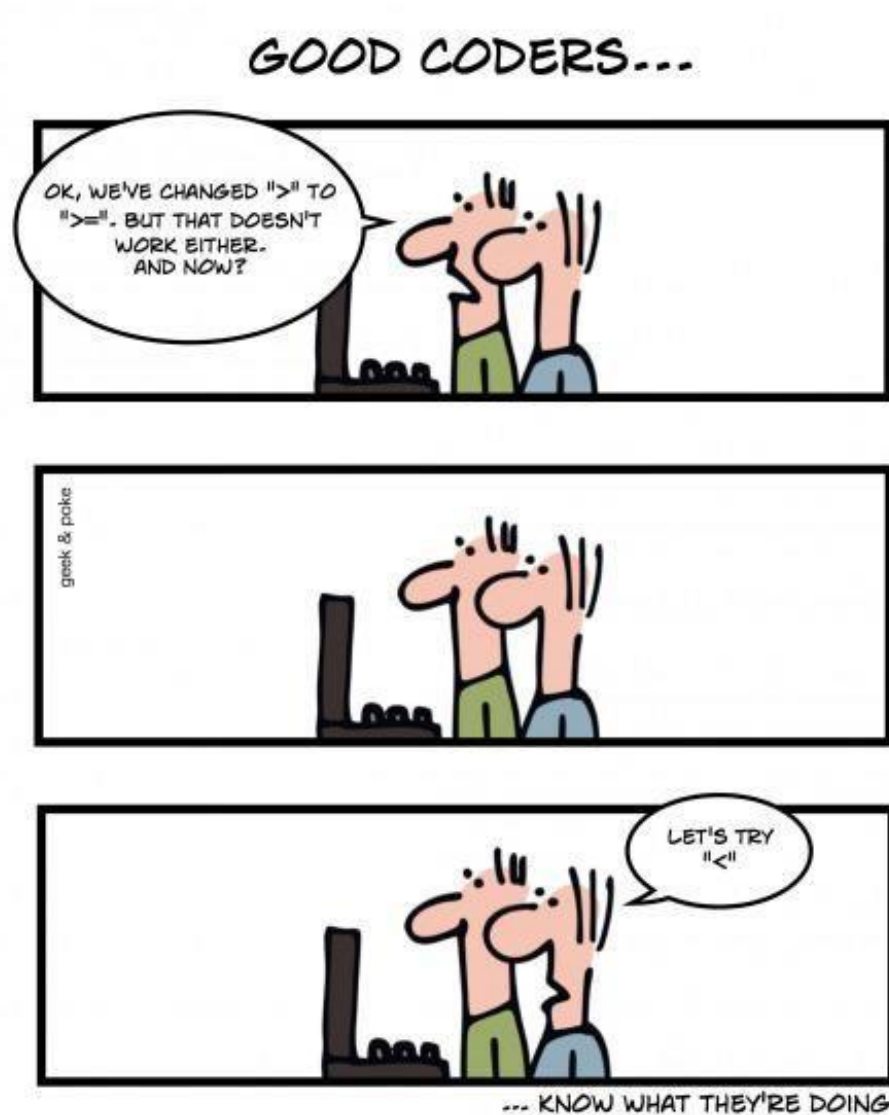


Enriching the CVM-4 with a graphical interface

productdocument



Student: Youri Claes

629836

Vak: Inleiding Software Engineering, D-B-INSE-O

Docent: Jos Onokiewicz

18-1-2020

Documenthistorie

Datum	Versie	Wie	Veranderingen
9-1-2020	1v	Youri Claes	De initiale niet gedocumenteerde productdocument bijwerken
18-1-2020	1.1v	Youri Claes	Geheel ingevuld. Vraag, waar moet ik nog meer de nadruk opleggen?

SAMENVATTING

Eisen

Initiale eisen:

- Het maken van een GUI wrapper om CLI programma
- Functie die print naar het GUI programma
- Uitbreiding van oorspronkelijk programma

De eisen waren aanvankelijk het maken van een GUI wrapper om een CLI programma. Echter kwam wij er al gauw achter dat dit niet te doen was met de API die wij gebruikte. Van daar dat we de eisen heb veranderd naar het uitbreiden en compleet nieuw ontwikkelen van een GUI programma met de functionaliteit van de CVM als basis deze is oorspronkelijk ontwikkeld is door Jos Onokiewicz.

Eisen:

- Registreer en verwerk van het ingeworpen geldstuk.
- Registreer van pinpas.
- Registreer van pincode als pinpas is geregistreerd.
- Check of genoeg is betaal bij selecteren van een product uit machine.
- Check of pinpas en pincode kloppen.
- Wisselgeld teruggeven wanneer dat benodigds.

VOORWOORD

Goedendag, beste lezer.

