

L01P01 - Flowcharts

Maken van programma's met een plan.

Geschreven door

Arjan Kamberg

<https://www.linkedin.com/in/arjankamberg/>

Leerjaar: 2021/ 2022

Copyright
Arjan Kamberg

Van stroomdiagram naar Python code

© 2020, Arjan Kamberg

Uitgegeven in eigen beheer

(Python@ElsArjan.nl)

Alle rechten voorbehouden.

Niets uit deze uitgave mag worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand en/of openbaar gemaakt in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of op enige andere manier zonder voorafgaande schriftelijke toestemming van de uitgever. Het gebruik maken van dit boek voor opleidingsdoeleinden mag alleen gedaan worden met uitdrukkelijke toestemming van de uitgever.

Versie Datum: 23-01-2023

Inhoudsopgave

Voorwoord	3
H1 - Stroomdiagram	4
1.1 Draw.io	6
1.2 Begin en Einde	8
1.3 Processen	9
1.4 Invoer (Input) en uitvoer (output)	10
1.5 Keuze	11
1.6 Opdrachten	12
1.6.1 Dranken automaat	12
1.6.2 Lichtschakelaar	12
1.6.3 Deur	12
1.6.4 Getal raden	12
1.6.5 Hoger / Lager	12
2 Een spel in een stroomdiagram	13
2.1 Beschrijving Galgje	13
2.2 Spelregels Galgje	13
2.3 Stroomdiagram maken van galgje	14
2.3.1 Stap 1 – woord kiezen	14
2.3.2 Stap 2 – Eerste invoer	15
2.3.3 Stap 3 – Controle op winst en verlies	16
2.3.4 De computer laten reden	17
2.4 Opdrachten	18
2.4.1 Hoger Lager	18
2.4.2 Boter-Kaas-en-Eieren	18
2.4.3 Boter-Kaas-en-Eieren slimmer maken	19
3 Complexe Stroomdiagrammen	20
3.1 Opdrachten	21
3.1.1 Keuze van programma's	21
3.1.2 Yahtzee	21
3.1.3 Stroomdiagram Boter-Kaas-en-Eieren slimmer maken	21
3.1.4 Paper-Scissors-Rock	21

Voorwoord

Alle apparaten om ons heen hebben tegenwoordig wel een computerprogramma in zich dat de functies bepaald. Dit boek is geschreven voor MBO-studenten die programmeren leren aan de hand van de taal Python. Python is een taal die op steeds meer gebieden ingezet wordt door zijn kracht en eenvoud.

In dit boek worden niet alle dingen behandeld, de focus zit op het begrip van programmeren leren. Zo worden lussen bewust alleen met while-statements gemaakt. De opbouw van het programmeren moet eenduidig zijn en er moet zo min mogelijk keuzestress zijn bij het programmeren. De overwegingen en didactische uitleg wordt beschreven in de docenten versie van dit boek.

Bij een goed begrip van programmeren, en een opbouw die duidelijk en testbaar is wordt bereikt dat het eindproduct ook uitbreidbaar is en dat het zich aan de conventies houdt die bruikbaar zijn bij programmeurs met jaren ervaring.

**Programmers when
they don't understand
someone else's code!**

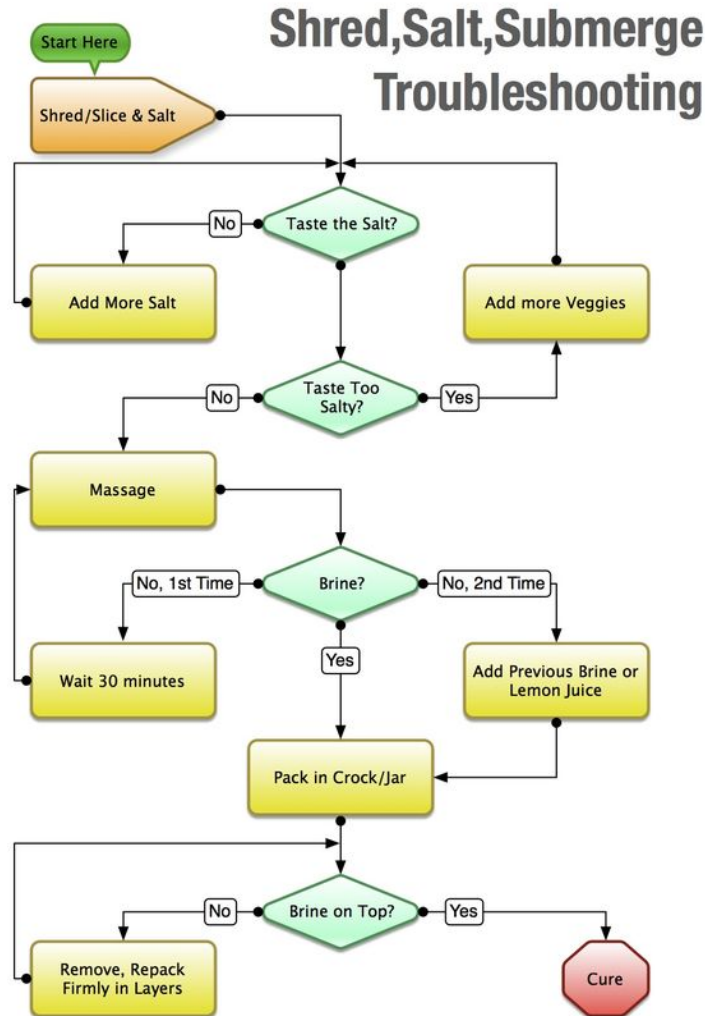


Figuur 1: Programmeurs die iemand anders code niet snappen. (C) CSEstack.org

H1 - Stroomdiagram

Een stroomdiagram geeft schematisch weer wat er moet gebeuren, en niet zozeer hoe iets moet gebeuren. Een stroomdiagram is er niet alleen voor programma's te beschrijven maar kan je voor van alles gebruiken waar logische stappen elkaar opvolgen. Het geeft de acties en keuzes weer van begin tot einde.

Zo kan je een stroomdiagram maken van een computerprogramma, maar ook hoe je een ei moet bakken, of voor het bedienen van een lichtschaakelaar.



Figuur 2: Stroomdiagram van internet. (Niet geheel volgens onze normen)

Stroomdiagrammen worden ook heel veel gebruikt om foutopsporingen te doen. Er wordt dan steeds een vraag gesteld, waarbij je uiteindelijk tot het probleem en oplossing komt.

Er worden meerdere symbolen gebruikt om een stroomdiagram te tekenen. In dit boek houden we het eenvoudig met 4 verschillende symbolen. Met deze symbolen is alles te beschrijven. De symbolen die we niet gebruiken zijn handig om speciale gevallen duidelijk te onderscheiden, maar deze zijn uiteindelijk weer met de 4 symbolen ook te maken.

