#include<iostream>

#include<ctime>

#include<Windows.h>

#include<array>

#include<string>

#include<cstdlib>

#include<iomanip>

#include<algorithm>

#include<random>

using namespace std;

const size\_t suit = 4;

const size\_t ranks = 13;

class MFG

{

public:

MFG();

int roll();

void deal(int &player, int &computer, array<array<int, ranks>, suit> &);

void blackjack();

void roulette();

void craps();

int hit(array<array<int, ranks>, suit> &);

int getBalance();

void balDeposit(int);

private:

int balance;

};

MFG::MFG()

{

balance = 1000;

}//end MFG

int MFG::getBalance()

{

return balance;

}//end get balance

void MFG::balDeposit(int deposit)

{

balance += deposit;

}//end deposit function

void MFG::blackjack()

{

int bet;

int choice;

int hitStand;

while (balance != 0)

{

cout << "(1) Place Wager (2) Main Menu" << endl;

cout << "Choice: ";

cin >> choice;

if (choice == 1)

{

cout << "Current balance is $" << balance << endl;

cout << "How much would you like to bet? $";

cin >> bet;

if (bet == balance)

{

cout << "ALL IN BABY!" << endl;

}//end if all in

if (bet < balance)

{

cout << "Best of luck!" << endl;

}//end if bet

else if (bet > balance)

{

cout << "Invalid funds!" << endl;

cout << "Your current balance is $" << balance << endl;

cout << "Please enter a valid amount. $";

cin >> bet;

}//end else if bet

srand(time(0));

int player = 0;

int computer = 0;

array<array <int, ranks>, suit> deck = { 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10, 11, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10, 11, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10, 11, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10, 11 };

deal(player, computer, deck);

for (auto suit : deck)

{

for (auto ranks : suit)

cout << ranks << " ";

cout << endl;

}//end for loop

cout << "Player has " << player << endl;

cout << "Computer has " << computer << endl;

cout << "Hit (1) Stand (2): ";

cin >> hitStand;//taking input for hit or stand

while (hitStand != 2)

{

if (hitStand == 1)//if player wants to hit

{

player = player + hit(deck);

cout << "Player has " << player << endl;

cout << "Computer has " << computer << endl;

break;

}//end if hit card

}//end while hit card

if (hitStand == 2)//if player wants to stand

{

cout << "You choose to stand!" << endl;

cout << "Player has " << player << endl;

cout << "Computer has " << computer << endl;

}//end if stand

if (computer < 17)//if computer has less than 17, force hit

{

computer = computer + hit(deck);

cout << "Player has " << player << endl;

cout << "Computer has " << computer << endl;

}//end if computer force hit

if ((player > computer) && (player <= 21))

{

cout << "Player Wins!" << endl;

balance += bet;

cout << "Your current balance is $" << balance << endl;

}//end if

if (player > 21)

{

cout << "You have over 21. Sorry but you broke!" << endl;

balance -= bet;

}//if player busts

if (computer > 21)

{

cout << "Computer busted, You win!" << endl;

balance += bet;

}//if computer busts

if (computer == 21)

{

cout << "Computer hit Blackjack! You lose!" << endl;

balance -= bet;

}

}//end if choice 1

if (choice == 2)

{

cout << "Thanks for playing Blackjack!" << endl;

cout << endl;

break;

}//end if choice 2

if (choice < 1 || choice > 2)

{

cout << "WARNING: You did not enter a valid argument!" << endl;

cout << endl;

break;

}//end error choice

}//end while

}//end Blackjack

void MFG::craps()

{

enum class Status { CONTINUE, WON, LOST };

Status gameStatus;

srand(time(0));

int sumOfDice;

int myPoint;

int wager;

int choice;

int count;

while (balance != 0)

{

cout << "Your current balance is $" << balance << endl;

cout << "How much would you like to bet? $";//take in wager

cin >> wager;

cout << "How many times would you like to play? ";//take in number of games

cin >> count;

if (wager == balance)

{

cout << "Going for broke? I like your style!" << endl;

}//end if wager balance

if (wager < 250)

{

cout << "C'mon take a chance! Throw some more money on the table!" << endl;

}//end if wager less than

if (wager < balance)

{

cout << "Let it ride!\n" << endl;

}//end if wager

else if (wager > balance)

{

cout << "Invalid funds!" << endl;

cout << "Your current balance is $" << balance << endl;

cout << "Please enter a valid amount. $";

cin >> wager;

}//end else if error message

while (count)

{

sumOfDice = roll();

switch (sumOfDice)

{

case 7:

case 11:

gameStatus = Status::WON;

break;

case 2:

case 3:

case 12:

gameStatus = Status::LOST;

break;

default:

myPoint = sumOfDice;

gameStatus = Status::CONTINUE;

}//end switch

while (Status::CONTINUE == gameStatus)

{

sumOfDice = roll();

if (sumOfDice == myPoint)

{

gameStatus = Status::WON;