Dit vak liep eigenlijk prefect. Door dat ik al aardig wat programmeer kennis had, hetzij voor mijn plezier en niet in C maar in C++ en shell. Heeft dit mij enorme geholpen. Door het eerste blok heen heb ik echt de meeste finesses geleerd. Hier door was het ook niet moet om het tentamen met een goed resultaat af te sluiten. Aangezien ik eigenlijk wat meer uitdaging wou ging ik het mezelf moeilijker maken en de opdrachten uitbreiden. Ook ging ik eigenproject je opzetten. Hierdoor leerde ik eigenlijk meer dingen dat noodzakelijk was voor dit vak. Misschien dat ik volgende keer zou kijken of ik niet meteen al de lat wat hoger kan leggen.

De opdracht die we kregen wat het maken van een state machine voor een apparaat naar keuze. Maar ook hier ging ik het me zelf moeilijker maken. Ik ging in overleg met Jos win32api gebruiken om een GUI te maken voor de CVM-4. Het is idee was natuurlijk en leuk, maar de uitvoering was wat minder. Winapi is enorm lastig en kan als een taal an sich worden gezien. Naar veel research, documentatie lezen en vragen stellen aan mensen die er verstand van hadden is het me gelukt. Door het gebruik van winapi heb ik niet alleen veel geleerd over windows, ik heb ook enorme veel geleerd over C en hoe je je code leesbaar kun houden. Als ik het over zou doen had ik een ander api gekozen, windows api is best wel achterhaalt vandaar dat ik dan een moderne api zou gebruiken.

Al met al was dit een erg leuk vak, en ik vind het dan ook jammer dat het vak niet klaar is. Ik heb enorm veel geleerd en ga zeker als hobby door met programmeren

Youri Claes

Inhoudsopgave

Samenvatting.....	3
Eisen.....	3
Voorwoord	4
1. Inleiding.....	6
2. Definitiefase.....	7
2.1 Kennismaking en doel van Cola Vending Machine	7
2.2 Functionele eisen	8
2.3 Technische eisen	8
2.4 Schets user interface	8
3. Ontwerpen	9
3.1 Architectuur	9
3.2 State chart	9
4. Realisatie en testen	10
4.1 Realisatie	10
4.2 Acceptatietesten	11
5. Eindresultaat en aanbevelingen	12
5.1 Eindresultaat	12
5.2 Aanbevelingen	12

1. INLEIDING

Dit hoofdstuk gaat in op de opbouw van dit document.

Hoofdstuk 2, gaat dieper in op de technisch aspecten en realisatie van de CVM-5

Hoofdstuk 3, gaat dieper in op hoe de deelsystemen met elkaar samenwerken

Hoofdstuk 4, In dit hoofdstuk kijken hoe de realisatie van het product ging en aan welke testen die onderworpen is

Hoofdstuk 5, spreek over het hoe het resultaat uiteindelijk geworden is en wat we de volgende keer anders zouden aanpakken

