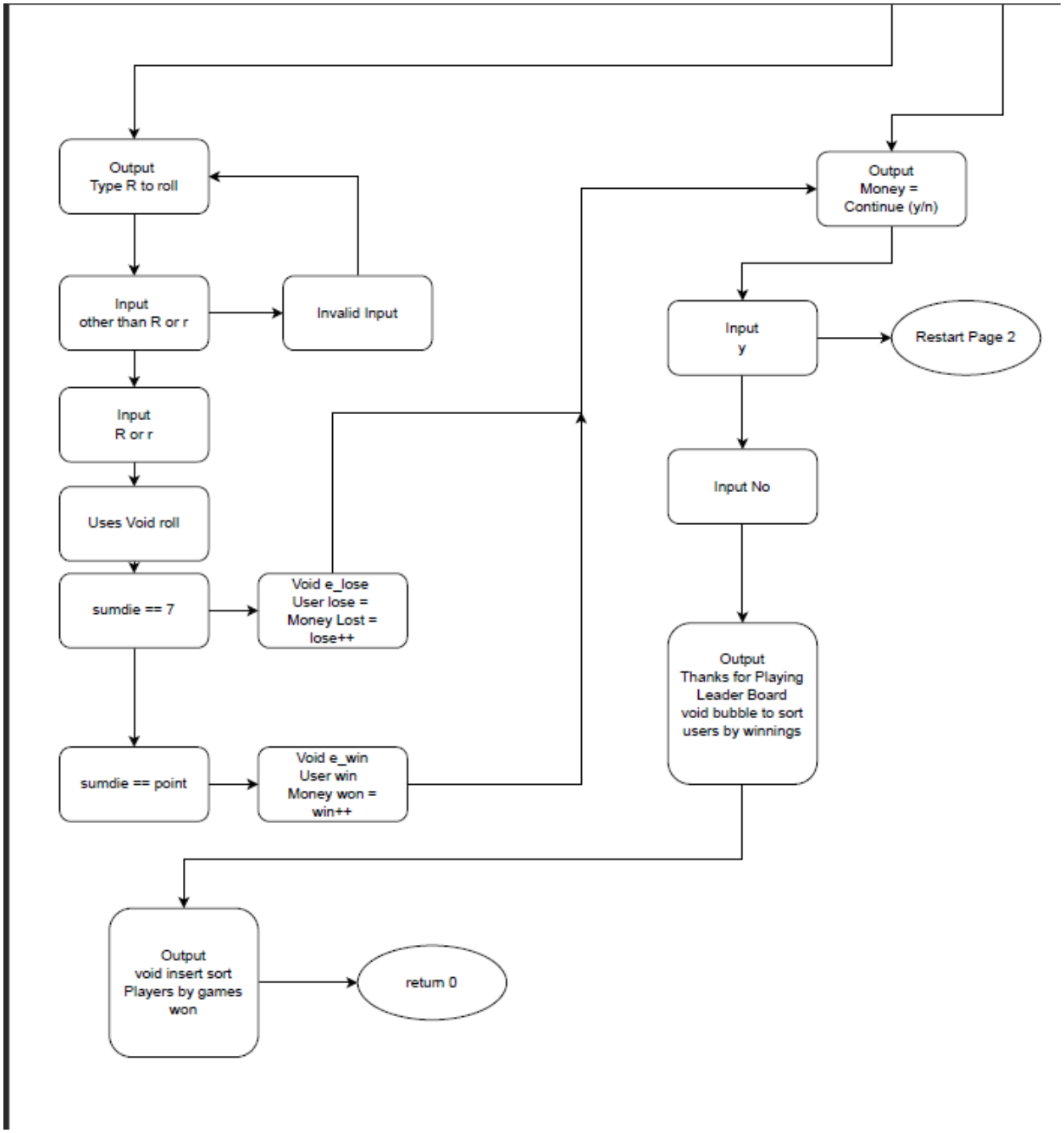
6. The shooter rolls the dice until they land a 7 or the point.

