Flowchart

Author: Aurelia Juan Sindhunirmala
Created on June 7, 11:40 PM
Purpose: Flowchart for Project 2 (Blackjack)

System Libraries
I/O Object
Format Library
Random Library
File Stream Library
String Library
Math Library
Vector Library
Standard Namespace

User Libraries
none

Global Constants
const int OPTIONS = 3
const string menuOpt[OPTIONS][2]

Function Prototypes
void first
void Menu
void display
void def
void toc2
void toc2
void toc3
bool Rate
int vldRate
bool exitGm
void printCS
void printFV
void slSt
bool linSc

main

Set random number seed
srand(static_cast <unsigned int>(time(0)));

Declare Variables
none

do-while loop

first → page 2

True

not exitGm() → page 23

False

return 0

```
              ( page 2 )
                  |
                  v
        +-------------------+
        |  Declare Variables|
        |    choice, inp    |
        +-------------------+
                  |
                  v
        +-------------------+
        | static int gmCnt = 0 |
        +-------------------+
                  |
                  v
        / Output                    /
        / "Welcome to Blackjack"    /
        / ======================    /
                  |
                  v
do - while   +----------------+        ( page 3 )
loop         | Menu(choice)   | -->
             +----------------+
                  |
                  v
            < choice < 3 > --False--> < choice == 1 > --True--> ( page 5 )
                  |                          | False
                  | False                    v
                  v                   < choice == 2 > --True--> ( page 7 )
            +-----------+                    | False
            | gmCnt ++  |                    v
            +-----------+             < choice == 3 > --True--> ( page 19 )
                  |                          | False
                  v                          v
        / Output                    /  < deafult > --True--> ( page 5 )
        / "Number of games played" /
```

```
                    ( page 3 )
                        |
                        v
                  +-----------+
                  |   bool    |
                  | vldInp = false |
                  +-----------+
                        |
                        v
                  +-----------+
                  | option = 0 |
                  +-----------+
                        |
          while         |
                        v
                    <not vldInp>  -->  [| display() |]  -->  ( page 4 )
                                                                  |
                                                                  v
                                                          /  Ouput          /
                                                         / "Enter your choice" /
                                                                  |
                                                                  v
  +-----------+    False                                   True              <option == 1
  | vldInp = true | <----------  <option>  ----------------------------------->   or
  +-----------+                                                                 option == 2
                                     |                                             or
                                     | False                                    option == 3>
                                     v                                             |
                           <option < 1 or option > 3>                             v
                                     |                                    /  Output  /
                                     v                                   /  option  /
                              /  Ouput     /
                             / Input invalid /
```

```
string menuOpt [OPTIONS][2] = {
{"1. Rules", "Display the game rules"},
{"2. Start", "Start the game"},
{"3. Exit Game", "Exit the game"}
```

Output
"Menu"

Output
"--"

Output
"Choose the options below:"

i++

i < OPTIONS

Output
setw(12)

i == OPTIONS - 1

Output
setw(16)

Output
menuOpt[i][0] "-" menuOpt[i][1]

page 5

def (int inp = 0)

Output
"Typing " inp " exits the program."

page 6

toc1

Output
"Here are the rules for the game!"
"--"
"1. The game starts with each player placing their bet on the table (max bet: $1000)."
cout << "2. The dealer deals two cards to each player, including themselves." << endl;
" The dealer's first card is dealt face up (shown), while the second card is dealt face down (not shown)."
"3. The value of each card is as follows: "
"   a. All numbered cards are worth their face value"
"   b. Face cards (such as Jacks, Queens,and Kings) are worth 10, and"
"   c. Aces are worth 1 or 11 based on the system itself."
"4. Players can choose to hit (receive another card) or stand (keep their current card) in order to get "
"   as close to 21 as possible without going over. Players can continue to hit until they decide to stand, "
"   or until they go over 21, or known as bust."
"5. Once the player have completed their turn, the dealer reveals their hole card and hits or stands according"
"   to a set of predetermined rules"
"6. If the dealer busts, the player win their bets. If the dealer does not bust, then the player and the"
"   dealer with the hand value closest to 21 without going over wins. If the player's hand value is the same as "
"   the dealer's, it is a tie and the player's bet is returned."
"--"