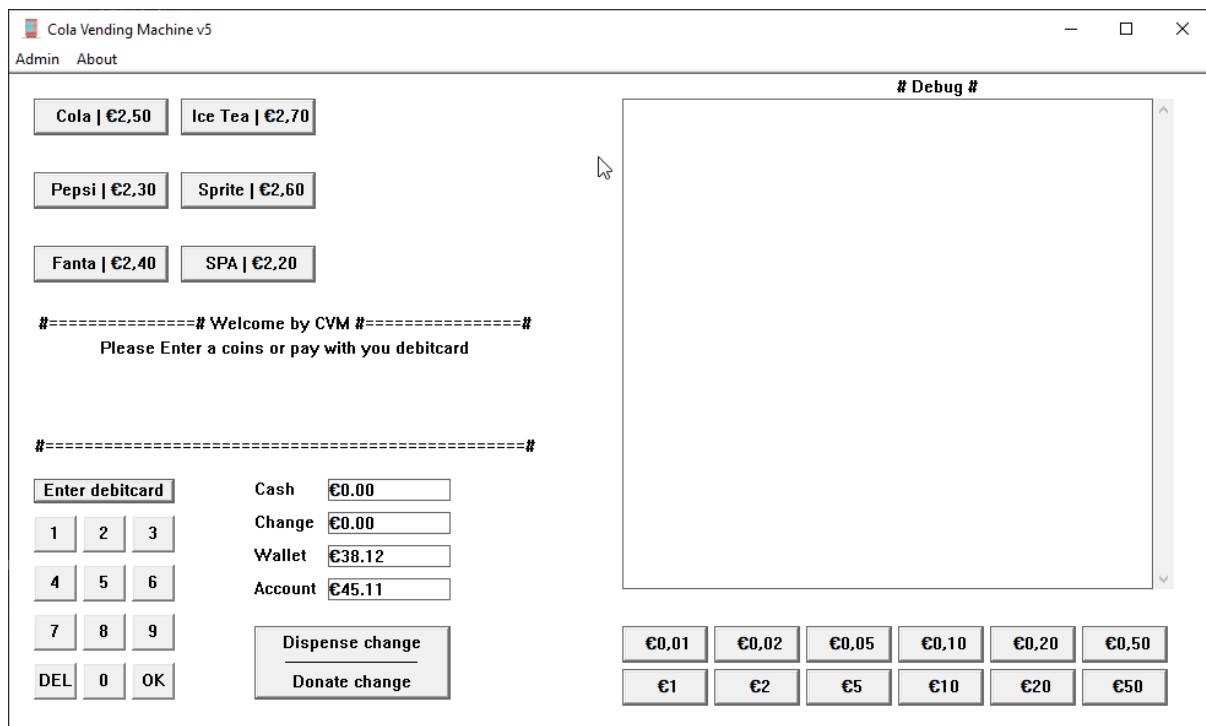
2. DEFINITIEFASE

Dit hoofdstuk gaat in op de gestelde eisen aan de te realiseren besturingssoftware van Cola Vending Machine Gui

2.1 Kennismaking en doel van Cola Vending Machine

<Korte niet-technische beschrijving apparaat, doel? Welk probleem lost het op? Door wie gebruikt? Wat levert het op?>

Het apparaat zorgt voor de mogelijkheid om drankjes te kunnen kopen zonder dat het bemant hoeft te zijn. Hierdoor kunnen je 24/7 drinken kopen en hoeft de verkoper niet aanwezig te zijn. Het is mogelijk dat het zou worden gebruikt door winkels en/of winkelcentra en op soortgelijke pleken



Rechts zitten de echte componenten, Pin menu, Portemonnee, Bank, knopen voor kopen en wisselgeld knop.

Links zitten de gesimuleerde componenten munten in werpen en debug scherm

2.2 Functionele eisen

De functionele eisen zijn als volgt:

1. Koop van frisdrank met contant geld
 - 1.1. Het geven van de frisdrank
 - 1.2. Het geven wisselgeld
2. Koop van drankje met bankpas
 - 2.1. Het geven van de frisdrank
 - 2.2. Het verwerken met bankrekening
3. Debug functie
 - 3.1. Het kunnen inzien van wat er op de achtergrond gebeurt
 - 3.2. Extra geld voor testen
4. Het schrijven naar een display
 - 4.1. Gebruiker op inlichten via menu

2.3 Technische eisen

De technische eisen zijn als volgt:

1. De programmeertaal C moet worden gebruikt.
2. De ontwikkelomgeving Visual Studio 2019 Professional
3. Gebruikte API: winapi

2.4 Schets user interface

Debug scherm, Weergave scherm voor geld, inwerp sleuven voor contant geld, pin knoppen, sleuf pinpas.