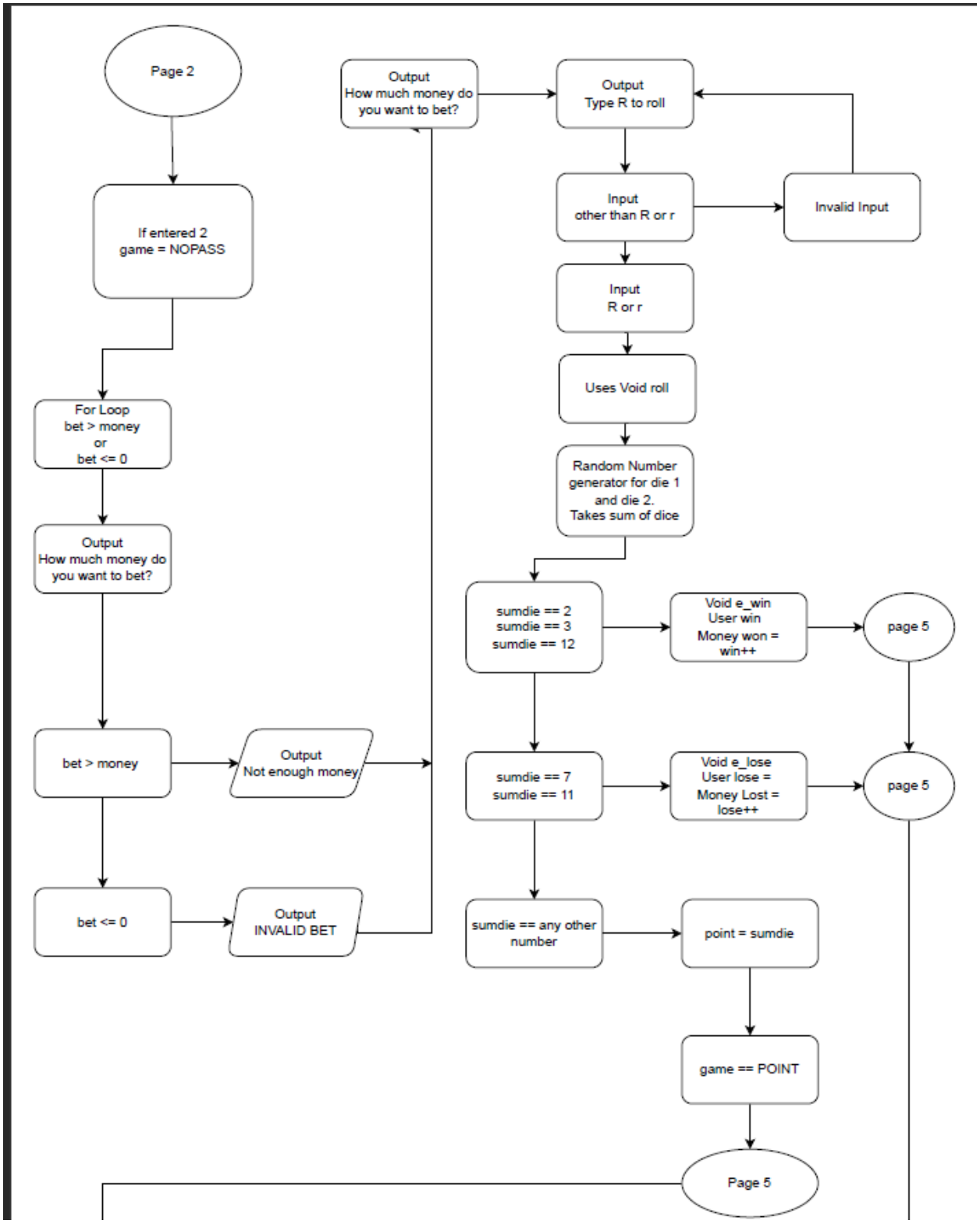
THE LOGIC OF IT ALL

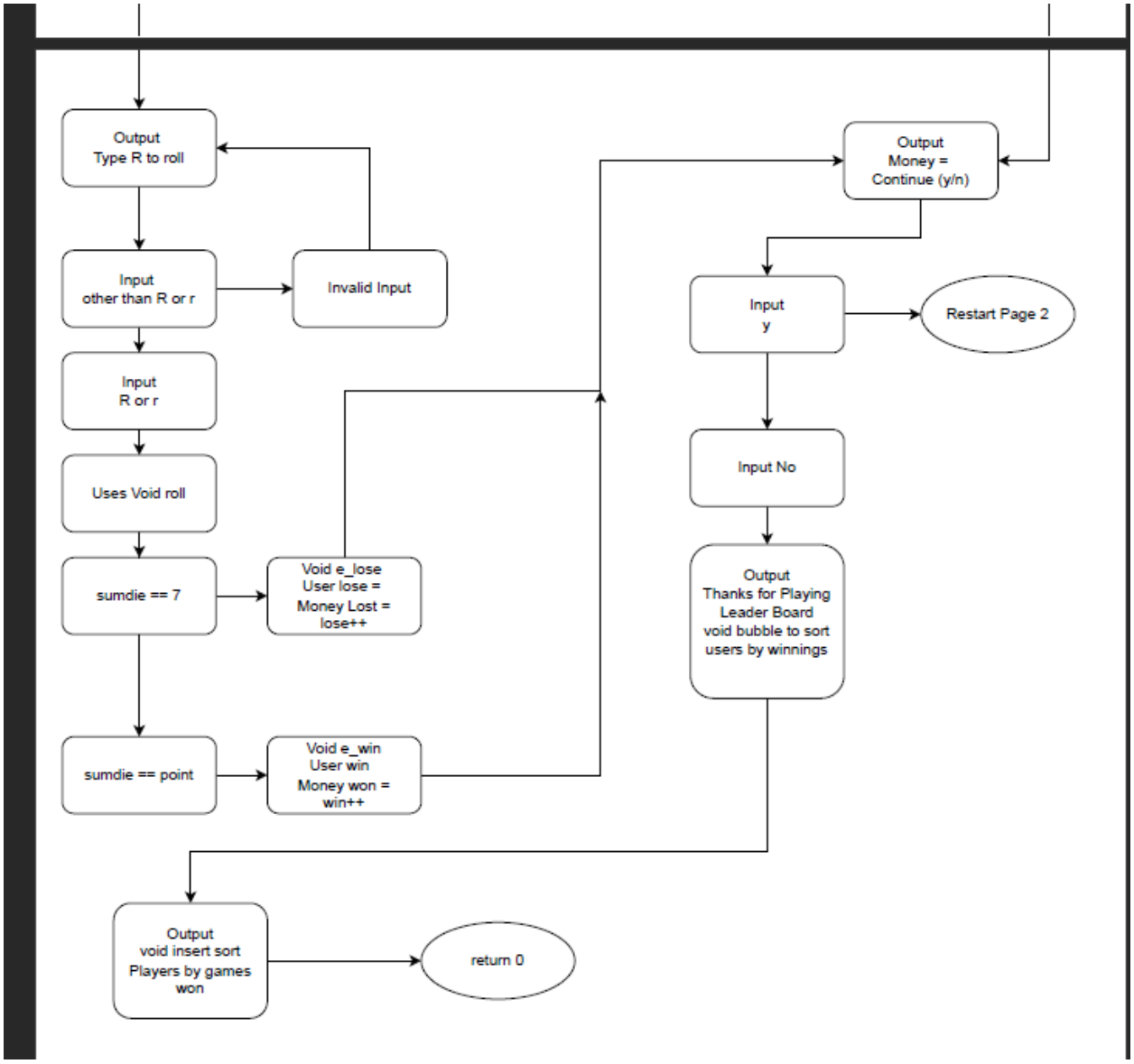
Flowchart

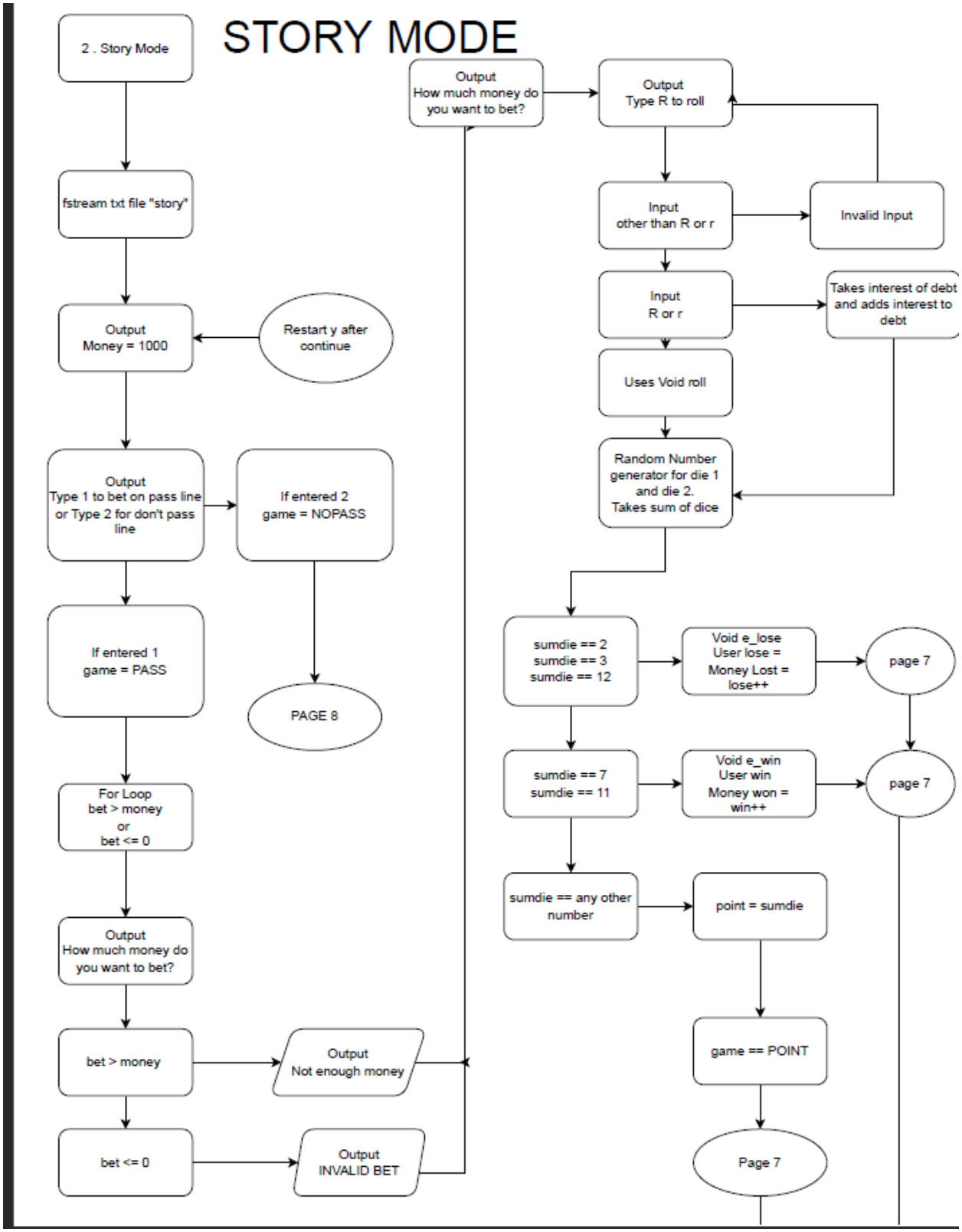


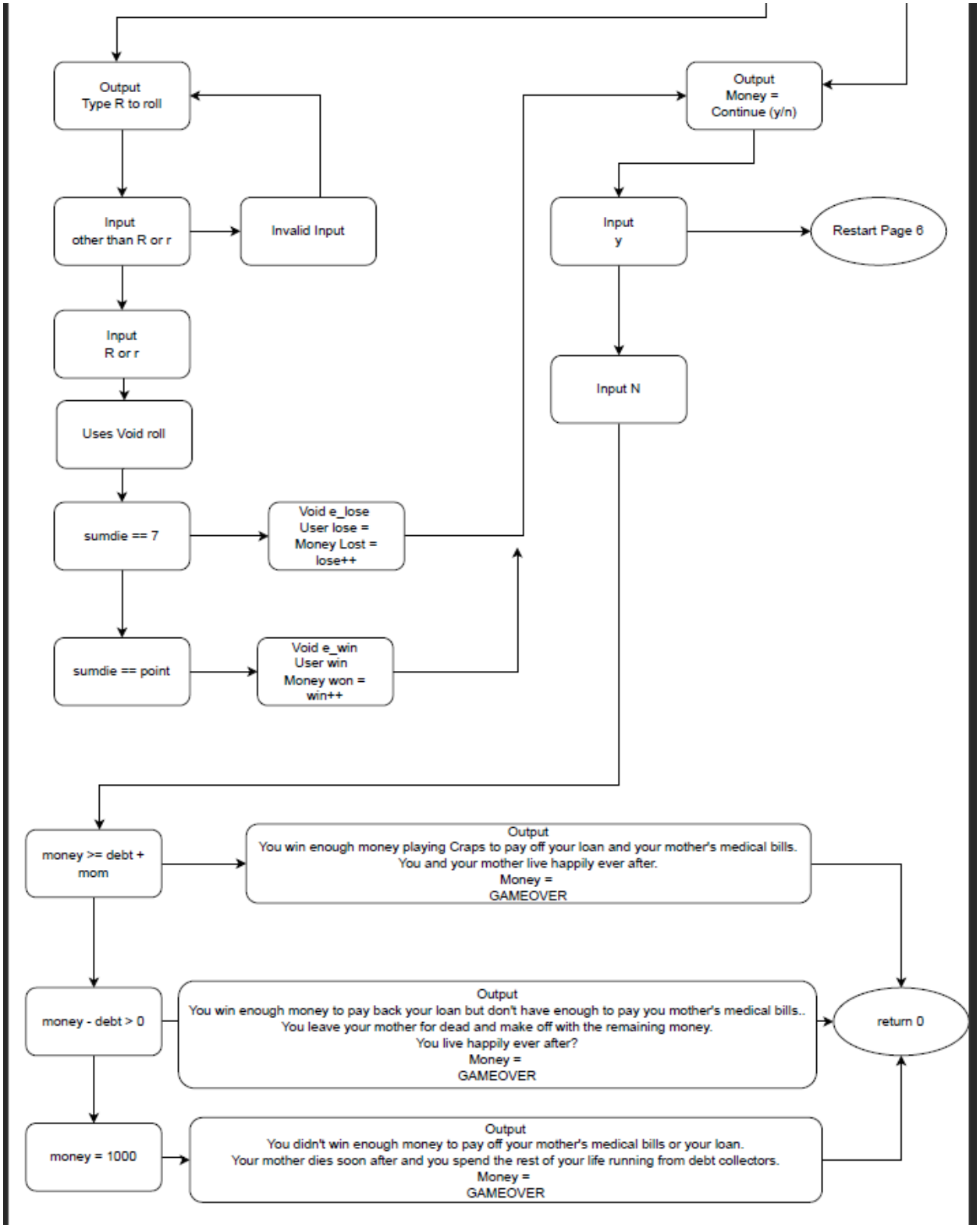

MULTIPLAYER MODE
Output Money = 1000
Output Enter player 1: Enter player 2:
player 1 and player 2 same path
Output Type 1 to bet on pass line or Type 2 for don't pass line
If entered 2 game = NOPASS
PAGE 5
If entered 1 game = PASS
For Loop bet > money or bet <= 0
Output How much money do you want to bet?
bet > money
Output Not enough money
bet <= 0
Output INVALID BET
Output How much money do you want to bet?
Output Type R to roll
Input other than R or r
Invalid Input
Input R or r
Uses Void roll