- Een cirkel geeft het begin en einde van het proces aan.
- Een rechthoek geeft een proces aan. Een proces is een berekening. Een proces kan ook een hele verzameling van stappen zijn die uitgevoerd moeten worden.
- Een parallelogram die aangeeft dat er invoer van buitenaf nodig is.
Dit kan een gebruiker zijn die wat in moet voeren, maar het kan ook een temperatuur sensor zijn. Een parallelogram kan ook de uitvoer zijn naar buitenaf. Een tekst die op het scherm getoond wordt.
- Een ruit geeft een keuze aan. Een keuze is altijd een Ja of Nee vraag.

Al deze symbolen zijn verbonden met een pijltje. De punt van de pijl geeft aan in welke richting het proces loopt. Pijlen mogen elkaar alleen kruisen als dat echt niet anders op te lossen is.

Een stroomdiagram wordt bij voorkeur gemaakt en gelezen van boven naar beneden, en van links naar rechts. De eerste keuze om een volgend symbool neer te zetten is dan dus onder het vorige symbool. Helemaal bovenaan staat dus een cirkel met daarin "Begin" geschreven. In verband met de ruimte kan er ook voor gekozen worden om vanaf linksboven naar rechts te werken.

1.1 Draw.io

Er zijn veel programma's die we kunnen gebruiken voor het tekenen van een stroomdiagram. In dit boek wordt gebruik gemaakt van het programma DRAW.IO.

Dit programma is gratis, en kan zowel in een browser als offline op je computer worden geïnstalleerd.

De onlineversie is gelijk te vinden door de naam draw.io in je browser te typen.

Als je het programma wilt downloaden, of naar de site van de makers wilt gaan dan moet je naar :

<https://www.diagrams.net/>



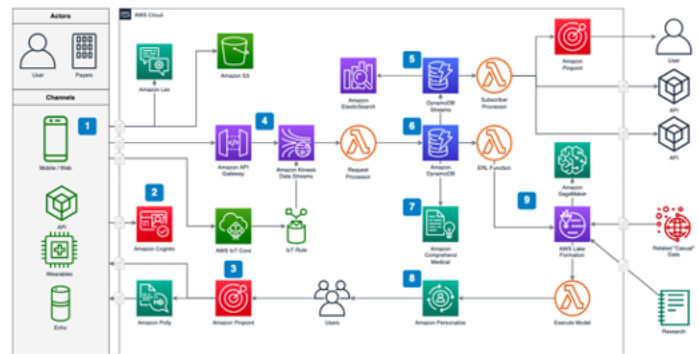
Diagram with anyone,
anywhere.

diagrams.net is open source, online, desktop and container deployable diagramming software

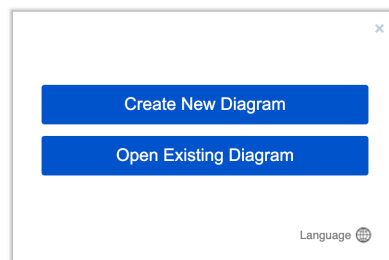


No login or registration required.

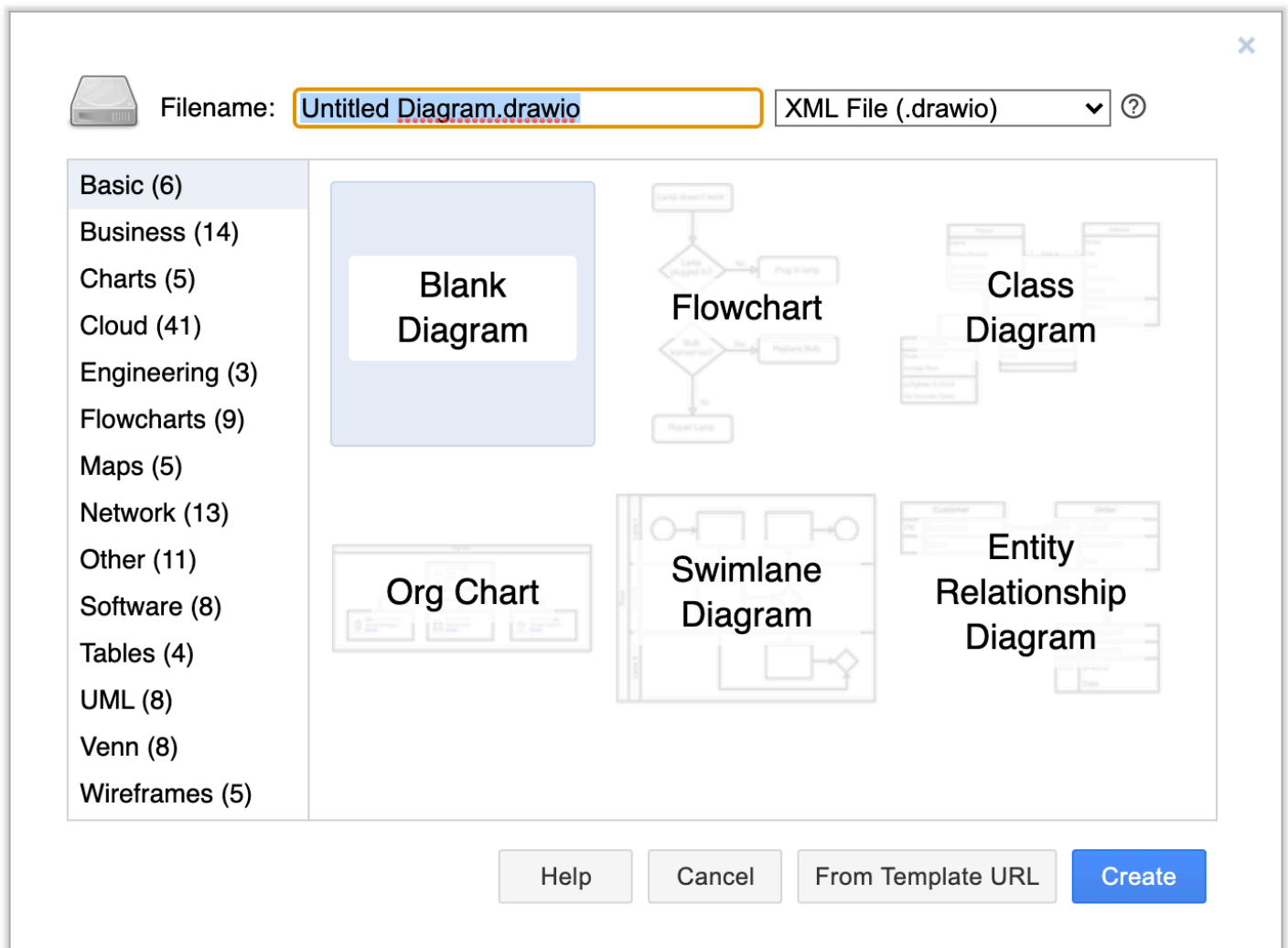
[Blog](#)

[Start Now](#)

Na het starten van het programma wordt gevraagd of je een nieuw diagram wilt maken, of een bestaande wilt openen.