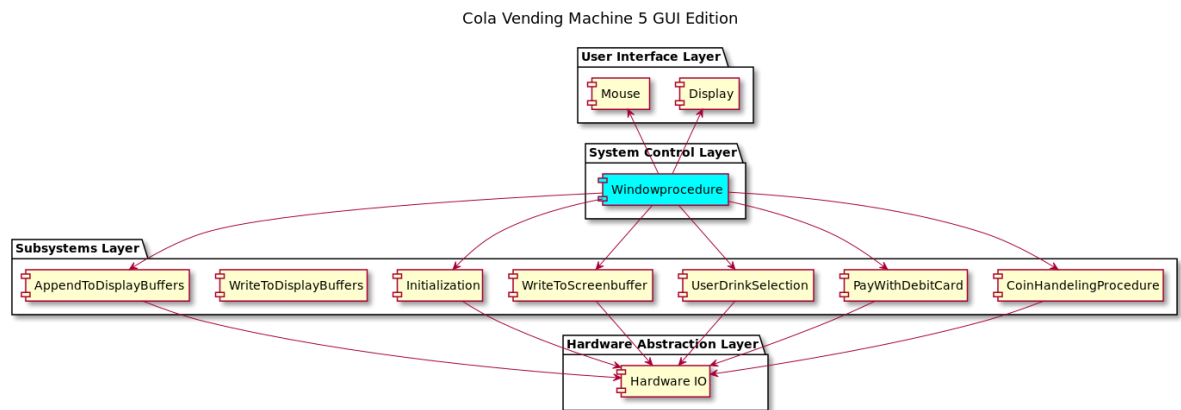
3. ONTWERPEN

Dit hoofdstuk gaat in op de Cola Vending Machine en het ontwerp van de systemen

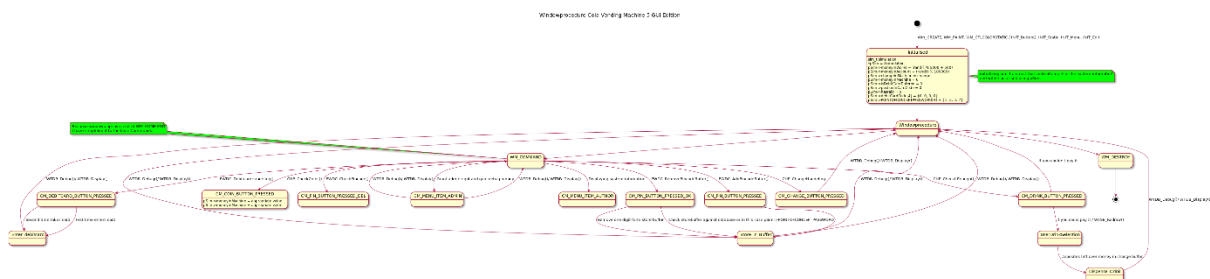
3.1 Architectuur

Subsystemen in CVM-5

- AppendToDisplayBuffers
- WriteToDisplayBuffers
- Initialization
- WriteToScreenbuffer
- UserDrinkSelection
- PayWithDebitCard
- CoinHandlingProcedure



3.2 State chart



Note voor Jos “De Statechart is te groot, is goed als die in een apart mapje komt en er dan naar verwijst?”

4. REALISATE EN TESTEN

Dit hoofdstuk bevat informatie over hoe het programma gerealiseerd is en aan welke testen die onderworpen is.

4.1 Realisatie

<Korte weergave: Qt gebruik, instellingen, versies? Inhoud .pro file, Modulaire opbouw code, welke files? Directory structuur? Wat niet gemaakt? Bijv HAL, Bekende bugs? plantUML-gebruik, welke tools? is er codedocumentatie? Doxygen, Gebruikershandleiding beschikbaar?>

Voor de realisatie van de CVM-5 hebben we gebruik gemaakt van Microsoft Visual Studio Professional 2019.

Hier bij is er rekening gehouden met niet te lange regels echter zijn er wel uitzonderingen gemaakt, de api die wij gebruikt had een aantal functie die ons vereisten lange regels te maken.

We hebben volledig kunnen uit werken, naar wat het plan was. Toch hebben we een aantal dingen niet uitgewerkt. We hebben geen weergave voor de pincode tekens gemaakt, ook hebben we geen database achter de pin functionaliteit, we hebben alleen een pincode die het concept laat zien dit kan worden uitgebreid.

Het UML diagrammen zij gemaakt in plantUML, hiervoor is gebruik gemaakt van een plugin van VS19.

De code is gedocumenteerd door doxygen.

Opbouw file structuur:

- ColaVendingMachine-5
 - Docs
 - html
 - Doxyfile
 - Icon.png
 - Productdocument
 - IMG
 - Productdocument
 - PUML
 - PlantUML files
 - Source
 - ColaVendingmachine
 - Sourcefiles
 - Debug
 - x64
 - ColaVendingMachineGui.sln

4.2 Acceptatietesten

De volgende testen zijn Chronologisch uitgevoerd door de developer.

Test	Action/input	Expected result	Pass Fail	Actual result if test has failed
1.	Monkey Test: Allerlei knoppen drukken en kijken wat gebeurt	Dat er fouten uit naar voren komen	FAIL	Programma deed niet wat ik had moeten doen
2.	Gebruiker laten kijk	Dat is gebruiker er zonder fouten mee kan werken	PASS	
3.	Pin functionaliteit Test	Dat pincode geverifieerd word en dat je de pincode kunt wissen	PASS	
4.	Monkey Test	Dat er fouten uit naar voren komen	PASS	Geen fouten, had bugs toen gemaakt
5.	Change Test	Dat je je wisselgeld terug krijgt en dat je ook kun doneren	PASS	
6.	Drink Test	Dat je drinken kunt kopen en dat je niet te veel betaalt	PASS	
7.	Display Test	Dat het display naar behoren werkt	PASS	

De testen keken ieders naar een specifiek onderdeel van het programma.

De monkey testen waren om bugs die je normaal niet ziet te filteren vandaar dat die twee keer gedaan is

5. EINDRESULTAAT EN AANBEVELINGEN

Dit hoofdstuk bevat het slot van het document het zet dan ook dat laatste dingen recht betreffende de CVM-5

5.1 Eindresultaat

We zijn erg trots met het eind resultaat. De CVM-5 heeft nog ons weten geen fouten en is een erg mooi stukje techniek.

5.2 Aanbevelingen

De volgende zouden we een andere api gebruiken. Ook zouden we er dan voor zorgen dat je bij deze automaat ook warme dranken kan krijgen en ook wat te eten