Random Number generator for die 1 and die 2. Takes sum of dice
sumdie == 2 sumdie == 3 sumdie == 12
Void e_lose User lose = Money Lost = lose++
page 3
sumdie == 7 sumdie == 11
Void e_win User win Money won = win++
page 3
sumdie == any other number
point = sumdie
game == POINT
Page 3

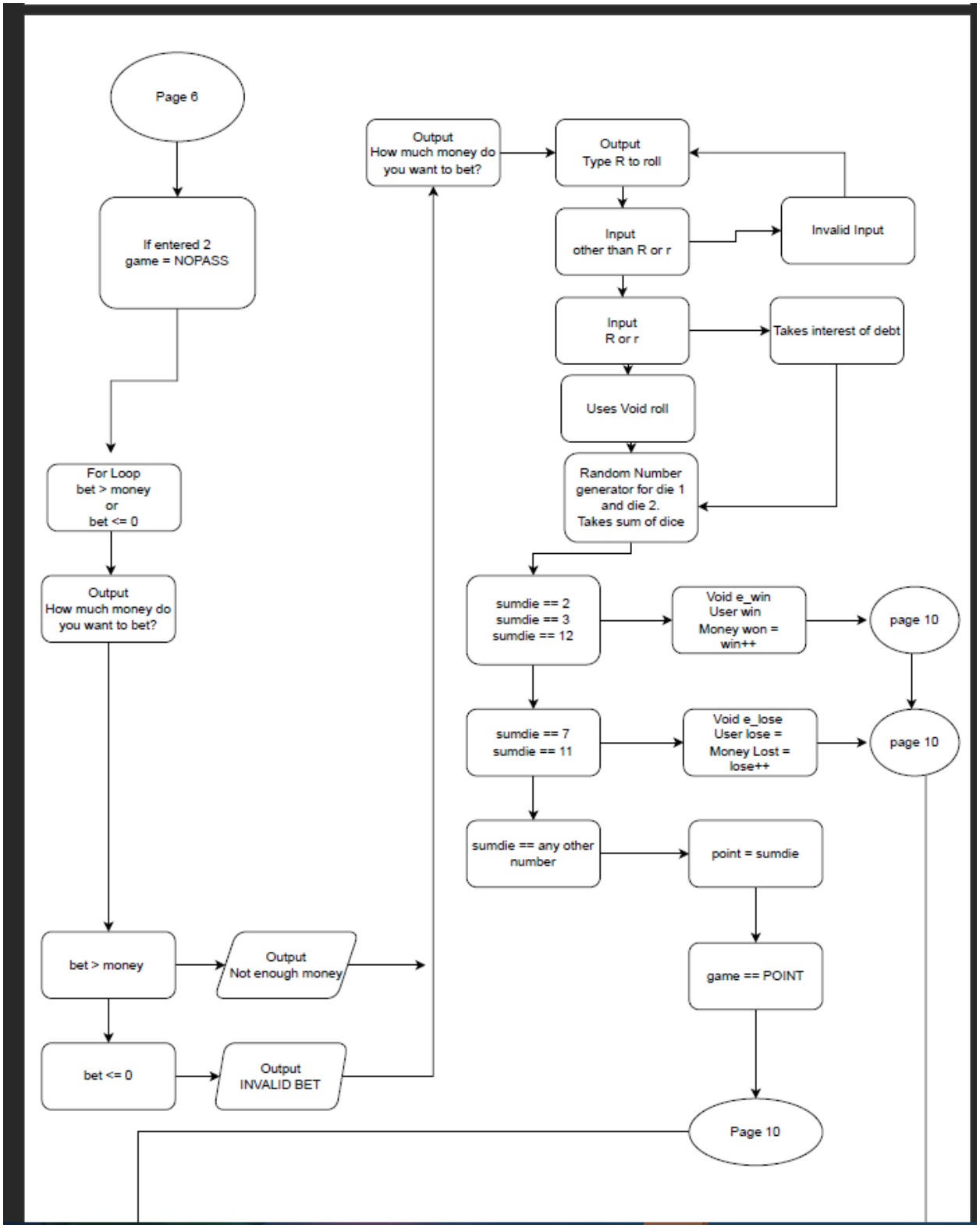

Output
Type R to roll
Input
other than R or r
Invalid Input
Input
R or r
Uses Void roll
sumdie == 7
Void e_lose
User lose =
Money Lost =
lose++
sumdie == point
Void e_win
User win
Money won =
win++
Output
Money =
Continue (y/n)
Input
y
Restart Page 2
Input No
Output
Thanks for Playing
Leader Board
void bubble to sort
users by winnings
Output
void insert sort
Players by games
won
return 0