Bij het maken van een nieuw diagram komt er een scherm naar voren die vraagt of we met een voorbeeld willen beginnen, of met een Blank Diagram. Het is heel verleidelijk om een standaard stroomdiagram te selecteren en om te bouwen. Wij gebruiken een beperkt aantal symbolen die afwijkt wat de makers van dit programma gebruiken. Voor het stroomdiagram die wij dus gebruiken beginnen we met de Blank Diagram.

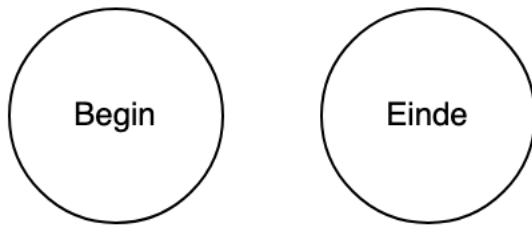


Aan de linker kant van het scherm zijn de verschillende symbolen te zien die in het programma beschikbaar zijn. Voor de stroomdiagrammen in dit boek gebruiken we alleen de symbolen die hieronder met rood onderstreept zijn. Deze kan je vinden in het tabblad van General en van stroomdiagram.



Op <https://www.youtube.com/watch?v=64MaQYyAN2w> staat een korte en handige uitleg hoe je moet werken met DRAW.IO. Let op dat in dit filmpje symbolen worden gebruikt die wij niet gebruiken. Dit filmpje is alleen om te kijken hoe de interface werkt.

1.2 Begin en Einde



Figuur 3: Symbool voor begin en einde

Dit zijn de symbolen die je altijd in je stroomdiagram zal tegenkomen. Ieder stroomdiagram heeft maar 1 begin, en minimaal 1 einde. Je mag meerdere keren het rondje met einde in je stroomdiagram gebruiken, maar dat is dan alleen om de tekening duidelijk te houden. Er wordt dan met ieder eindrondje hetzelfde eindpunt bedoeld.

Start is het beginpunt van je stroomdiagram. Vanuit de start gaat er maar een pijl naar een volgend punt. Een pijl geeft aan wat er na dat element moet gebeuren.

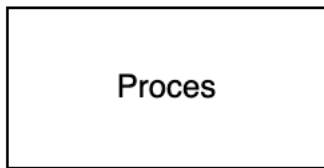
Einde is het stoppunt van je stroomdiagram. We beschrijven de normale situaties en geen bijzondere gevallen. We zetten niet in een stroomdiagram dat op ieder punt de stroom kan uitvallen, en daardoor het stroomdiagram tot een einde komt.

In “Figuur 4: Meest simpele stroomdiagram” is een perfect stroomdiagram van een proces dat helemaal niets doet weergegeven.



Figuur 4: Meest simpele stroomdiagram.

1.3 Processen



Figuur 5: Symbool proces

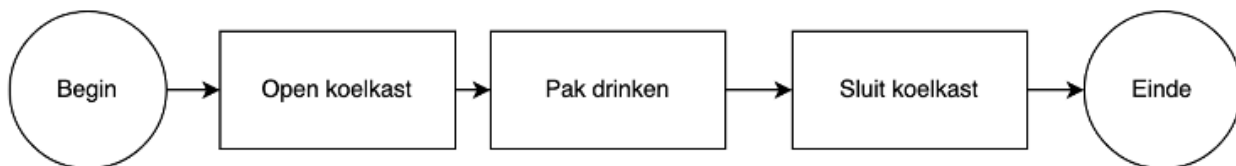
Een proces in het stroomdiagram is waar er berekeningen gedaan worden. Wat er gedaan moet worden wordt kort en bondig in het rechthoekje gezet. Omdat er hier iets gedaan wordt, staat in deze beschrijving altijd een **werkwoord**.

In de beschrijving staat nooit het woord 'en'. Op het moment dat het woord 'en' gebruikt zou worden, moet er een nieuw proces achter het vorige gezet worden.



Figuur 6: Een proces in een stroomdiagram

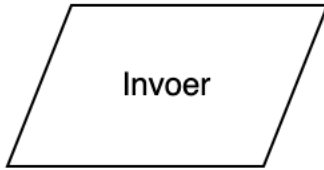
Probeer de verschillende processen in taal duidelijk te onderscheiden. Een stroomdiagram om drinken uit de koelkast te halen kan je heel kort over doen: Pak drinken. Maar er gaan heel veel stappen aan vooraf voordat je drinken hebt. In de processen zie je de werkwoorden: open, pak en sluit.



Figuur 7: Proces drinken uit koelkast

Dit stroomdiagram kan nog veel uitgebreider, we kunnen ook nog zeggen dat eerst naar de koelkast gelopen moet worden. Het is de kunst om het stroomdiagram minimaal functioneel te houden.

1.4 Invoer (Input) en uitvoer (output)

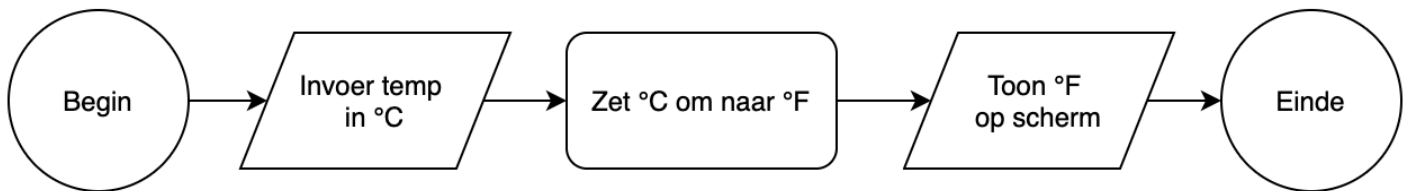


Figuur 8: Symbool invoer

In een stroomdiagram geeft het symbool voor invoer alle invoer van buitenaf weer. Dit kan zijn dat een gebruiker zijn naam moet invoeren, maar het kan ook zijn dat een temperatuur wordt uitgelezen. Ook hier is de taal die we gebruiken in het symbool erg belangrijk. Invoer is **geen** vraag “staat het licht aan”, invoer is “Meet licht intensiteit” of “Bepaal stand schakelaar”.

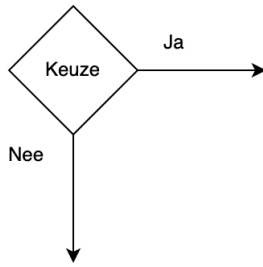
Dit symbool wordt ook gebruikt om alle uitvoer naar bijvoorbeeld het scherm te beschrijven.