exitGm()

page 23

```
( page 7 )  →  [| toc2 |]
                  │
                  ▼
        ┌──────────────────────┐
        │  Declare Variables   │
        │ card, faceValm sum,  │
        │     sum2,            │
        │    hsAns, pBet       │
        └──────────────────────┘
                  │
                  ▼
        ╱──────────────────────╲
        │       Output         │
        │    "Let's Start!"    │
        │ "Place your bet. (Bet│
        │  should be less than │
        │       $1000)"        │
        ╲──────────────────────╱
                  │
             True │
                  ▼
             ╱─────────╲              ╱──────────────────╲
             │ pBet >  │─────────────▶│      Output      │
             │  1000   │              │ "Value exceed the│
             ╲─────────╱              │  maximum rule"   │
                  │                   ╲──────────────────╱
            False │
                  ▼
             ╱─────────╲              ┌──────────────────┐
             │ pBet < 0│─────────────▶│ pBet = abs (pBet)│
             ╲─────────╱              └──────────────────┘
                  │
                  ▼
        ╱──────────────────────╲
        │       Output         │
        │ "You put a bet for   │
        │     $ " pBet         │
        ╲──────────────────────╱
                  │
                  ▼
            ┌───────────┐
            │  sum = 0  │
            └───────────┘
                  │
                  ▼
              ( page 8 )
```

**Declare and
Initialize For Loop**
i = 0

i++

i < 1

card = rand() % NUMCRD + 1;
faceVal = (card % 11) + 1;
sum += faceVal;

Output
"Dealer's Card #1 : "

printFV(faceVAl)

printCS(rand() % 4 + 1

pCrd[2]
numPC = 2

i++

i < numPC

card = rand() % NUMCRD + 1;
faceVal = (card % 11) + 1;
sum += faceVal;

Output
"Players Card "

printFV(faceVAl)

printCS(rand() % 4 + 1

i < numPc

True

j = 0

i++

False

j++

j < numPc - i - 1

temp = pCrd[j]
pCrd[j] = pCrd[j + 1]
pCrd[j + 1] = temp

pCrd[j] > pCrd[j + 1]

slSt (pCrd, numPc)

sum2 = 0

i++

i < numPc

sum2 += pCrd[i]

```
page 10
```

Output
"Player's Total : "

sum2 > 21

scVal = 21

bool found → found = linSc (pCrd, numPc, scVal) → found

True → **Output**
"Player has a card with value" scVal

False → **Output**
"Player does not have a card with value " scVal → **Output**
Please proceed the game

Output
"Do you want to hit or stand? "

True

hsAns == H
or
hsAns == h → page 11

False

hsAns == S
or
hsAns == s → page 12?

```
card = rand() % NUMCRD +1;
faceVAl = (card % 11) + 1;
sum2 += faceVal;
```

Output
"Player's new card: "

→ printFV(faceVAl) →

printCS (rand() & 4 +1) →

Output
"Player's Total : " sum 2

True

Output
"Player bust and dealer win!" ← sum2 > 21

False

Output
"You lose you bet of $ " pBet

sum2 == 21 → **Output**
"Player wins!"

Output
"Congrats! You win your bet of $ " pBet

Rate() →

exitGm() →

```
page 12
```

→

```
card = rand() % NUMCRD +1;
faceVAl = (card % 11) + 1;
sum += faceVal;
```

Output
"Dealer's Card #2: "

printFV(faceVAl) →

printCS (rand() & 4 +1) →

Output
"Dealer's Total : " sum 2

sum < 17 →

```
card = rand() % NUMCRD +1;
faceVAl = (card % 11) + 1;
sum += faceVal;
```

Output
(Dealer's new card)

← printFV(faceVAl)

← printCS (rand() & 4 +1)

Output
"Dealer's Total : " sum

→

page 13 → sum > 21

Output
"Dealer busts, player wins!"

