

***Normally Busy Computer Consultants***

***"A Member of the JoeMarr Family of Companies"***

**From: Jack Watt / Developer**

**PROG1195 Applications Development Group**

**To: J.E.Marriott / Project Manager**

**Date: 2012/9/28**

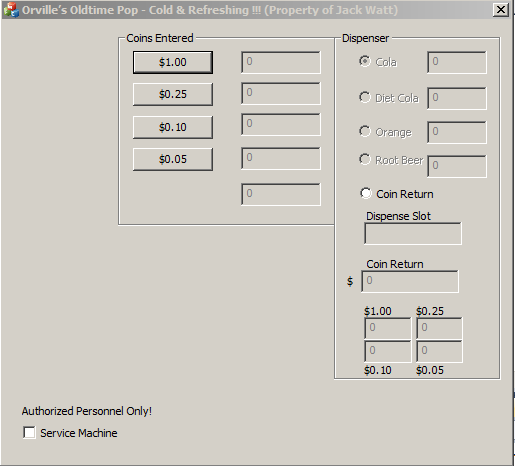
**Re: OOP Phase 5**

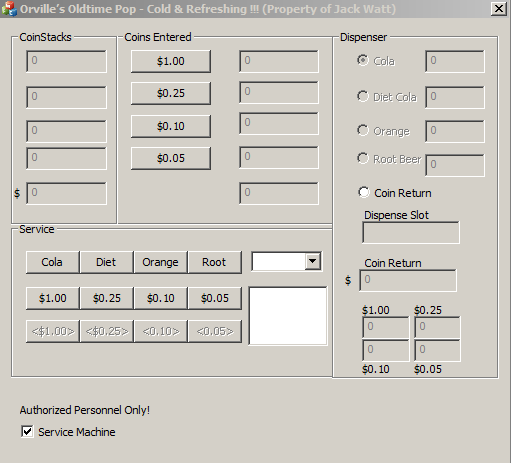
|  |
| --- |
| Lab went well, made some rookie mistakes. Learn from them though and move forward. I really enjoy C++. It’s not really that hard. Just a lot of work. I have noticed that the project folders are big. But I guess that’s windows programming. Pointers soon? I remember being told that pointers are crucial in Win32 C++ GUI programming. I ran into an issue where I had a return statement. And after that I had some code. Wondering why it didn’t work. I was very frustrated . I got through it. |

# Testing Verification Document

[  **tested & verified ][** **partially functional, explanation provided] [ *MCj04325370000[1]*test failed or not completed ]**

1. [] Place a screen shot of your application here…





1. [] Above screen shot shows Correct Application Design

( Size / Titlebar / Control Attributes & Alignment / Spelling )

1. [] Service controls Show / Hide with Password Protection.

“Cancel” (does not produce message) and

“Incorrect Password” (appropriate message) handled correctly.

1. [] “PopDispenser Radios” & “Service Unload Coins” controls Enable / Disable

depending on corresponding contents of CoinStack and Dispenser Edit controls

1. [] Enter $2. If CoinReturn is selected, money is returned through CoinReturn controls,

and CoinsEntered controls clear. Dispense Slot displays appropriate message.

1. [] If a Pop is requested, that kind is available, and the Money matches the PopPrice of

$1.05,a Pop “appears” in the Dispense Slot, the appropriate pop Dispenser decreases by one, and the values found in CoinsEntered are added to the CoinStacks. CoinsEntered controls are then set to zero.

1. [] After the dispense action of step6, the CoinReturn slots should not still be showing

the results of step 5.

1. [] If a Pop is requested and not enough money has been entered, a message is displayed

but nothing happens with CoinsEntered.

1. [] “More than $1.05 entered” is correctly handled.

(eithercorrect change is made, or all money is returned with appropriate message.)

1. [] Message on Exit with correct functionality (OK, CANCEL).

***Continued on next page…***

1. [MCj04325370000[1]] “Other” issues you have identified? (state “None”, or elaborate)
2. [] No floating point variables or literals used in ANY financial calculations/comparisons (other than DISPLAY PURPOSES ONLY) (avoiding penny problem)
3. []

Paste your Dispense code into this box…

int CWattOOPDlg::Dispense(void)

{

enum ACTION { COLA, DIET, ORANGE, ROOT, COINRETURN } ButtonPressed;

ButtonPressed = (ACTION) m\_optDispenserSelected;

if ( calculateCoinsEntered() > 105 && ButtonPressed != 4) {

MessageBox("Exact change only, please." , "Sorry!", MB\_ICONSTOP);

ButtonPressed= COINRETURN;

}

if ( ButtonPressed == COINRETURN ) {

//move all CoinEntered fields to CoinReturn fields

m\_txtDispenserCoinReturn = 0;

m\_txtDispenserCoinReturn += m\_txtCoinsEntered10 \* 10;

m\_txtDispenserCoinReturn += m\_txtCoinsEntered100 \* 100;

m\_txtDispenserCoinReturn += m\_txtCoinsEntered25 \* 25;

m\_txtDispenserCoinReturn += m\_txtCoinsEntered5 \* 5;

m\_txtDispenser10 = m\_txtCoinsEntered5;

m\_txtDispenser100 = m\_txtCoinsEntered100;

m\_txtDispenser25 = m\_txtCoinsEntered25;

m\_txtDispenser5 = m\_txtCoinsEntered5;