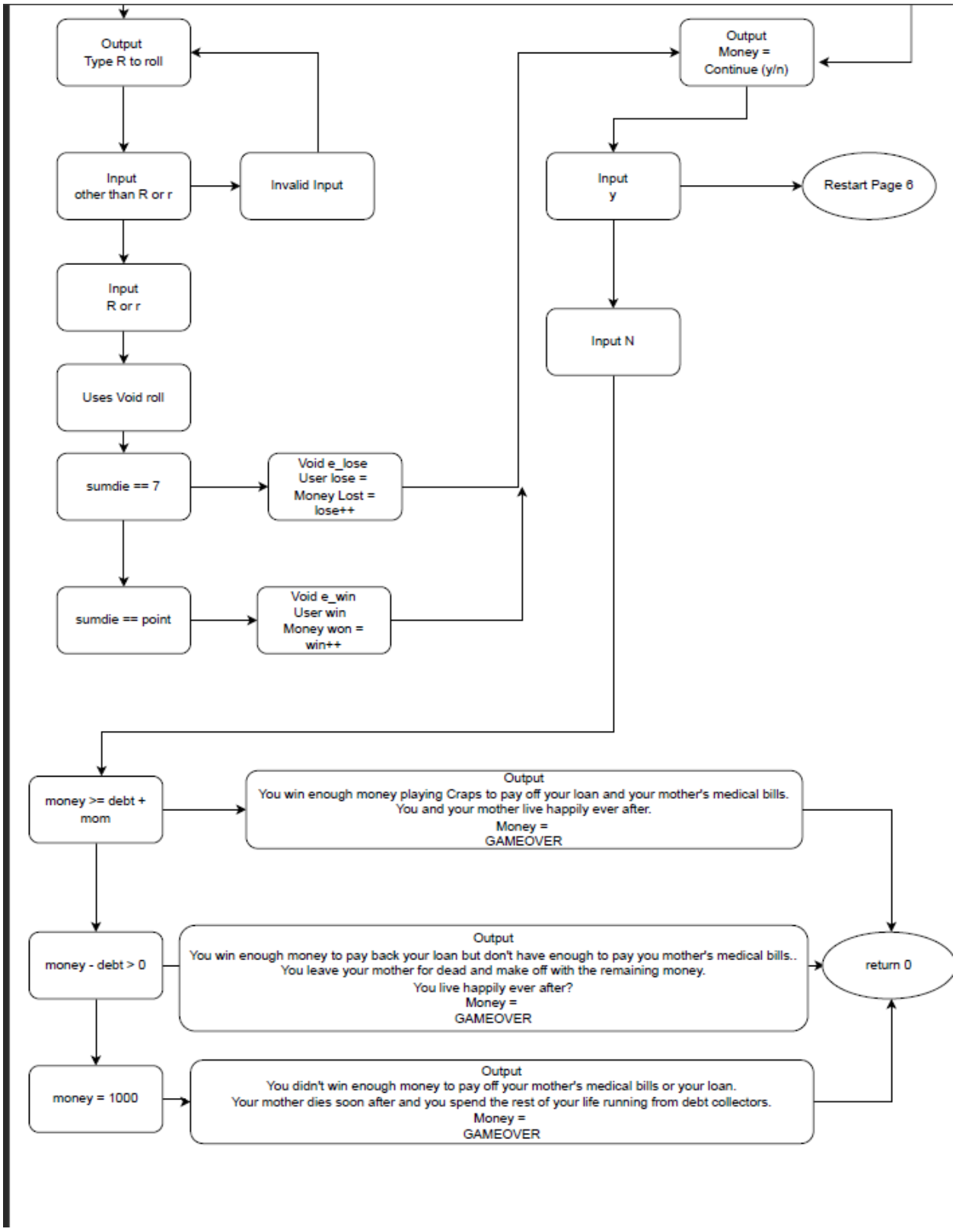

Page 2
If entered 2
game = NOPASS
For Loop
bet > money
or
bet <= 0
Output
How much money do
you want to bet?
bet > money
Output
Not enough money
bet <= 0
Output
INVALID BET
Output
How much money do
you want to bet?
Output
Type R to roll
Input
other than R or r
Invalid Input
Input
R or r
Uses Void roll
Random Number
generator for die 1
and die 2.
Takes sum of dice
sumdie == 2
sumdie == 3
sumdie == 12
Void e_win
User win
Money won =
win++
page 5
sumdie == 7
sumdie == 11
Void e_lose
User lose =
Money Lost =
lose++
page 5
sumdie == any other
number
point = sumdie
game == POINT
Page 5


Output
Type R to roll
Input
other than R or r
Invalid Input
Input
R or r
Uses Void roll
sumdie == 7
Void e_lose
User lose =
Money Lost =
lose++
sumdie == point
Void e_win
User win
Money won =
win++
Output
Money =
Continue (y/n)
Input
y
Restart Page 2
Input No
Output
Thanks for Playing
Leader Board
void bubble to sort
users by winnings
Output
void insert sort
Players by games
won
return 0


STORY MODE
2 . Story Mode
fstream txt file "story"
Output
Money = 1000
Restart y after continue
Output
Type 1 to bet on pass line or Type 2 for don't pass line
If entered 2
game = NOPASS
PAGE 8
If entered 1
game = PASS
For Loop
bet > money
or
bet <= 0
Output
How much money do you want to bet?
bet > money
Output
Not enough money
bet <= 0
Output
INVALID BET
Output
How much money do you want to bet?
Output
Type R to roll
Input
other than R or r
Invalid Input
Input
R or r
Takes interest of debt and adds interest to debt
Uses Void roll
Random Number generator for die 1 and die 2.
Takes sum of dice
sumdie == 2
sumdie == 3
sumdie == 12
Void e_lose
User lose =
Money Lost =
lose++
page 7
sumdie == 7
sumdie == 11
Void e_win
User win
Money won =
win++
page 7
sumdie == any other number
point = sumdie
game == POINT
Page 7


Output
Type R to roll
Input
other than R or r
Invalid Input
Input
R or r
Uses Void roll
sumdie == 7
Void e_lose
User lose =
Money Lost =
lose++
sumdie == point
Void e_win
User win
Money won =
win++
Output
Money =
Continue (y/n)
Input
y
Restart Page 6
Input N
money >= debt +
mom
Output
You win enough money playing Craps to pay off your loan and your mother's medical bills.
You and your mother live happily ever after.
Money =
GAMEOVER
money - debt > 0
Output
You win enough money to pay back your loan but don't have enough to pay you mother's medical bills..
You leave your mother for dead and make off with the remaining money.
You live happily ever after?
Money =
GAMEOVER
money = 1000
Output
You didn't win enough money to pay off your mother's medical bills or your loan.
Your mother dies soon after and you spend the rest of your life running from debt collectors.
Money =
GAMEOVER
return 0


Page 6
If entered 2
game = NOPASS
For Loop
bet > money
or
bet <= 0
Output
How much money do
you want to bet?
bet > money
Output
Not enough money
bet <= 0
Output
INVALID BET
Output
How much money do
you want to bet?
Output
Type R to roll
Input
other than R or r
Invalid Input
Input
R or r
Takes interest of debt
Uses Void roll
Random Number
generator for die 1
and die 2.
Takes sum of dice
sumdie == 2
sumdie == 3
sumdie == 12
Void e_win
User win
Money won =
win++
page 10
sumdie == 7
sumdie == 11
Void e_lose
User lose =
Money Lost =
lose++
page 10
sumdie == any other
number
point = sumdie
game == POINT
Page 10


Output
Type R to roll
Input
other than R or r
Invalid Input
Input
R or r
Uses Void roll
sumdie == 7
Void e_lose
User lose =
Money Lost =
lose++
sumdie == point
Void e_win
User win
Money won =
win++
Output
Money =
Continue (y/n)
Input
y
Restart Page 6
Input N
money >= debt +
mom
Output
You win enough money playing Craps to pay off your loan and your mother's medical bills.
You and your mother live happily ever after.
Money =
GAMEOVER
money - debt > 0
Output
You win enough money to pay back your loan but don't have enough to pay you mother's medical bills..
You leave your mother for dead and make off with the remaining money.
You live happily ever after?
Money =
GAMEOVER
money = 1000
Output
You didn't win enough money to pay off your mother's medical bills or your loan.
Your mother dies soon after and you spend the rest of your life running from debt collectors.
Money =
GAMEOVER
return 0

CONSTRUCTS & CONCEPTS UTILIZED

PROJECT 2 CHECK SHEET

Cross Reference from Project 1

You are to fill-in with where located in code

Chapter	Section	Topic	Where Line #"s	Pts	Notes
2	2	cout			
	3	libraries	6-13	5	iostream, iomanip, cmath, cstdlib, fstream, string, ctime
	4	variables/literals			No variables in global area, failed project!
	5	Identifiers			
	6	Integers	47 - 49	1	
	7	Characters	45	1	
	8	Strings	51	1	
	9	Floats No Doubles	46	1	Using doubles will fail the project, floats OK!
	10	Bools	53	1	
	11	Sizeof *****			
	12	Variables 7 characters or less			All variables <= 7 characters
	13	Scope ***** No Global Variables			
	14	Arithmetic operators			
	15	Comments 20%+	60	2	Model as pseudo code
	16	Named Constants			All Local, only Conversions/Physics/Math in Global area
	17	Programming Style ***** Emulate			Emulate style in book/in class repositiory