De invoer wordt in dit blok niet verwerkt, het is alleen de invoer. De verwerking van de invoer wordt dan weer in het opvolgende proces blok gedaan.



Figuur 9: Stroomdiagram van temp invoer naar berekening en uitvoer

1.5 Keuze



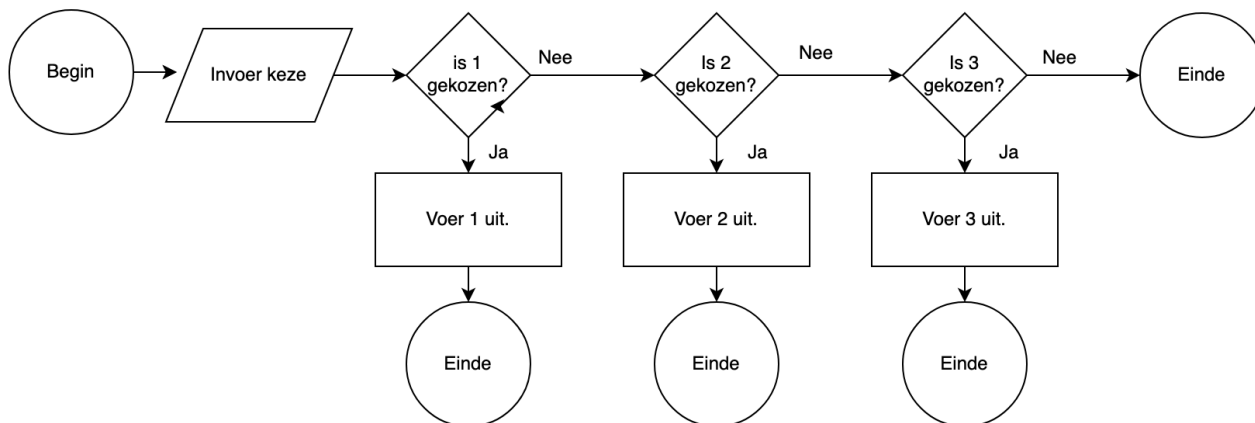
Figuur 10: Symbol keuze

Overall worden keuzes gemaakt, en een keuze wordt weergegeven met een ruit. In de keuze staat niet dat de huidige temperatuur gemeten wordt, en bij > 0 iets gebeurt. Voor het meten wordt het aparte invoer-symbool. Een keuze zorgt ervoor dat er een andere kant in het stroomdiagram opgegaan wordt.

“Is de temperatuur groter dan 0 graden?”.

Een keuze heeft maar twee antwoorden, “Ja” en “Nee”. De vraag moet dan ook zo gesteld worden dat er een “Ja” of een “Nee” uit komt. Een keuze is een vraag, en eindigt dan ook met een vraagteken.

Als er een keuze is met meerdere antwoorden, dan moet je deze opsplitsen in meerdere vragen zodat je uiteindelijk weer alleen vragen hebt die “Ja” en “Nee” als resultaat heeft.



Figuur 11: Opstapeling van keuzen.

1.6 Opdrachten

1.6.1 Dranken automaat

Teken het stroomdiagram die hiernaast staat weergegeven na in DRAW.IO.

Benoem de stappen die je tegenkomt.

Vertel bij ieder symbool waarom dat symbool is gekozen.

1.6.2 Lichtschakelaar

Teken in DRAW.IO het stroomdiagram van een lichtschakelaar.

Bij de start is het licht aan of uit. Vervolgens kan het naar de andere stand gezet worden en is het stroomdiagram klaar.

1.6.3 Deur

Teken in DRAW.IO het stroomdiagram van een Deur. Als je aankomt kan de deur open of dicht zijn. Als de deur open is, loop je door de deur en doet deze dicht. Dan stop het stroomdiagram. Was de deur dicht, dan doe je hem open, loopt door de deur en doe hem dan dicht. Dan stop het stroomdiagram.

1.6.4 Getal raden

Teken in DRAW.IO het stroomdiagram van een spelletje getal raden.

Na start neemt de computer een getal in gedachte. Het volgende is dat de speler een getal moet ingeven.

Er is dan een keuze:

- Als het getal goed is, dan feliciteren we de speler en het proces is klaar.
- Als de keuze niet goed is gaan we verder.
Het volgende wat kan is dat het getal te hoog of laag is.
De computer zeg bij te hoog dat het getal kleiner is.
Als het getal te klein is zegt de computer dat het getal groter moet zijn.
We gaan dan weer terug naar het invoeren van het getal.

1.6.5 Hoger / Lager

Teken in DRAW.IO het stroomdiagram van een spelletje Hoger/Lager.

Na het starten worden de kaarten geschut.

De bovenste kaart wordt omgedraaid.

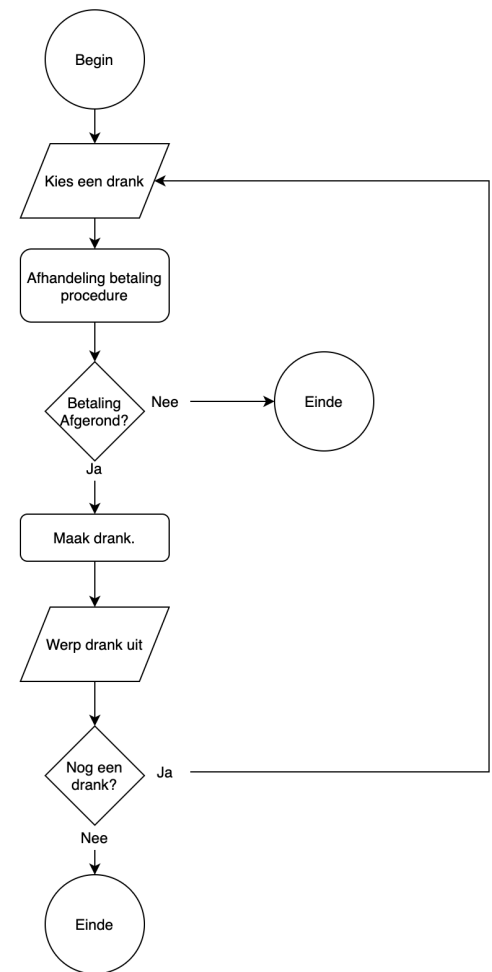
(A) De speler gok of de volgende kaart hoger of lager is.

De kaart wordt omgedraaid.

Als de kaart gelijk is, dan heeft de speler verloren (Aas 7 en Klaver 7 zijn gelijk en beide 7 punten).

Als de speler niet goed heeft geraden is het spel over.

Als de speler goed heeft geraden gaan we weer naar (A).