}//end if

if (sumOfDice == 7)

{

gameStatus = Status::LOST;

}//end if

}//end while

if (Status::WON == gameStatus)

{

cout << "\*\*\*\*\*\*Player Won!\*\*\*\*\*\*" << endl;

balance = balance + wager;

if (balance >= 2000)

{

cout << "You're up big! Now is the time to cash in your chips." << endl;

}//end if

}//end if

else

{

cout << "\*\*\*\*\*\*Player Lost! Better luck next time.\*\*\*\*\*\*" << endl;

balance = balance - wager;

}//end else

count -= 1;

}//end while outter

break;

}//end while

}//end Craps

void MFG::roulette()

{

int const MIN\_NUMBER = 1, MAX\_NUMBER = 36;

int number;

int random;

double bet = 0;

double winnings = 0;

int gametype;

int evenodd;

int choice;

while (balance != 0)

{

cout << "Your current balance is $" << balance << endl;

cout << "(1) Place Wager (2) Main Menu" << endl;

cout << "Choice: ";

cin >> choice;

if (choice == 1)

{

cout << "How much would you like to bet? $";

cin >> bet;

if (bet == balance)

{

cout << "You're chancing it all!" << endl;

}//end if

if (bet < balance)

{

cout << "Let's see how lucky you really are!" << endl;

}//end if

else if (bet > balance)

{

cout << "Invalid funds!" << endl;

cout << "Your current balance is $" << balance << endl;

cout << "Please enter a valid amount. $";

cin >> bet;

}//end else if

cout << "Would you like to bet on a specific number (1), odd/even (2), quit (4)? ";

cin >> gametype;

if (gametype == 1)

{

cout << "What number would you like to bet on? ";

cin >> number;

if (number == 00)

number = 37;

srand(time(NULL));

random = rand() % (MAX\_NUMBER - MIN\_NUMBER + 1) + MIN\_NUMBER;

cout << "Spinning..." << endl;

Sleep(2500);

cout << "The number is: " << random << "\n";

if (number != random)

{

cout << "You lose $" << bet << "\n";

balance -= bet;

}//end if

else

{

cout << "You win $" << 35 \* bet << endl;

balance += 35 \* bet;

cout << "Your current balance is $" << balance << endl;

}//end else

}

if (gametype == 2)

{

cout << "Would you like to bet on even (1) or odd (2)? ";

cin >> evenodd;

srand(time(NULL));

random = rand() % (MAX\_NUMBER - MIN\_NUMBER + 1) + MIN\_NUMBER;

cout << "Spinning..." << endl;

Sleep(2500);

cout << "The number is: " << random << "\n";

if (evenodd == 1)

{

if (2 \* (random / 2) == random)

{

cout << "You win $" << bet << endl;

balance += bet;

}//end if

else

{

cout << "You lose $" << bet << endl;

balance -= bet;

}//end else

}//end if

if (evenodd == 2)

{

if (2 \* (random / 2) == random)

{

cout << "You lose $" << bet << endl;

balance -= bet;

}//end if

else

{

cout << "You win $" << bet << endl;

balance += bet;

}//end else

}//end if

}//end gametype == 2

if (gametype == 4)

{

cout << "Thanks for playing!" << endl;

break;

}//end gametype quit

}//end if choice 1

if (choice == 2)

{

cout << "Thanks for playing Roulette!" << endl;

cout << endl;

break;

}//end if choice 2

if (choice < 1 || choice > 2)

{

cout << "WARNING: You did not enter a valid argument!" << endl;

cout << endl;

break;

}//end error choice

}//end while

}//end roulette

int MFG::hit(array<array<int, ranks>, suit> &d)//used for blackjack

{

array<string, 13>display{ "2","3","4","5","6","7","8","9","10","Jack","Queen","King", "Ace" };

int card;

int suit = rand() % 4;

int ranks = rand() % 13;

cout << "Dealing..." << endl;

Sleep(2500);

card = d[suit][ranks];

if (card == 0)

{

card = hit(d);

return card;

}

d[suit][ranks] = 0;

cout << suit << " " << ranks << " " << card << endl;

if (suit == 0)

cout << display[ranks] << " of clubs " << endl;

if (suit == 1)

cout << display[ranks] << " of hearts " << endl;

if (suit == 2)

cout << display[ranks] << " of diamonds " << endl;

if (suit == 3)

cout << display[ranks] << " of spades " << endl;

return card;

}//end hit

void MFG::deal(int &player, int &computer, array<array<int, ranks>, suit> &d)//used for blackjack

{

player = player + hit(d);

computer = computer + hit(d);

player = player + hit(d);

computer = computer + hit(d);

}//end void deal

int MFG::roll()//used for craps game

{

int die1 = 1 + rand() % 6;

int die2 = 1 + rand() % 6;

cout << "Rolling..." << endl;

Sleep(2500);

cout << "Player rolled " << die1 + die2 << endl;

cout << endl;

return die1 + die2;

}//end roll