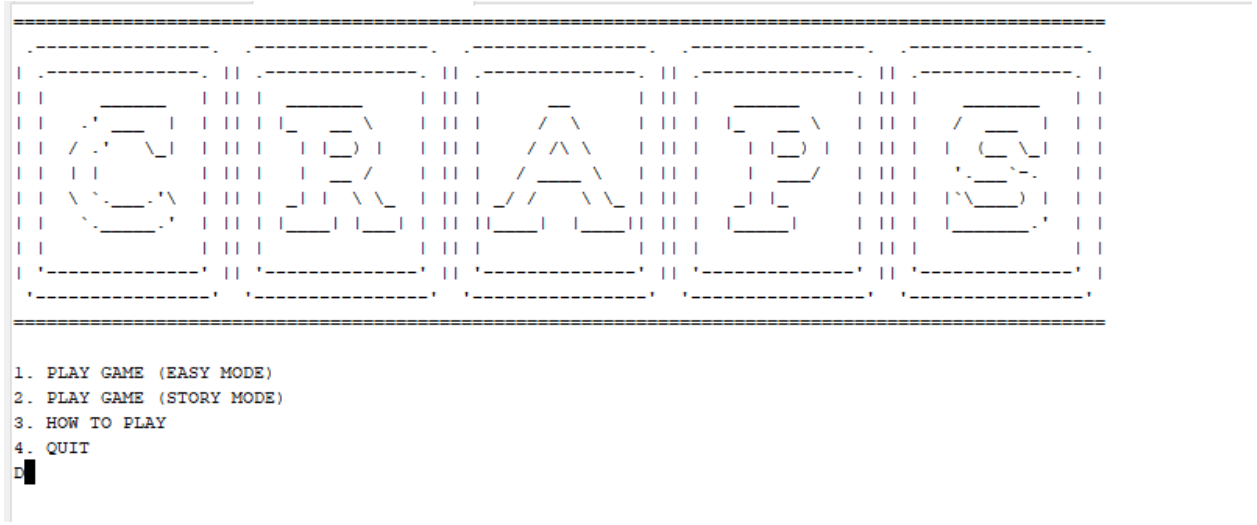
3	1	cin			
	2	Math Expression			
	3	Mixing data types ****			
	4	Overflow/Underflow ****			
	5	Type Casting		1	
	6	Multiple assignment *****			
	7	Formatting output	84	1	
	8	Strings	73	1	
	9	Math Library	8	1	All libraries included have to be used
	10	Hand tracing ******			
4	1	Relational Operators			
	2	if	62	1	Independent if
	4	If-else	62- 67	1	
	5	Nesting	76 -106	1	
	6	If-else-if	91-98	1	
	7	Flags *****			
	8	Logical operators	99	1	
	11	Validating user input		1	
	13	Conditional Operator	91	1	
	14	Switch	148 - 166	1	
5	1	Increment/Decrement	176/587	1	
	2	While	167	1	
	5	Do-while	368 - 428	1	
	6	For loop	56-59	1	
	11	Files input/output both	371-381	2	
	12	No breaks in loops ******			Failed Project if included
****** Not required to show			Total	30	

Cross Reference for Project 2

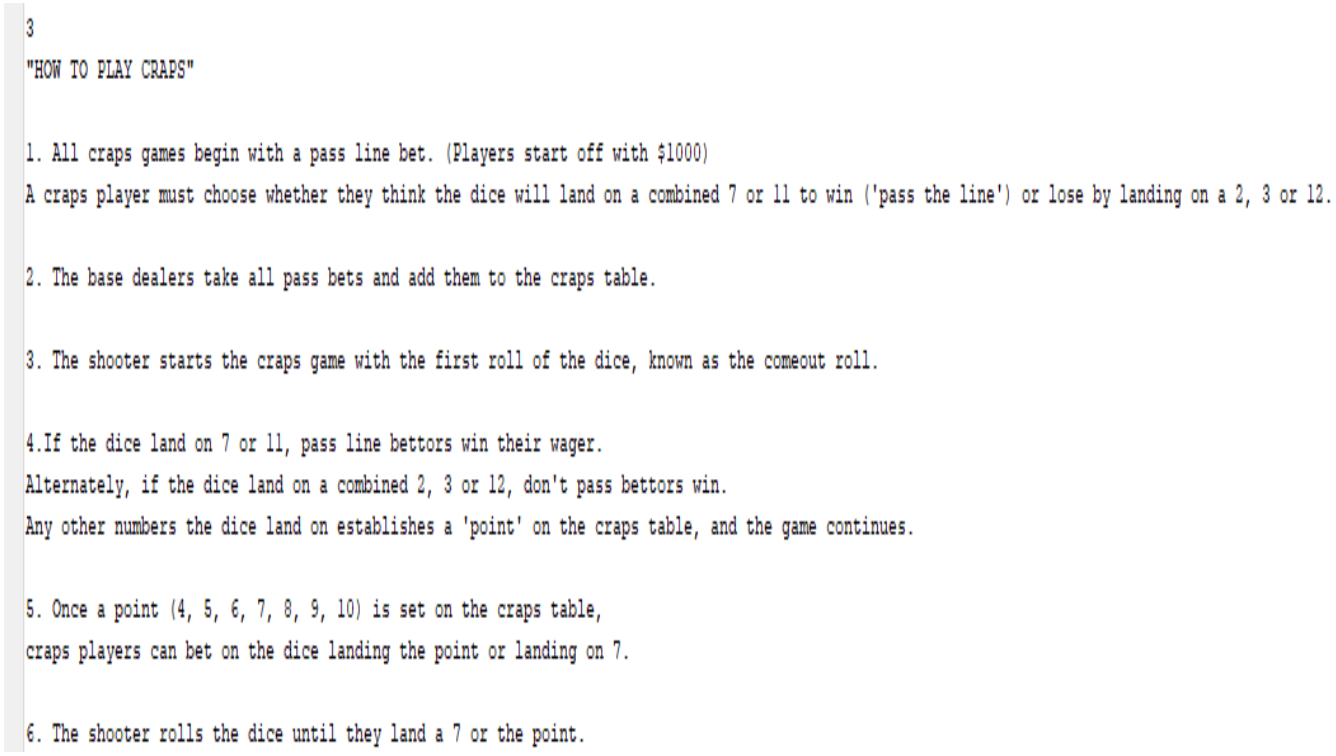
You are to fill-in with where located in code

Chapter	Section	Topic	Where Line #"s	Pts	Notes
6		Functions			
	3	Function Prototypes	26- 38	4	Always use prototypes
	5	Pass by Value	31-32	4	
	8	return	429	4	A value from a function
	9	returning boolean	429	4	
	10	Global Variables		XXX	Do not use global variables -100 pts
	11	static variables	64	4	
	12	defaulted arguments	53	4	
	13	pass by reference	27-30	4	
	14	overloading		5	
	15	exit() function	144	4	
7		Arrays			
	1 to 6	Single Dimensioned Arrays	47	3	
	7	Parallel Arrays	47- 48	2	
	8	Single Dimensioned as Function Arguments	27	2	
	9	2 Dimensioned Arrays	48	2	Emulate style in book/in class repositiory
	12	STL Vectors		2	
		Passing Arrays to and from Functions	27- 31	5	
		Passing Vectors to and from Functions		5	
8		Searching and Sorting Arrays			
	3	Bubble Sort	28	4	
	3	Selection Sort		4	
	1	Linear or Binary Search	29	4	
****** Not required to show			Total	70	Other 30 points from Proj 1 first sheet tab

Proof of Working Game



RULES



MULTIPLAYER MODE

```
1
EASY MODE

(NAME CANNOT CONTAIN SPACES)
Enter Player 1: rib
Enter Player 2: rub

rib MONEY = $1000
Type 1 to bet on pass line or Type 2 for don't pass line
1
How much do you want bet?
500
rub MONEY = $1000
Type 1 to bet on pass line or Type 2 for don't pass line
2
How much do you want bet?
300
TYPE R TO ROLL
r

+-------+
| o     |
|   o   |
|     o |
+-------+
+-------+
| o     |
|   o   |
|     o |
+-------+
SUM OF DICE = 6
POINT = 6

TYPE R TO ROLL
r

+-------+
| o   o |
| o   o |
| o   o |
+-------+
+-------+
| o   o |
| o   o |
| o   o |
+-------+
SUM OF DICE = 12

TYPE R TO ROLL
```

```
TYPE R TO ROLL
r

+-------+
| o     |
|   o   |
|     o |
+-------+
+-------+
| o   o |
| o   o |
| o   o |
+-------+
SUM OF DICE = 9

TYPE R TO ROLL
r

+-------+
| o     |
|       |
|     o |
+-------+
+-------+
| o     |
|       |
|     o |
+-------+
SUM OF DICE = 4

TYPE R TO ROLL
r

+-------+
| o   o |
|       |
| o   o |
+-------+
+-------+
| o   o |
|       |
| o   o |
+-------+
SUM OF DICE = 8

TYPE R TO ROLL
r
```