2 Een spel in een stroomdiagram

Met de symbolen die we tot nu toe hebben geleerd kunnen we van alles een stroomdiagram maken. We gaan nu niet gelijk programmeren. Voordat we iets kunnen programmeren moeten we eerst een goed stroomdiagram maken van wat we willen programmeren.

2.1 Beschrijving Galgje

In dit hoofdstuk gaan we stap voor stap een stroomdiagram maken van het spel galgje.

Op de wiki-pagina van galgje kunnen we een beschrijving van het spel lezen.

Galgje is een spelletje waarbij een woord moet worden geraden door middel van het raden van individuele letters, waarbij de spelers die het woord moeten raden maar een beperkt aantal fouten mogen maken. De fouten worden weergegeven door een galg met daaraan een hangend mannetje (een stokfiguur), welke per fout steeds verder wordt opgebouwd. Uit hoeveel onderdelen de galg bestaat varieert per persoon en regio, maar dit ligt meestal rond de 10.

* bron : <https://nl.wikipedia.org/wiki/Galgje> op 22-02-2020

2.2 Spelregels Galgje

Om geen ruzie te krijgen over de spelregels worden de regels gebruikt die verder op die pagina te lezen valt.

De uitdager (speler 1) neemt een woord in gedachten en zet een rijtje met even veel stippen op papier als er letters in het woord zijn. De tegenstander (speler 2) noemt een letter. Als die in het woord voorkomt zet speler 1 op alle stippen waar in het woord de letter voorkomt de geraden letter. Komt de door speler 2 genoemde letter echter niet voor in het woord, dan tekent speler 1 een onderdeel van de galg op het papier. Soms wordt, om het makkelijk te houden, de foute letter ook elders op het papier geschreven zodat speler 2 kan zien welke letters al geraden zijn, maar het kan ook zijn dat hij dit zelf moet zien te onthouden.

.....
Dit wordt herhaald tot ofwel speler 2 alle letters heeft genoemd, het woord heeft geraden en zo het spel wint, of als speler 2 te veel foute raadpogingen doet waardoor de galg wordt voltooid en speler 2 "hangt".

.....
Woorden die met galgje gebruikt worden mogen niet meer dan 16 verschillende letters hebben. Als er 17 verschillende letters in het woord zouden zitten, zouden er maar 9 'foute' letters zijn, en zou de tegenstander altijd winnen.

* bron : <https://nl.wikipedia.org/wiki/Galgje> op 22-02-2020

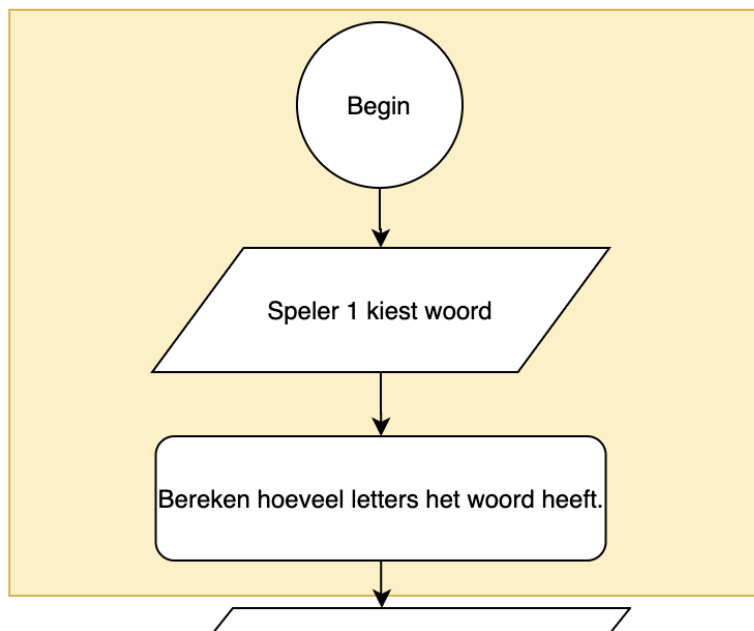
2.3 Stroomdiagram maken van galgje

Het makkelijkste is om stap voor stap de spelregels na te lopen. Speler 1 zou bij ons de computer kunnen zijn, maar we kunnen het spel ook zo maken dat het niet uitmaakt of er tegen een speler of tegen een computer gespeeld wordt. We kunnen uiteindelijk ook het stroomdiagram volgen als we het spel op papier spelen.

2.3.1 Stap 1 – woord kiezen

Uit de spelregels halen we volgende regel. Het is hier gelijk de vraag of het woord een proces is of een invoer. Als een speler het woord invoert is het duidelijk een “invoer” maar als de computer een woord kiest uit een lijst van woorden is er geen interactie vanaf buiten nodig. In dat geval zou het een proces kunnen zijn. Omdat we hier van de spelregels omzetten gebruiken we het invoer-symbool.

De uitdager (speler 1) neemt een woord in gedachten en zet een rijtje met het aantal streepjes op papier als er letters zijn in het woord.



