Jordan Golan

UID: 405328335

Project 6 Report

1. The most difficult obstacle that I encountered when completing this project was having the wheel pick a random character from the inputted sequence. My main issue with this was just using the wrong syntax to try and call the RandomNumber functions provided to us by the professor. I was forgetting to create an object using the RandomNumber constructor, and thus when calling the other random () function I was not getting the correct output. After attending office hours I realized my mistake and further understood the implementation of classes and objects. Another obstacle I encountered was creating the Paytable class. I was able to solve this easily however, It took testing and trial and error to adjust the order in which I coded the different payout scenarios so that they were all correct.
2. List of test data:

// to test paytable

Bank b;

SlotMachine s( "AKQJ987" );

b.deposit(10);

// three aces

b.setWager( 10 );

s.play( b, 'A', 'A', 'A' );

assert( b.balance( ) == 110 );

b.cashout();

// two aces

b.deposit(10);

b.setWager( 10 );

s.play( b, 'A', 'A', '5' );

assert( b.balance( ) == 60 );

b.cashout();

//three of a kind not aces

b.deposit(10);

b.setWager( 10 );

s.play( b, '5', '5', '5' );

assert( b.balance( ) == 80 );

b.cashout();

// any pair other than aces

b.deposit(10);

b.setWager( 10 );

s.play( b, '6', '5', '5' );

assert( b.balance( ) == 40 );

b.cashout();

// a single ace

b.deposit(10);

b.setWager( 10 );

s.play( b, 'A', '7', '5' );

assert( b.balance( ) == 20 );

b.cashout();

// having A, K, Q

b.deposit(10);

b.setWager( 10 );

s.play( b, 'A', 'K', 'Q' );

assert( b.balance( ) == 60 );

b.cashout();

// pair not aces + ace

b.deposit(10);

b.setWager( 10 );

s.play( b, 'A', '5', '5' );

assert( b.balance( ) == 50 );

b.cashout();

// Testing Bank class

Bank c;

//Testing canwager function

c.deposit (10);

// can wager a regular amount less than balance

assert(c.canwager(10) == true);

c.cashout();

// cant wager more than bankamount

c.deposit(10);

assert(c.canwager(100) == false);

c.cashout();

// cant wager negative numbers

c.deposit(10);

assert(c.canwager(-10) == false);

c.cashout();

// testing cashout()

c.deposit(10)

assert(c.balance() == 10);

c.cashout();

assert(c.balance() == 0);

// testing SlotMachine class

// testing play cheating function

Bank d;

SlotMachine t( "AKQJ987" );

d.deposit(10);

d.setWager( 10 );

t.play( d, 'A', 'A', 'A' );

assert( d.balance( ) == 110 );

// testing play not cheating fuction

Use main provided:

d100

b100

p

observe what happends.