

Project bachelor 3: escaperoom

De vuilbak

Tim Meskens en Jarno De Jonckheere

5/05/2022
2021-2022

Inhoudsopgave

1	Inleiding	2
2	Analyse	2
3	Ontwerp en realisatie	2
3.1	Github	2
3.2	Kostanalyse	3
4	Evaluatie	3
4.1	Aftoetsen van vereisten	3
4.2	SWOT	3
4.3	Uitbreidingen	4
4.4	Moeilijkheden	4
4.5	Verdeling	5
5	Conclusie	5

1 Inleiding

We moeten niemand meer uitleggen dat het sorteren van afval de dag van vandaag een must is. Toch loopt dit in sommige gevallen nog eens fout. Het doel van deze puzzel is het vinden en juist sorteren van afval. Wat dient er gerealiseerd te worden?

- een puzzel maken die in staat is te communiceren met andere puzzels.
- een systeem ontwerpen dat in staat is het soort afval te herkennen.
- Het afval moet gewogen kunnen worden en dit moet een integraal deel van de puzzel zijn.
- Een gebruiksvriendelijke en intuïtieve puzzel ontwerpen en implementeren.

2 Analyse

Wanneer de vuilnisbakken het signaal krijgen dat de buffer op groen staat zal het scherm aan gaan en kan er een code ingegeven worden. De vuilnisbakken zullen geopend worden via de code die via vorige opdrachten wordt gevonden. De eerste 4 cijfers van de code zullen gestuurd worden door 'K ga meegaan' en het laatste cijfer door 'De trein is altijd een beetje reizen'. Wanneer de buffer weer op oranje of rood komt te staan zal het scherm weer uit gaan.

De speler zullen aan de hand van scanplatformpjes op de vuilbak hun vuilnis kunnen scannen. Als het juiste vuilnis op de juiste scanner wordt geplaatst en er wordt gedrukt op de bijhorende knop dan zal er een succesgeluidje te horen zijn en komt er op de LCD dat het afval juist gesorteerd werd, en dat het afval in de juiste vuilbak gegooid mag worden. De puzzel zal wachten totdat dit gebeurd is. Wanneer dit fout gesorteerd is zal er een fout geluidje komen en zal er een fout naar de buffer gecommuniceerd worden. Als er niets wordt gescand gebeurd er niets. Uiteindelijk zal de som van gewicht in de verschillende vuilnisbakken een code vormen. Deze wordt dan doorgestuurd naar het eindslot.

3 Ontwerp en realisatie

3.1 Github

Via deze link komt u bij onze documentatie:

<https://plan-it-b.github.io/ba3-docs/docs/DeVuilbak/DeVuilbak.html>

3.2 Kostanalyse

Component	Aantal	Prijs/stuk	Totaal
ESP32	1	4,36	4,36
20x4 LCD +I2C	1	10	10
Keypad 3x4 Matrix	1	4	4
PN532	3	6	18
10Kg Load sensor	3	5	15
HX711	3	2	6
Multiplexer TCA9548A	1	3,5	3,5
Audio amplifier	1	4,5	4,5
Speaker	1	2	2
PCB	1	5	5
AMS1117 3.3v	1	0,5	0,5
Knoppen	3	0,5	1,5
Totaal			74,36

In het totaal (zonder karton, weerstanden en condensators) zou deze puzzel €74,36 kosten. In de realiteit zijn we niet aan dit bedrag gekomen aangezien de RFID scanners en de LCD al op school aanwezig waren.