Figuur 12: Stap 1 stroomdiagram Galgje

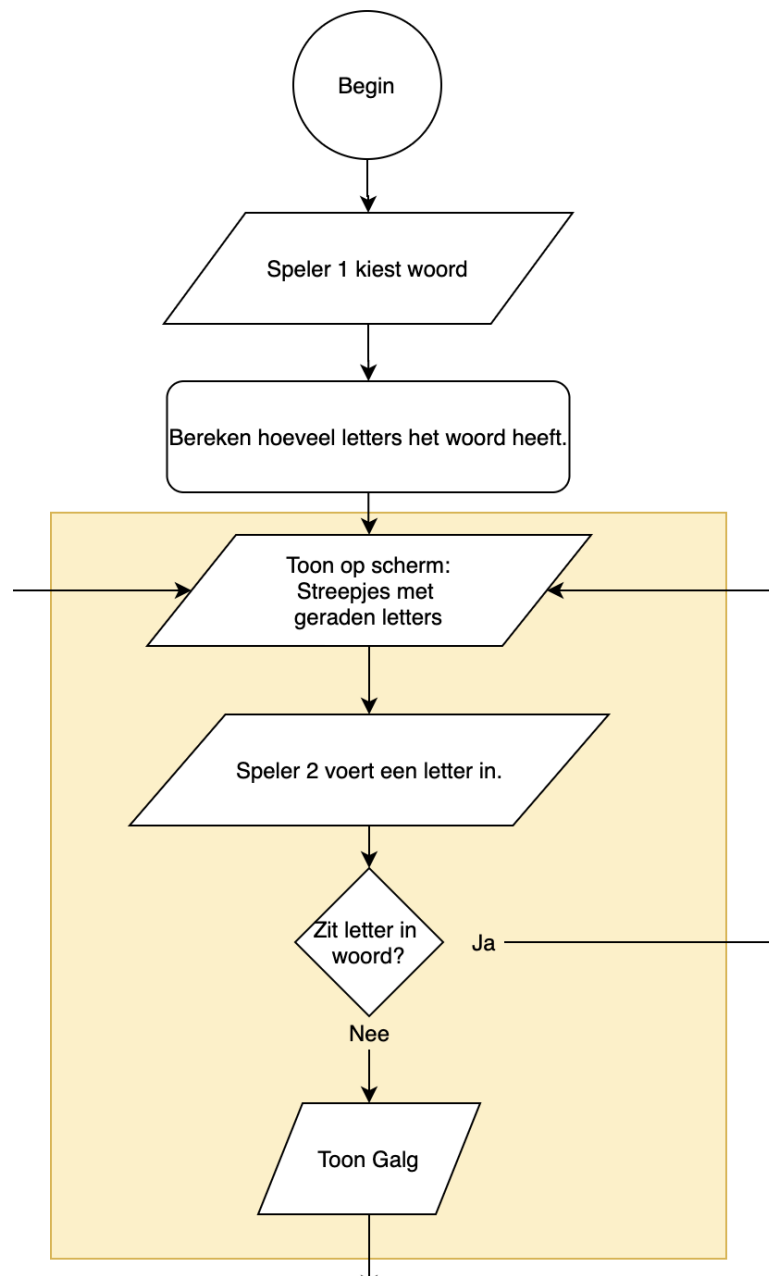
2.3.2 Stap 2 – Eerste invoer

De tegenstander (speler 2) noemt een letter.

De volgende regel is een invoer van de andere speler, dus het volgende is een invoer-symbool.

Het volgende is een keuze. “Komt letter voor in woord?” Op die keuze kan het antwoord “Ja” of “Nee” zijn. Er wordt beschreven wat er moet gebeuren nadat het antwoord “Ja” is, en wat als het antwoord “Nee” is.

Als die in het woord voorkomt zet speler 1 op alle stippen waar in het woord de letter voorkomt de geraden letter. Komt de door speler 2 genoemde letter echter niet voor in het woord, dan tekent speler 1 een onderdeel van de galg op het papier. Soms wordt, om het makkelijk te houden, de foute letter ook elders op het papier geschreven zodat speler 2 kan zien welke letters al geraden zijn, maar het kan ook zijn dat de speler dit zelf moet zien te onthouden.

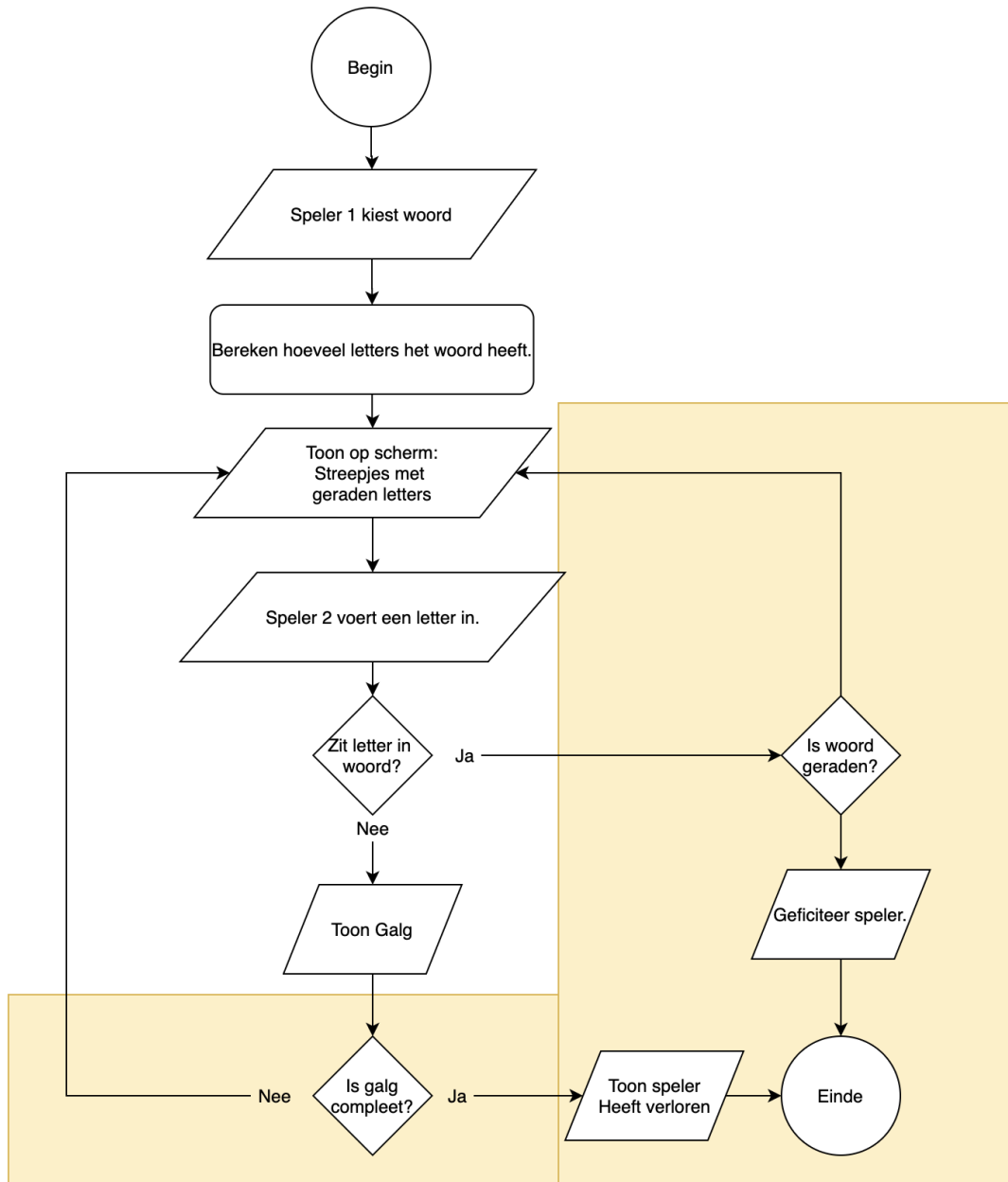


2.3.3 Stap 3 – Controle op winst en verlies

Dit wordt herhaald tot ofwel speler 2 alle letters heeft genoemd, het woord heeft geraden en zo het spel wint, of als speler 2 te veel foute raadpogingen doet waardoor de galg wordt voltooid en speler 2 "hangt".

Als de letter goed is gekozen dan kunnen we kijken of het hele woord is geraden. Zo niet dan mag de speler nog een keer raden.