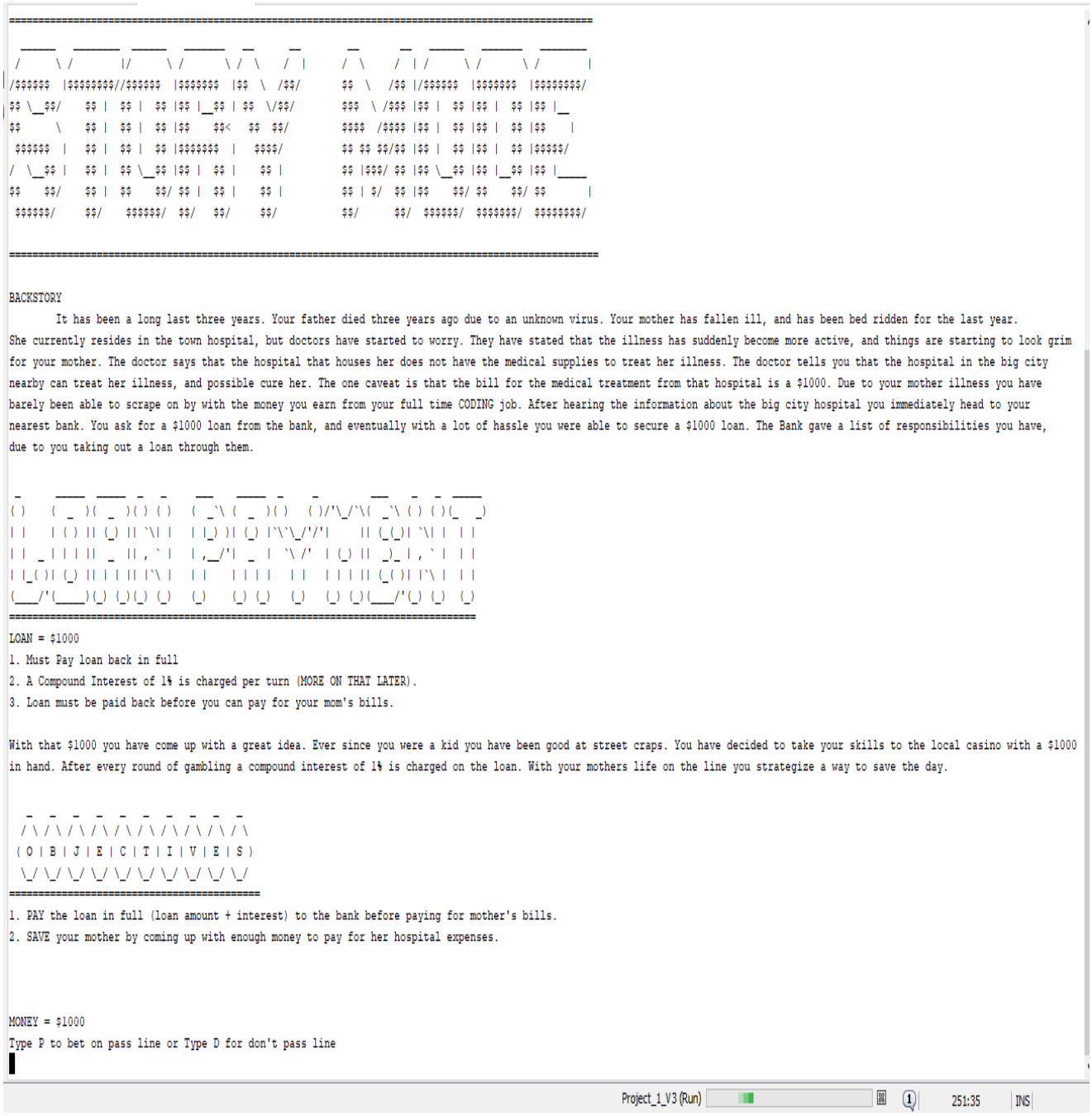
```
+-------+
|       |
|   o   |
|       |
+-------+
+-------+
| o     |
|   o   |
|     o |
+-------+
SUM OF DICE = 4

TYPE R TO ROLL
r

+-------+
| o   o |
|       |
| o   o |
+-------+
+-------+
| o     |
|   o   |
|     o |
+-------+
SUM OF DICE = 7

rib LOST
MONEY LOST = $500
rub LOST
MONEY LOST = $300
rib MONEY = $500.00
rub MONEY = $700.00

CONTINUE?(y/n)
n
THANKS FOR PLAYING!
LEADERBOARD
```

```
   FINAL BALANCE
Player                  MONEY
rub                       $700
rib                       $500
Player                  GAMES WON
rub                       0
rib                       0
```

STORY MODE

Story txt

```
================================================================================================

 _____   _______  _____   _______  __      __       __      __  _____   _______   _______
 /     \ /       |/     \ /       \ /  \    /  |     /  \    /  | /     \ /       \ /       |
/$$$$$  |$$$$$$$$//$$$$$  |$$$$$$$  |$$  \  /$$/      $$  \   /$$ |/$$$$$  |$$$$$$$  |$$$$$$$$/
$$ \__$$/    $$ |  $$ |  $$ |$$ |__$$ | $$  \/$$/       $$$  \ /$$$ |$$ |  $$ |$$ |  $$ |$$ |__
$$      \    $$ |  $$ |  $$ |$$    $$<   $$  $$/        $$$$  /$$$$ |$$ |  $$ |$$ |  $$ |$$    |
 $$$$$$  |   $$ |  $$ |  $$ |$$$$$$$  |   $$$$/         $$ $$ $$/$$ |$$ |  $$ |$$ |  $$ |$$$$$/
/  \__$$ |   $$ |  $$ \__$$ |$$ |  $$ |    $$ |         $$ |$$$/ $$ |$$ \__$$ |$$ |__$$ |$$ |_____
$$    $$/    $$ |  $$    $$/ $$ |  $$ |    $$ |         $$ | $/  $$ |$$    $$/ $$    $$/ $$       |
 $$$$$$/     $$/    $$$$$$/  $$/   $$/     $$/          $$/      $$/  $$$$$$/  $$$$$$$/  $$$$$$$$/

================================================================================================

BACKSTORY
        It has been a long last three years. Your father died three years ago due to an unknown virus. Your mother has fallen ill, and has been bed ridden for the last year.
She currently resides in the town hospital, but doctors have started to worry. They have stated that the illness has suddenly become more active, and things are starting to look grim
for your mother. The doctor says that the hospital that houses her does not have the medical supplies to treat her illness. The doctor tells you that the hospital in the big city
nearby can treat her illness, and possible cure her. The one caveat is that the bill for the medical treatment from that hospital is a $1000. Due to your mother illness you have
barely been able to scrape on by with the money you earn from your full time CODING job. After hearing the information about the big city hospital you immediately head to your
nearest bank. You ask for a $1000 loan from the bank, and eventually with a lot of hassle you were able to secure a $1000 loan. The Bank gave a list of responsibilities you have,
due to you taking out a loan through them.


 _      ____  ____  _  _     ___   ____  _     _          ___   _  _  ____
( )    (  _ )( _  )( ) ( )   (  _`\ (  _ )( )   ( )/'\_/`\( _`\ ( ) ( )(_  _)
| |    | ( ) || (_) || `\| |   | |_) )| (_) |`\`\_/'/'|     || (_(_)| `\| |  | |
| |  _ | | | ||  _  || , ` |   | ,__/'|  _  | `\ /'  | (_) ||  _)_ | , ` |  | |
| |_( )| (_) || | | || |`\ |   | |    | | | |  | |   | | | || (_( )| |`\ |  | |
(____/'(____)(_) (_)(_) (_)   (_)    (_) (_)  (_)   (_) (_)(____/'(_) (_)  (_)
================================================================================
LOAN = $1000
1. Must Pay loan back in full
2. A Compound Interest of 1% is charged per turn (MORE ON THAT LATER).
3. Loan must be paid back before you can pay for your mom's bills.

With that $1000 you have come up with a great idea. Ever since you were a kid you have been good at street craps. You have decided to take your skills to the local casino with a $1000
in hand. After every round of gambling a compound interest of 1% is charged on the loan. With your mothers life on the line you strategize a way to save the day.

  _   _   _   _   _   _   _   _   _
 / \ / \ / \ / \ / \ / \ / \ / \ / \
( O | B | J | E | C | T | I | V | E | S )
 \_/ \_/ \_/ \_/ \_/ \_/ \_/ \_/ \_/ \_/
========================================
1. PAY the loan in full (loan amount + interest) to the bank before paying for mother's bills.
2. SAVE your mother by coming up with enough money to pay for her hospital expenses.



MONEY = $1000
Type P to bet on pass line or Type D for don't pass line
```