Output
"Congrats! You win you bet of $" pBet

Rate() → page 20

exitGm() → page 23

```
      ( page 14 )  ──────►  ◇ sum <= 21 &&
                              sum2 <= 21 ◇
                                   │
                                 True
                                   │
                                   ▼
                            ◇ sum > sum2 ◇ ──────►  [ Output          ] ──────►  [ Output                        ]
                                   │                [ "Dealer wins!"   ]          [ "You lose your bet of $ " pBet ]
                                 False
                                   │
                                   ▼
                            ◇ sum < sum2 ◇ ──────►  [ Output          ] ──────►  [ Output                          ]
                                   │                [ "Player wins!"   ]          [ "Congrats! You win your bet of $" ]
                                 False                                            [ pBet                            ]
                                   │
                                   ▼
                            [ Output        ]
                            [ "It's a tie!" ]
                                   │
                                   ▼
                            [ Output                      ]
                            [ "You get your bet back!"    ]
                                   │
                                   ▼
                            ‖ Rate()  ‖ ──────►  ( page 20 )
                                   │
                                   ▼
                            ‖ exitGm() ‖ ──────►  ( page 23 )
```

```
        ( page 15 )
             |
             |                    [ i++ ]◄─────────────┐
             |                       │                 │
             ▼                       │                 │
          ╱─────╲              ╱──────────╲            │
         ╱ i <   ╲───────────►╱ arr[i] ==  ╲───────────┘
         ╲ size  ╱            ╲    key     ╱
          ╲─────╱              ╲──────────╱
             │                       │
             ▼                       ▼
      ( return false )        ( return true )
```

```
                         True
   ( page 16 ) ───────────────▶ ◇ sign == 1 ◇ ──────▶ ▱ Output
                                      │                  "Diamond"
                                      │ False
                                      ▼
                                 ◇ sign == 2 ◇ ──────▶ ▱ Output
                                      │                  "Club"
                                      │ False
                                      ▼
                                 ◇ sign == 3 ◇ ──────▶ ▱ Output
                                      │                  "Heart"
                                      │ False
                                      ▼
                                 ◇ sign == 4 ◇ ──────▶ ▱ Output
                                      │                  "Spade"
                                      │ False
                                      ▼
                                 ◇ default ◇
```

```
( page 17 )  --True-->  < value == 1
                          or
                          value == 10 >  ---->  [ Output
                                                  "Ace" ]
                              |
                            False
                              |
                              v
                        < value >=2
                          or
                          value < 10 >  ---->  [ Output
                                                 value " " ]
                              |
                            False
                              |
                              v
                        < value == 10 >  ---->  [ Output
                                                  "Jack" ]
                              |
                            False
                              |
                              v
                        < value == 11 >  ---->  [ Output
                                                  "Queen" ]
                              |
                            False
                              |
                              v
                        < value == 12 >  ---->  [ Output
                                                  "King" ]
```

page 18

Declare and Initialize For Loop
i = 0

i++

i < size - 1

indx = 0

j++

j < size

arr[j] < arr[indx]

indx = j

temp = arr[indx'
arr[indx] = arr[i]
arr[i] = temp

```
    page 19

       │
       ▼
┌──┬─────────┬──┐
│  │ exitGm  │  │
└──┴─────────┴──┘
       │
       ▼
    page 23
```

page 20

Output
"THANK YOU FOR VISITING BLACKJACK"

Output
"Please rate us rating from 1 - 10"

inpRate = vldRate(ratings)

Output
"You have rated this program as " ratings[inpRate - 1]

Output
"Thank you for rating this program! Hope you enjoy your visit!"

return true

```
Rate()
```

```
fstream in
```

Declare Variables
rate

```
fstream =
in.open("rate.dat", ios::in)
```

```
vector<string> ratings
```

getline(in, rate) — True → ratings.push_back(rate)

```
in.close()
```

return Rate(ratings)

```
vldRate (const vector<string> &ratings)
```

Declare variable
inpRate

bool
valid = false

not valid

Output
"Enter a number between 1 and " ratings.size()

return inpRate

inpRate >= 1
and
inpRate <= ratings.size()

True

valid = true

False

Output
"Invalid rating. Please try again!"

Declare variable
ext

Output
"Are you sure to exit the game? (y/n)"

ext == Y
or
ext == y

True

Output
"Exiting the game. Goodbye!"

exit(0)

False

first()

return true