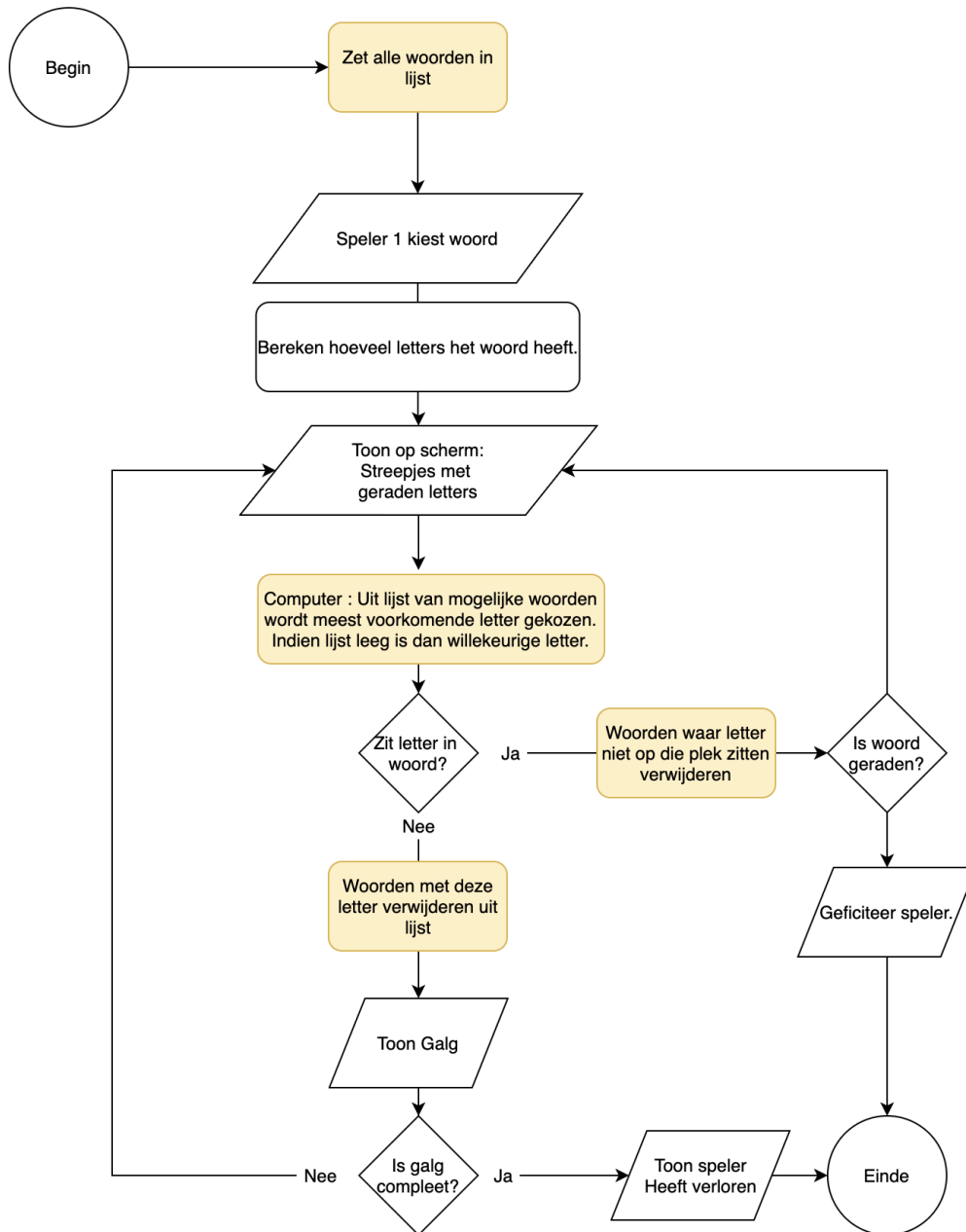
Als de speler fout heeft geraden dan werd de galg al met een stap uitgebreid. Dit is het punt om de galg te controleren of deze al compleet is. Als dat zo is dan heeft de speler verloren, zo niet dan mag de speler nog een keer een letter kiezen. In het stroomdiagram heb ik me niet beziggehouden met hoeveel beurten je precies mag gebruiken. Dit kan je erin zetten, maar je kunt ook dit vager beschrijven zodat je die keuze voor een verdere verfijning van het stroomdiagram houdt.



Figuur 13: Complete stroomdiagram galgje

2.3.4 De computer laten reden.

Als je het spel zou willen omdraaien, en dat je zelf een woord verzint, en de computer moet het woord raden dan moet je de computer wil “Slim” maken. Want willekeurige lekkers raden heeft geen zin. Je zou kunnen beginnen dat de computer een lijst van heel veel woorden inleest. De letter die heft meeste voorkomt wordt door de computer gevraagd. Als de letter niet voorkomt worden alle woorden met die letter verwijderd. Vervolgens ga je weer zoeken in de over gebleven lijst. Komt de letter wel voor, dan weet je ook op welke plek. Alle woorden waarbij de letter niet op die plek staat worden verwijderd. Vervolgens ga je weer zoeken in de over gebleven lijst. In het onderstaande stroomdiagram kan je de wijzigingen zien die gemaakt zijn om dit slimme te maken.



2.4 Opdrachten

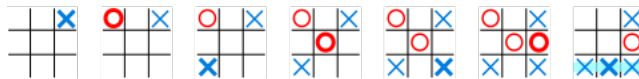
2.4.1 Hoger Lager

Maak in DRAW.IO het stroomdiagram van het spel Hoger/Lager. Als eerste wordt op het scherm de tekst neergezet dat de speler zijn naam moet intypen. Nadat de speler dat gedaan heeft kiest de computer een getal tussen de 1 en 20. De speler kiest een getal. Als het getal overeenkomt met het getal dat door de computer gekozen is wordt de speler met zijn naam in de tekst gefeliciteerd en is het spel klaar. Indien het getal niet is geraden dan krijgt de speler de melding of het getal te hoog of te laag was. De speler mag dan nog een keer raden. Er mag maar in totaal 5x door de speler geraden worden. Als het getal dan nog niet geraden is krijgt de speler de melding dat hij verloren heeft, en het spel is ten einde.

2.4.2 Boter-Kaas-en-Eieren

Maak in DRAW.IO het stroomdiagram van boter-kaas-en-eieren.

Als je de spelregels niet weet staat het onderstaande in Wikipedia. In het stroomdiagram speelt de mens tegen de computer. De mens mag op ieder leeg veld een stuk zetten. De computer moet slim zijn en een slimme strategie hebben waar hij het stuk zet. Voor het gemak mag je ervan uit gaan dat de speler altijd kruisje heeft, en de computer het rondje. Aan het begin van het spel moet je wel even bepalen wie als eerste mag (random).



Boter-kaas-en-eieren is een eenvoudig spel voor twee spelers. Hoewel 'boter, melk, kaas', 'oo maal oo', 'kruisje rondje' en 'tik-tak-tor' ook voorkomen, is boter-kaas-en-eieren in het Nederlands de meest gebruikte naam voor het spel.