One game played and lost

```
MONEY = $1000
Type P to bet on pass line or Type D for don't pass line
P
How much do you want bet?
500
TYPE R TO ROLL
r

SUM OF DICE = 10
POINT = 10

TYPE R TO ROLL
r

SUM OF DICE = 12

TYPE R TO ROLL
r

SUM OF DICE = 5

TYPE R TO ROLL
r

SUM OF DICE = 9

TYPE R TO ROLL
r

SUM OF DICE = 4

TYPE R TO ROLL
r

SUM OF DICE = 7

YOU LOSE

MONEY LOST = $500
Current Loan Amount = $1010.00
Mother's Medical Bills = $1000.00
MONEY = $500.00
```

```
CONTINUE?(y/n)
n



You didn't win enough money to pay off your mother's medical bills or your loan.
Your mother dies soon after and you spend the rest of your life running from debt collectors.
MONEY = $-510.00
GAMEOVER
RUN SUCCESSFUL (total time: 2m 14s)
```

Multiple games play to win

```
MONEY = $1000
Type P to bet on pass line or Type D for don't pass line
p
How much do you want bet?
500
TYPE R TO ROLL
r

SUM OF DICE = 11
PASS THE LINE
YOU WIN

MONEY WON = $1000
Current Loan Amount = $1010.00
Mother's Medical Bills = $1000.00
MONEY = $1500.00

CONTINUE?(y/n)
y



MONEY = $1500.00
Type P to bet on pass line or Type D for don't pass line
p
How much do you want bet?
500
TYPE R TO ROLL
r

SUM OF DICE = 5
POINT = 5

TYPE R TO ROLL
r

SUM OF DICE = 4
```

```
r

SUM OF DICE = 5

YOU WIN

MONEY WON = $1000.00
Current Loan Amount = $1030.30
Mother's Medical Bills = $1000.00
MONEY = $2000.00

CONTINUE?(y/n)
y



MONEY = $2000.00
Type P to bet on pass line or Type D for don't pass line
p
How much do you want bet?
100
TYPE R TO ROLL
r

SUM OF DICE = 8
POINT = 8

TYPE R TO ROLL
r

SUM OF DICE = 8

YOU WIN

MONEY WON = $200.00
Current Loan Amount = $1061.52
Mother's Medical Bills = $1000.00
MONEY = $2100.00

CONTINUE?(y/n)
n'



You win enough money playing Craps to pay off your loan and your mother's medical bills.
You and your mother live happily ever after.
MONEY = $38.48
GAME OVER
```

