

VERSLAG TAAK MICRO:BIT

Besturingssystemen & Systeemfundamenten

Rayane Kouidane

N°0587073

1^e Bachelor Computerwetenschappen

Academiejaar 2021-2022

1. Multiplexed display

Bij de eerste oefening van de taak moeten we patronen laten verschijnen op de 5x5 led display. Deze sectie bespreekt de belangrijkste procedures die hiervoor nodig zijn.

Om een led aan te zetten moet je een bepaalde rij aanzetten en een bepaalde kolom uit. Om dit te kunnen doen heb ik, de code die uitlegt hoe multiplexing werkt, bekeken. Deze code heb ik dan omgezet in Scheme-code. Eerst zet ik alle nodige pins op output modus. Hiermee zeg ik dat elk van deze pin spanning kan doorgeven en niet dat het de waarde moet aflezen. In deze oefening is de draw-procedure de belangrijkste procedure. Deze zal ervoor zorgen dat we het patroon op onze display zien. De draw-procedure werkt als volgt: Het gaat door alle kolommen en voor elke kolom gaat het door alle rijen. Afhankelijk van de waarde in de matrix (als het bezig is met rij *i* en kolom *j*, dan kijkt het de waarde op rij *i* en kolom *j* in de matrix) en gaat het een rij aanzetten. Door snel de kolom aan en uit te zetten lijkt het alsof het lichtje brandt. Als we het delay argument vervangen door een groter getal, dan kunnen we het effect waarnemen.

2. Bediening m.b.v. drukknoppen

Deze sectie bevat cruciale code dat nodig was voor de implementatie van de tweede oefening. Hiermee moeten we een led bedienen door gebruik te maken van de drukknoppen.

In deze oefening maak ik een led aan als een object met een positie (rij en kolom). Door object-gericht te gaan werken kan ik een licht abstraheren tot een object dat berichten verstaat. Wanneer ik op een knop druk stuur ik een bericht naar dit object dat een procedure zal uitvoeren. Deze procedure zal ervoor zorgen dat de positie van de led veranderd (in de matrix). Voordat het de positie verandert, zet het een 0 in de matrix op de huidige positie. Daarna geef ik een nieuwe positie aan de led en zet ik een 1 in de matrix op de nieuwe positie. Omdat de draw-procedure mijn matrix bij elke spel-lus tekent, zien we dat de led van plaats is veranderd.

3. Sneeuwvlokken

Voor deze oefening moeten we een klein spel implementeren waarbij de speler sneeuwvlokken moet vangen. Als de speler 3 sneeuwvlokken op de grond laat vallen dan is het spel gedaan en geeft het de score van de speler terug.

Voor dit spel heb ik het balkje en de sneeuwvlok als een led-object geïmplementeerd. Een led-object onthoudt belangrijke informatie zoals: zijn positie (het kan ook zijn positie wijzigen), een variabele die aangeeft of een led gevangen werd (en procedures om deze variabele te wijzigen), een procedure die controleert of een led-object een ander object heeft aangeraakt en een procedure die controleert of een led-object de grond heeft aangeraakt.

Daarna is er een spel-lus (gerepresenteerd door een for-loop) waarin de logica van het spel geïmplementeerd is. Hierbij laten we het balkje bewegen, de sneeuwvlok vallen, kijken we of een sneeuwvlok genomen werd of niet. Ik laat mijn sneeuwvlok versnellen door te kijken naar de 'range' van de for-loop. Als de 'range' deelbaar is door een variabele, dat een getal voorstelt en dan laat ik het sneeuwvlok vallen. Na 5 seconden deel ik de variabele door 2 en gaat de sneeuwvlok versnellen. (Bijvoorbeeld: Er zijn meer getallen die deelbaar zijn door 200 dan door 400). Ik werk op deze manier omdat we voor de cursus 'Programmerproject 1', een spel object-gericht moesten implementeren. Ik ben dus vertrouwd met het object-gericht programmeren.

4. Sterren vangen m.b.v. accelerometer

Deze sectie bespreekt de implementatie van het spel 'sterren vangen' m.b.v. de accelerometer.

Om dit spel te implementeren heb ik gebruik gemaakt van de code die in de slides staan om de waarden af te lezen die de accelerometer teruggeeft. De waarden die de z-as teruggaf waren niet nodig, omdat het net links, rechts, op en neer moet bewegen. Eerst maak ik een object aan waarin deze waarden terug te vinden zijn. De procedure om zo'n object aan te maken is `make-axis-controller`. Daarna begin ik met een `for-loop`, dat mijn spel-lus voorstelt, en hierin lees ik bij elke lus de waarden van de assen af, controleer ik steeds of het net een ster heeft gevangen, geef ik een nieuwe willekeurige positie aan de ster, laat ik mijn net bewegen door te kijken of de gegeven waarden van de x en y-as boven of onder een zekere grens liggen en geef ik de score terug.

5. Creatieve opdracht met Piezo buzzer

Voor de laatste oefening heb ik gekozen om het spel sneeuwvlokken uit te breiden met de piezo-buzzer. De uitbreiding is de toevoeging van zelfgemaakte geluidseffecten. Ik heb voor de volgende gebeurtenissen een geluidseffect ingevoerd: intro van het spel, bij het nemen van een sneeuwvlok, bij elke versnelling en bij het verliezen van het spel. Dit zorgt ervoor dat het spel aangenamer wordt. Om dit te kunnen doen, heb ik de procedure (`play duration dly`) gebruikt die we hebben gezien tijdens de les over de Raspberry Pi. Eerst heb ik enkele geluidseffecten gedefinieerd en deze in een lambda geplaatst. Als ik dit niet deed, dan speelde het geluid af in het begin. Daarna zet ik deze geluidseffecten op de juiste plaats in mijn spel-lus. Een andere uitbreiding is dat ik ook grotere patronen kan tonen op mijn display door het te laten scrollen over mijn display. Om dit te doen kopieer ik de vijf rijen tussen de huidige kolom en de vijf volgende kolommen in mijn grotere matrix. Daarna laat ik de draw-procedure deze matrix tekenen en dit doe ik voor elke kolom. Als de score groter dan 10 is dan werkt het op dezelfde manier. Eerst heb ik een vector gemaakt voor elk getal van 0 tot 9. Als ik een getal heb dat groter is dan 10 dan plak ik de vectoren samen (dankzij de procedure `append-two-numbers`). Eerst zet ik het getal om in een string en ik kijk naar de individuele cijfers van het getal. (Bijvoorbeeld: voor het getal 24 neem ik de vector '2' en plak ik die aan de vector '4'.)

Gebruikte bronnen

Uitleg multiplexing

<https://www.appelsini.net/2011/how-does-led-matrix-work/>

<https://www.youtube.com/watch?v=X9tsfOeYnAU>

Hulp-procedure

(zie oefening5.rkt lijn 144)

<https://stackoverflow.com/questions/67648496/how-to-cut-a-part-of-a-matrix-in-racket>