Het spel wordt met een potlood of pen op papier gespeeld. Het spel is ook op een computer te spelen, soms zelfs bij een chatprogramma, zoals MSN Messenger. Bordspellen bestaan ook, maar deze hebben vaak meer een decoratief doel.

Boter-kaas-en-eieren wordt gespeeld op 3 bij 3 velden. Bij het begin zijn alle velden leeg.

De ene speler zet een 'kruis' en de andere speler een 'rondje'.

Degene die drie van zijn eigen tekens op een rij heeft, dat mag diagonaal, verticaal of horizontaal zijn, heeft gewonnen. Iemand die het spel goed genoeg kent, kan het in ieder geval in een gelijkspel laten eindigen, ongeacht wat de tegenstander doet.

Zeker voor beginners is het veld in het midden het belangrijkste veld.

....

Het aantal verschillende mogelijke bordposities is 764, het lege bord niet meegerekend. Posities die door roteren of door spiegelen hetzelfde zijn, tellen daarbij voor een. Het aantal toegestane zetten vanaf het lege bord en daarna tussen deze bordposities is 2096. Zo zijn er drie verschillende mogelijke begin zetten: in het midden, in de hoek of in het midden van een rand.

Voor deze eerste versie van het spel mag je de zetten van de computer een proces maken met de tekst : De computer doet een slimme zet.

2.4.3 Boter-Kaas-en-Eieren slimmer maken

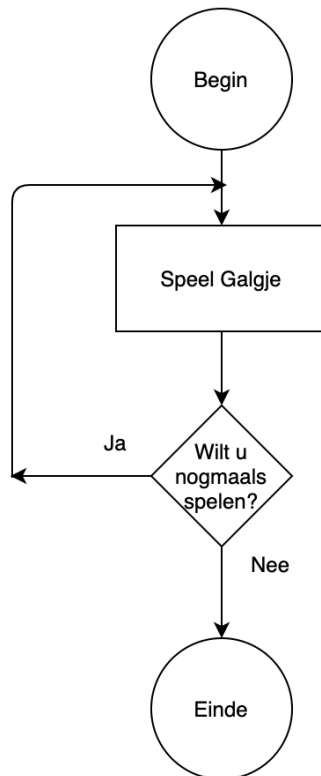
Het stukje dat je in opdracht 2.4.1 hebt genoemd “De computer doet een slimme zet.” Kan je nu wijzigen om daar iets slims neer te zetten. Wat zijn de stappen die de computer moet doen om de slimme zet te bepalen. Het zijn 5 stappen die moeten worden doorlopen. Verzin de stappen, en verander je stroomdiagram met deze stappen.

1. Is er een zet waarbij de computer zelf kan winnen?
2. ?
3. ?
4. ?
5. Kies een lege zijkant?

3 Complexe Stroomdiagrammen

Stroomdiagrammen kunnen steeds ingewikkelder worden naarmate het programma steeds groter wordt. Het is daarom belangrijk dat in het stroomdiagram alleen beschreven wordt wat gevraagd wordt. Een mooie uitbreiding voor het spel galgje is dat er wordt gevraagd aan de speler of deze nog een keer wilt spelen. Deze vraag kan weer in het stroomdiagram van "Figuur 13: Complete stroomdiagram galgje" toegevoegd worden.

Een betere oplossing is om dat stroomdiagram te laten wat het is. Galgje is eigenlijk een proces. Het symbool voor een proces is veel eenvoudiger dan het gehele stroomdiagram.



Figuur 14: Wilt U nog een keer spelen?

Op deze manier kan je dan heel eenvoudig je stroomdiagrammen hergebruiken. Of je kan een proces later uitbreiden door voor dat proces een stroomdiagram te maken. Alleen processen kan je op deze manier verfijnen. Als je de keuze in het oorspronkelijke stroomdiagram gezet zou je nooit een spel kunnen maken waarbij verschillende spelen achter elkaar gespeeld moeten worden zoals weergegeven in "Figuur 11: Opstapeling van keuzen." op bladzijde 11.

Zonder dat we een regel kunnen programmeren kunnen we ook een stroomdiagram maken voor het stuk "Speler kiest letter" in het stroomdiagram "Stap 3 – Controle op winst en verlies" op pagina 16 als daar de computer de invoer zou geven.

3.1 Opdrachten

3.1.1 Keuze van programma's

Je hebt nu het stroomdiagram van galgje en Boter-Kaas-en-Eieren gemaakt. Maak in Draw.io een stroomdiagram dat een programma beschrijft die de keuze tussen de spelletjes doet. Er wordt een keuze tekst op het scherm neergezet. Vervolgens kan de speler kiezen voor 1 om Galgje te spelen, en 2 om Boter-Kaas-en-Eieren te spelen. Als 3 wordt ingegeven dan stopt het programma. Bij alle andere keuze wordt weer het keuzemenu op het scherm getoond. De spelletjes kan je zien als complexe stroomdiagrammen.

3.1.2 Yahtzee

Maak in DRAW.IO een stroomdiagram van het spel Yahtzee. De spelregels zijn te vinden op : <https://www.dobbelstenenshop.nl/spelregels-yahtzee/>

Dit is een heel uitgebreid spel. Daarom moet je als eerste gaan kijken of je het spel in stukjes hakt. Ten eerste het gooien met de dobbelsteen. Je mag drie keer gooien. Dit drie keer gooien kan je ook als een proces zien. De keuze of je 1. 2 of drie keer gooit is of het zin heeft. Dit kan je hierin gewoon als keuze zetten.

Verder moet het score bord ingevuld worden. Het stroomdiagram van yahtzee hoeft maar van 1 speler, want je tegenstander heeft geen invloed op jouw stroomdiagram. Je kan dan laten kiezen of je boven of op het onderste gedeelte de score invult.

De totale puntentelling kan altijd gedaan worden. Indien boven het aantal punten groter of gelijk is aan 63, komen er nog 35 bonuspunten bij.

Het werpen van de dobbelsteen duurt net zolang totdat er geen lege vakjes meer zijn waar een score ingevuld kan worden. Er mag altijd een lege score ingevuld worden.

3.1.3 Stroomdiagram Boter-Kaas-en-Eieren slimmer maken

In 2.4.3 Boter-Kaas-en-Eieren slimmer maken heb je de stappen neergezet hoe je de computer kan laten nadenken wat die moet doen. Je kan van dat stuk nu een apart stroomdiagram maken.

3.1.4 Paper-Scissors-Rock

Op internet kan je de hele geschiedenis en spelregels van dit spel vinden. https://en.wikipedia.org/wiki/Rock_paper_scissors.

Maar een stroomdiagram van alle varianten van dit spel.