4 Evaluatie

4.1 Aftoetsen van vereisten

- Communicatie: De communicatie met andere puzzels werkt. Zowel het ontvangen als het verzenden van berichten gebeurt zonder fouten.
- Herkenning afval: De RFID scanners scannen het afval normaal gezien goed. Het enige probleem dat er kan ontstaan is dat de sticker niet gevonden wordt. Daarom hebben we besloten om geen fouten door te sturen als er niets gescand wordt, hierdoor worden er geen onnodig fouten doorgegeven.
- Wegen van het afval: Jammer genoeg werkten 2 weegschalen niet, we vermoeden dat dit kwam door slechte solderingen van ons (of een kapotte sensor). De weegschaal voor het pmd werkt wel.
- Gebruiksvriendelijk: Naar onze vaststellingen is de vuilbak gebruiksvriendelijk. Nadat de code is ingevoerd staat er op het lcd-scherm duidelijk te lezen wat de bedoeling is. De scanplatformen en vuilbakken zijn duidelijk aangegeven en door het geluid kom je te weten of je een fout hebt gemaakt of niet. Voor nog meer verduidelijking hebben we de vuilbak beplakt met aanwijzingen.

4.2 SWOT

Sterktes:

- Het scannen verloopt goed
- De puzzel kan goed op afstand bedient worden

- De communicatie verloopt goed

Zwaktes:

- Het herkennen wanneer afval in de juiste vuilnisbak wordt gegooit is niet waterdicht.
- De PCB is te groot

Kansen:

- Uitbreidbaar met andere soorten afval
- Maar 1 RFID scanner gebruiken of automatisch scannen zonder op een knop te drukken.
- Mogelijkheid om afval te wegen

Risico's:

- Door vele niet gesoldeerde draadjes is het niet heel robuust en komen er soms draadjes los.

4.3 Uitbreidingen

Als uitbreiding zouden we meer soorten vuilnis kunnen gebruiken en voor elke nieuwe ronde verschillende soorten vuil aanduiden die moeten gesorteerd worden.

4.4 Moeilijkheden

Tijdens het ontwerpen van de puzzel zijn we op enkele moeilijkheden gestoten. Zo was het grootste probleem dat de ESP32 te weinig GPIO's had om al onze componenten aan te sturen. Hierdoor hebben de keuze moeten maken om één GPIO die met het toetsenbord verbonden was ook te gebruiken als interrupt pin voor al de RFID scanners. Deze werden nooit tegelijk gebruikt. Bij het ontwerpen op het breadboard ondervonden we hier geen problemen bij. Ook hebben we de reset pinnen van de RFID's achterwege gelaten omdat dit op het breadboard zonder werkte. (Achteraf gezien hadden we twee GPIO-pins over die we hiervoor gebruikt konden hebben.)

Achteraf gezien zouden we dit probleem beter hebben opgelost door in plaats van aparte drukknoppen te gebruiken voor het aanspreken van de sensoren, dit via knoppen op het toetsenbord te doen. Ook zouden we maar van 1 RFID scanner en dus maar 1 scanplatform gebruik kunnen hebben gemaakt. In dit geval zou je gewoon moeten ingeven welk afval dat je denkt dat het is door bijvoorbeeld 1,2 of 3 in het toetsenbord in te geven. Door het toetsenbord te gebruiken kwamen er ook weer 3 pins van de knoppen vrij.

Daarnaast hadden we het ook lastig met de weegschalen. We hebben tegen het einde gemerkt dat er maar 1 weegschaal werkte omdat we altijd met eenzelfde testen deden. Omdat we een tijdstekort hadden hebben we besloten dat we enkel het pmd zouden wegen. Wanneer we de 2 andere weegschalen werkende hadden gekregen was dit zeer makkelijk uitbreidbaar.

4.5 Verdeling

We hebben eerst samen de werking van alle componenten onderzocht. Hierna hebben we het werkt verdeeld. Jarno ontwierp de behuizing en de PCB, terwijl Tim de ESP heeft geprogrammeerd. Het debuggen deden we hierna samen.

5 Conclusie

Onze belangrijkste verwezenlijkingen vinden we dat we in staat zijn geweest alle componenten met elkaar te laten samenwerken ondanks de beperkte IO-pins. Moesten we het opnieuw doen hadden we het anders aangepakt qua gebruik van IO's en randapparaten maar we zijn al bij al toch tevreden met het resultaat.