Program

```
/*
 * File:   main.cpp
 * Author: Richard R. Rico
 * Created on July 27, 2022, 2:00 PM
 * Purpose: Project 2
 */

//System Libraries
#include <iostream>
#include <iomanip>
#include <cmath>
#include <cstdlib>
#include <fstream>
#include <string>
#include <ctime>
#include <stdlib.h>

using std::cout;
using std::cin;
using std::endl;
using std::fstream;
using std::string;
using std::setprecision;
using std::fixed;
using std::pow;
using std::getline;

//Function Pre-Intialization
bool menu();
void roll(int *dice);
void bubble(float *money, string *users);
void insert(int *win, string *users);
void e_lose(int, string);
void e_win(int, string);
void die1();
void die2();
void die3();
void die4();
void die5();
```

```
void die6();

//Execution Begins Here
int main(int argc, char** argv) {
    //Initialize the Random Number Seed
    srand(static_cast<unsigned int>(time(NULL)));
    //Declare Variables/Initialize Variables
    char betltr, die1, die2, rolln, cont = 'y';
    float debt, mom;
    int  bet, point, sumdie, win[2] = { 0, 0 }, lose[2] = { 0, 0 };
    int dice[2];
    int bets[2][2];
    float money[2];
    string users[2];
    enum Status { POINT, PASS, NOPASS, NPOINT };
    bool hard, zero = false;
    Status game;
    //initializer
    for (int i = 0; i < 2; i++)
    {
        money[i] = 1000;
    }
    //PLAY GAME
    hard = menu();
   if(hard == true )
   {
      mom = 1000;
      debt = money[0];
   }
   else
   {
      cout << "(NAME CANNOT CONTAIN SPACES)" << endl;
      for (int i = 0; i < 2; i++)
      {
         cout << "Enter Player " << i + 1 << ": ";
         cin >> users[i];
      }
   }
    do
    {
```

```
cout << endl;
if (hard == false)
{
    for (int i = 0; i < 2; i++)
    {
        cout << users[i] << " MONEY = $" << money[i] << endl;
        cout << "Type 1 to bet on pass line or Type 2 for don't pass line" << endl;
        cin >> bets[0][i];
        bets[1][i] = 0;
        for (int j = 0; bets[1][i] > money[i] || bets[1][i] <= 0; j++)
        {
            cout << "How much do you want bet?" << endl;
            cin >> bets[1][i];
            if (bets[1][i] > money[i])
            {
                cout << "NOT ENOUGH MONEY" << endl;
            }
            else if (bets[1][i] <= 0)
            {
                cout << "INVALID BET" << endl;
            }
            if (j > 0 && (bets[1][i] < money[i] && bets[1][i] > 0))
            {
                cout << "BET INCORRECTLY " << j << " TIMES" << endl << endl;
            }
        }
        money[i] -= bets[1][i];
    }
}
else
{
    cout << "MONEY = $" << money[0] << endl;
    cout << "Type 1 to bet on pass line or Type 2 for don't pass line" << endl;
    cin >> bets[0][0];
    bet = 0;
    for (int i = 0; bet > money[0] || bet <= 0; i++)
    {
        cout << "How much do you want bet?" << endl;
        cin >> bet;
        if (bet > money[0]) {
```

```
            cout << "NOT ENOUGH MONEY" << endl;
        } else if (bet <= 0) {
            cout << "INVALID BET" << endl;
        }
        if (i > 0 && (bet < money[0] && bet > 0)) {
            cout << "BET INCORRECTLY " << i << " TIMES" << endl << endl;
        }
    }
    money[0] -= bet;
}

cout << "TYPE R TO ROLL" << endl;
cin >> rolln;
cout << endl;
if (rolln == 'r' || rolln == 'R')
{
    sumdie = 0;
    roll(dice);
    for (int i = 0; i < 2; i++)
    {
        sumdie += dice[i];
    }
    cout << "SUM OF DICE = " << sumdie << endl;
}
else
{
    cout << "INVAILD INPUT";
    exit(0);
}
//Interpret Dice Roll as Pass the line, not Pass The Line, or assign point
switch (sumdie)
{
case 2:
case 3:
case 12:
    game = NOPASS;
    cout << "DID NOT PASS THE LINE" << endl;
    break;
case 7:
case 11:
```

```
    game = PASS;
    cout << "PASS THE LINE" << endl;
    break;
default:
    game = POINT;
    point = sumdie;
    cout << "POINT = " << point << endl << endl;
    break;
}
while (game == POINT)
{
    cout << "TYPE R TO ROLL" << endl;
    cin >> rolln;
    cout << endl;
    if (rolln == 'r' || rolln == 'R')
    {
        sumdie = 0;
        roll(dice);
        for (int i = 0; i < 2; i++)
        {
            sumdie += dice[i];
        }
        cout << "SUM OF DICE = " << sumdie << endl << endl;
    }
    else
    {
        cout << "INVAILD INPUT";
        exit(0);
    }
    //Test Case sum of dice is equal to point
    if (sumdie == point)
    {
        if (hard)
        {
          money[0] += (2 * bet);
          e_win(bet, "YOU");
          win[0]++;
          game = NPOINT;
        }
        else
```

```
        {
            for (int i = 0; i < 2; i++)
            {
                money[i] += (2 * bets[1][i]);
                e_win(bets[1][i], users[i]);
                win[i]++;
                game = NPOINT;
            }
        }
    }
    else if (sumdie == 7)
    {
        if (hard)
        {
            e_lose(bet, "YOU");
            lose[0]++;
            game = NPOINT;
        }
        else
        {
            for (int i = 0; i < 2; i++)
            {
                e_lose(bets[1][i], users[i]);
                lose[i]++;
                game = NPOINT;
            }
        }
    }
}
//Did not pass the line and bet on outcome
if (hard)   //STORY MODE
{
    if (game == NOPASS && (bets[0][0] == 2))
    {
        money[0] += (2 * bet);
        e_win(bet, "YOU");
        win[0]++;
    }               //Pass the line and bet on outcome
    else if (game == PASS && (bets[0][0] == 1))
    {
```

```
        money[0] += (2 * bet);
        e_win(bet, "YOU");
        win[0]++;
    }              //Outcome bet on did not occur
    else if (game == PASS || game == NOPASS)
    {
        e_lose(bet, "YOU");
        lose[0]++;
    }
}
else    //MULTIPLAYER MODE
{
   for (int i = 0; i < 2; i++)
   {
        if (game == NOPASS && (bets[0][i] == 2 ))
        {
            money[i] += (2 * bets[1][i]);
            e_win(bets[1][i], users[i]);
            win[i]++;
        }                  //Pass the line and bet on outcome
        else if (game == PASS && (bets[0][i] == 1))
        {
            money[i] += (2 * bets[1][i]);
            e_win(bets[1][i], users[i]);
            win[i]++;
        }                  //Outcome bet on did not occur
        else if (game == PASS || game == NOPASS)
        {
            e_lose(bets[1][i], users[i]);
            lose[i]++;
        }
   }
}
if (hard)
{
    debt *= pow((1 + .01), (win[0] + lose[0]));
    cout << "Current Loan Amount = $" << fixed << setprecision(2) << debt << endl;
    cout << "Mother's Medical Bills = $" << fixed << setprecision(2) << mom << endl;
}
//User input to end or continue game
```

```
        if (hard)
        {
            cout << "MONEY = $" << money[0] << endl;
        }
        else
        {
            for (int i = 0; i < 2; i++)
            {
                cout << users[i] << " MONEY = $" << money[i] << endl;
            }
        }
        cout << endl << "CONTINUE?(y/n)" << endl;
        cin >> cont;
        if (hard == false && (bets[1][0] == 0 || bets[1][1] == 0))
        {
            zero = true;
        }
    }
    while ( zero == false && (cont == 'y' || cont == 'Y'));
    //Display Results
    if (hard == false)
    {
        cout << "THANKS FOR PLAYING!" << endl;
        cout << "LEADERBOARD" << endl << endl;
        cout << "   FINAL BALANCE" << endl;
        bubble(money, users);
        cout << "Player               MONEY" << endl;
        for (int i = 0; i < 2; i++)
        {
            cout << users[i] << "                    $" << fixed << setprecision(0) << money[i] <<
endl;
        }
        cout << "Player               GAMES WON" << endl;
        insert(win, users);
        for (int i = 0; i < 2; i++)
        {
            cout << users[i] << "                    " << win[i] << endl;
        }
            //Exit stage right
        return 0;
```

```
   }
   else if (hard)
   {
      if (money[0] >= (debt + mom))
      {
         cout << "You win enough money playing Craps to pay off your loan and your
mother's medical bills." << endl;
         cout << "You and your mother live happily ever after." << endl;
         cout << "MONEY = $" << fixed << setprecision(2) << money[0] - (debt + mom)
<< endl;
         cout << "GAME OVER";
         return 0;
      }
      else
      {
         if (money[0] - debt > 0)
         {
            cout << "You win enough money to pay back your loan but don't have enough
to pay you mother's medical bills." << endl;
            cout << "You leave your mother for dead and make off with the remaining
money." << endl;
            cout << "You live happily ever after?" << endl;
            cout << "MONEY = $" << fixed << setprecision(2) << money[0] - debt << endl;
            cout << "GAME OVER";
            return 0;
         }
         else
         {
            cout << "You didn't win enough money to pay off your mother's medical bills or
your loan." << endl;
            cout << "Your mother dies soon after and you spend the rest of your life
running from debt collectors." << endl;
            cout << "MONEY = $" << fixed << setprecision(2) << money[0] - debt << endl;
            cout << "GAMEOVER";
            return 0;
         }
      }
   }
}
bool menu()
```

```
{
    bool hard, menu = true;
    int start;
    do
    {
        //Read Main Menu from txt file
        fstream main_t;
        main_t.open("C:/Users/woodc/OneDrive/Desktop/Project_2/Craps.txt", std::ios::in);
        if (main_t.is_open())
        {
            string line;
            while (std::getline(main_t, line))
            {
                cout << line << endl;
            }
            main_t.close();
        }
        //User Input to output selection
        cin >> start;
        switch (start)
        {
            //Start the Game
        case 1:
            cout << "EASY MODE" << endl << endl;
            menu = false;
            hard = false;
            break;
            //Rules read from text file
        case 2:
        {
            fstream story;
            story.open("C:/Users/woodc/OneDrive/Desktop/Project_2/Story.txt", std::ios::in);
            if (story.is_open())
            {
                string line;
                while (std::getline(story, line))
                {
                    cout << line << endl;
                }
                story.close();
```

```
            }
            hard = true;
            menu = false;
            break;
        }
        case 3:
        {
            fstream rules;
            rules.open("C:/Users/woodc/OneDrive/Desktop/Project_2/Rules.txt", std::ios::in);
            if (rules.is_open())
            {
                string line;
                while (std::getline(rules, line))
                {
                    cout << line << endl;
                }
            }
            break;
        }
        //Quit Game
        case 4:
            exit(0);
        }
    } while (menu);
    return hard;
}

void roll(int *dice)
{
    for(int i = 0; i < 2; i++)
    {
        dice[i] = rand() % 6 + 1;            //DIE ROLL
    }
    for(int i = 0; i < 2; i++)
    {
        switch (dice[i])
        {
            case 1:
                die1();
                break;
```

```
            case 2:
               die2();
               break;
            case 3:
               die3();
               break;
            case 4:
               die4();
               break;
            case 5:
               die5();
               break;
            case 6:
               die6();
               break;
         }
      }
};
void e_win(int b, string name)
{
   cout << name << " WON" << endl;
   cout << "MONEY WON = $" << fixed << setprecision(2) << 2 * b << endl;
};
void e_lose(int b, string name)
{
   int lose;
   cout << name << " LOST" << endl;
   cout << "MONEY LOST = $" << fixed << setprecision(2) << b << endl;
};
void die1()
{
   fstream die;
   die.open("C:/Users/woodc/OneDrive/Desktop/Project_2/Die/Die_1.txt", std::ios::in);
   if (die.is_open())
   {
      string line;
      while (std::getline(die, line))
      {
         cout << line << endl;
      }
```

```
        die.close();
    }
};
void die2()
{
    fstream die;
    die.open("C:/Users/woodc/OneDrive/Desktop/Project_2/Die/Die_2.txt", std::ios::in);
    if (die.is_open())
    {
        string line;
        while (std::getline(die, line))
        {
            cout << line << endl;
        }
        die.close();
    }
};
void die3()
{
    fstream die;
    die.open("C:/Users/woodc/OneDrive/Desktop/Project_2/Die/Die_3.txt", std::ios::in);
    if (die.is_open())
    {
        string line;
        while (std::getline(die, line))
        {
            cout << line << endl;
        }
        die.close();
    }
};
void die4()
{
    fstream die;
    die.open("C:/Users/woodc/OneDrive/Desktop/Project_2/Die/Die_4.txt", std::ios::in);
    if (die.is_open())
    {
        string line;
        while (std::getline(die, line))
        {
```

```
            cout << line << endl;
        }
        die.close();
    }
};
void die5()
{
    fstream die;
    die.open("C:/Users/woodc/OneDrive/Desktop/Project_2/Die/Die_5.txt", std::ios::in);
    if (die.is_open())
    {
        string line;
        while (std::getline(die, line))
        {
            cout << line << endl;
        }
        die.close();
    }
};
void die6()
{
    fstream die;
    die.open("C:/Users/woodc/OneDrive/Desktop/Project_2/Die/Die_6.txt", std::ios::in);
    if (die.is_open())
    {
        string line;
        while (std::getline(die, line))
        {
            cout << line << endl;
        }
        die.close();
    }
};
void bubble(float *money, string *users)
{
    float temp;
    string temp2;
    for (int i = 0; i < 1; i++)
    {
        for (int j = i + 1; j < 2; j++)
```

```
        {
            if (money[i] < money[j])
            {
                temp = money[i];
                money[i] = money[j];
                money[j] = temp;
                temp2 = users[i];
                users[i] = users[j];
                users[j] = temp2;
            }
        }
    }
};
void insert(int *win, string *users)
{
    for(int i = 1; i < 2; i++)
    {
        int key = win[i];
        string temp = users[i];
        int j = i - 1;
        while (key > win[j] && j >= 0)
        {
            win[j + 1] = win[j];
            users[j+1] = users[j];
            --j;
        }
        win[j + 1] = key;
        users[j+1] = temp;
    }
};
```