//Calculate the $$$ value of the CoinReturn fields

m\_txtDispenserCoinReturn /= 100;

//set all CoinEntered fields to zeros

m\_txtCoinsEntered100 = 0;

m\_txtCoinsEntered25 = 0;

m\_txtCoinsEntered10 = 0;

m\_txtCoinsEntered5 = 0;

UpdateData(FALSE);

return 0;

}

if (calculateCoinsEntered() < 105) {

MessageBox( "Insufficient funds entered." , "Sorry!" , MB\_ICONSTOP );

UpdateData(FALSE);

}

if ( ButtonPressed == COLA && m\_txtDispenserCola == 0) {

MessageBox( "None available. Try another selection." , "Sorry!", MB\_ICONSTOP);

UpdateData(FALSE);

}

if ( ButtonPressed == DIET && m\_txtDispenserDiet == 0) {

MessageBox( "None available. Try another selection." , "Sorry!", MB\_ICONSTOP);

UpdateData(FALSE);

}

if ( ButtonPressed == ORANGE && m\_txtDispenserOrange == 0) {

MessageBox( "None available. Try another selection." , "Sorry!", MB\_ICONSTOP);

UpdateData(FALSE);

}

if ( ButtonPressed == ROOT && m\_txtDispenserRootBeer == 0) {

MessageBox( "None available. Try another selection." , "Sorry!", MB\_ICONSTOP);

UpdateData(FALSE);

}

if ( ButtonPressed == COLA && m\_txtDispenserCola > 0 && calculateCoinsEntered() == 105) {

m\_txtCoinStack10 += m\_txtCoinsEntered10;

m\_txtCoinStack100 += m\_txtCoinsEntered100;

m\_txtCoinStack25 += m\_txtCoinsEntered25;

m\_txtCoinStack5 += m\_txtCoinsEntered5;

m\_txtCoinsEntered10 = 0;

m\_txtCoinsEntered100 = 0;

m\_txtCoinsEntered25 = 0;

m\_txtCoinsEntered5 = 0;

m\_txtDispenserCola--;

if (m\_txtDispenserCoinReturn != 0){

m\_txtDispenserSlot = "Klunk";

}

UpdateData(FALSE);

}

if ( ButtonPressed == DIET && m\_txtDispenserDiet > 0 && calculateCoinsEntered() == 105) {

m\_txtCoinStack10 += m\_txtCoinsEntered10;

m\_txtCoinStack100 += m\_txtCoinsEntered100;

m\_txtCoinStack25 += m\_txtCoinsEntered25;

m\_txtCoinStack5 += m\_txtCoinsEntered5;

m\_txtCoinsEntered10 = 0;

m\_txtCoinsEntered100 = 0;

m\_txtCoinsEntered25 = 0;

m\_txtCoinsEntered5 = 0;

m\_txtDispenserDiet--;

if (m\_txtDispenserCoinReturn != 0){

m\_txtDispenserSlot = "Klunk";

}

UpdateData(FALSE);

}

if ( ButtonPressed == ORANGE && m\_txtDispenserOrange > 0 && calculateCoinsEntered() == 105) {

m\_txtCoinStack10 += m\_txtCoinsEntered10;

m\_txtCoinStack100 += m\_txtCoinsEntered100;

m\_txtCoinStack25 += m\_txtCoinsEntered25;

m\_txtCoinStack5 += m\_txtCoinsEntered5;

m\_txtCoinsEntered10 = 0;

m\_txtCoinsEntered100 = 0;

m\_txtCoinsEntered25 = 0;

m\_txtCoinsEntered5 = 0;

m\_txtDispenserOrange--;

if (m\_txtDispenserCoinReturn != 0){

m\_txtDispenserSlot = "Klunk";

}

UpdateData(FALSE);

}

if ( ButtonPressed == ROOT && m\_txtDispenserRootBeer > 0 && calculateCoinsEntered() == 105) {

m\_txtCoinStack10 += m\_txtCoinsEntered10;

m\_txtCoinStack100 += m\_txtCoinsEntered100;

m\_txtCoinStack25 += m\_txtCoinsEntered25;

m\_txtCoinStack5 += m\_txtCoinsEntered5;

m\_txtCoinsEntered10 = 0;

m\_txtCoinsEntered100 = 0;

m\_txtCoinsEntered25 = 0;

m\_txtCoinsEntered5 = 0;

m\_txtDispenserRootBeer--;

if (m\_txtDispenserCoinReturn != 0){

m\_txtDispenserSlot = "Klunk";

}

UpdateData(FALSE);

}

}

[] All personal debug code (if any) removed or deactivated.

[] Submission package complete as per requirements. (See Blackboard)

**I attest I have tested and verified each of the above specifications, *unless indicated otherwise*.**

**I understand that falsely claiming a verified spec indicates my attempt to falsify results to the Quality Assurance Department, and I will be subject to additional penalties.**

**Name: Jack Watt**

**Date: 2012/9/28**