

Scratch (23)

René Suiker

Er is weer ruimte

Vorige keer moest ik flink woekeren met de ruimte. We hadden in eerste instantie te weinig kopij, een probleem waar we wel eens vaker mee kampen. Maar ineens was daar een interessant artikel over videobewerking en dat komt eigenlijk te weinig aan bod, dus dat wilden we wel plaatsen. Maar daardoor kreeg ik ineens wel heel weinig ruimte voor Scratch. Enfin, ik denk dat de oproep tot eigen activiteit helemaal zo slecht nog niet was, zo in de vakantie. Maar nu maar eens kijken of het ook allemaal gelukt is.

Het huiswerk

Ik besef wel, dat de opdrachten best pittig waren, maar ja, we hebben ook wel heel wat behandeld. De opdrachten moesten je aan het denken zetten. Hoe pak je zoiets nu aan? Als ik niet weet hoe ik een probleem op moet lossen, begin ik altijd te denken in hapklare brokken. Daarna ga ik dan eens kijken of ik die blokken uit kan werken. En als dat niet gelukt, ga ik een moeilijk blok weer opsplitsen in hapklare brokken. En zo ga ik door, totdat het probleem is opgelost (of totdat blijkt dat het probleem niet op te lossen valt).

En tijdens dat splitsen en uitproberen kom je ook situaties tegen die nopen tot wat testen. Want je kunt nog niet alles weten, en dat maakt het ook leuk. Je kunt gaandeweg ontdekken en door dat te doen en te testen in wat we noemen 'een gecontroleerde omgeving' leer je steeds meer van in dit geval Scratch, maar het geldt in feite voor elke programmeertaal en in feite ook voor heel veel configureerbare applicaties.

Elektronica bouwdoos

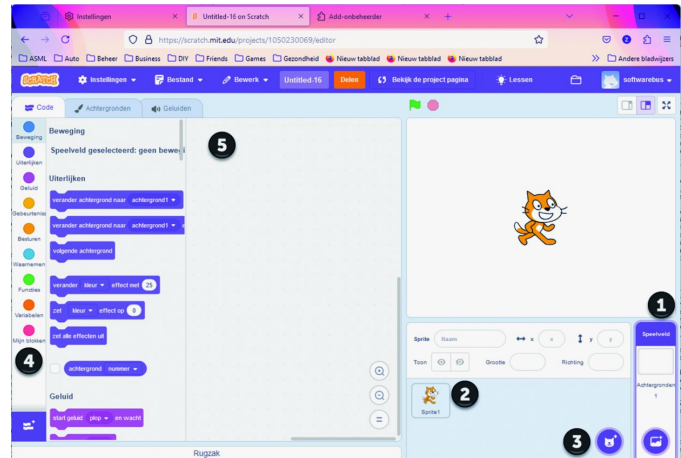
Dit was toch wat moeilijker dan het leek. Normaal beweeg je elementen over het scherm bij het bouwen van een programma en als je het programma laat lopen, dan zorgt het programma voor het bewegen van elementen. Hierbij kan de gebruiker wel invloed uitoefenen op de beweging, maar items ergens plaatsen waar we ze willen hebben tijdens het gebruik van de applicatie, dat hebben we nog niet gezien.

Daarom moeten we om dit mogelijk te maken eerst wat experimenteren. We gaan eerst even de bouwblokken en ook een achtergrond tekenen.

Beginnen

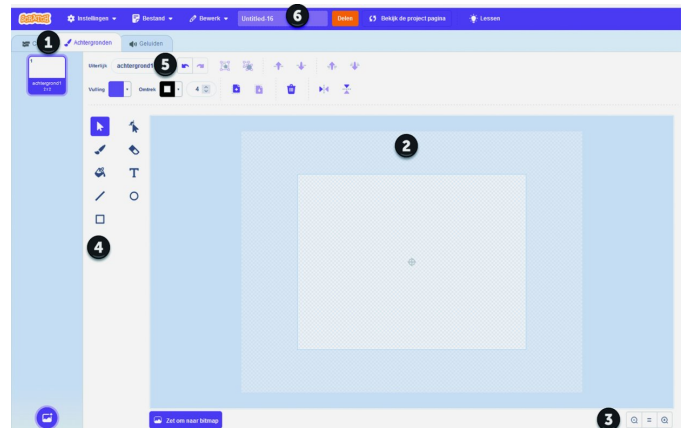
We starten dus Scratch op (<https://scratch.mit.edu>) en klikken bovenin op 'Maak'. Eventueel moet je natuurlijk de taal op Nederlands zetten, maar dat wordt nu wel als bekend verondersteld.

Herkent u het nog? Bij (1) (Zie volgedede kolom) kunt u het speelveld selecteren, dat gaan we zo doen. Bij (2) ziet u de sprites, in dit geval de kat, die hebben we voor dit programma nog niet nodig, maar laat nog maar even rustig staan. Bij (3) kunt u nieuwe sprites toevoegen, dat gaan we zo doen. Bij (4) kies je de thema's voor de codeblokken en onderin staat "mijn blokken", die gaan we ook nog wel gebruiken. En in het werkveld (5) gaan we dan het spel opbouwen.



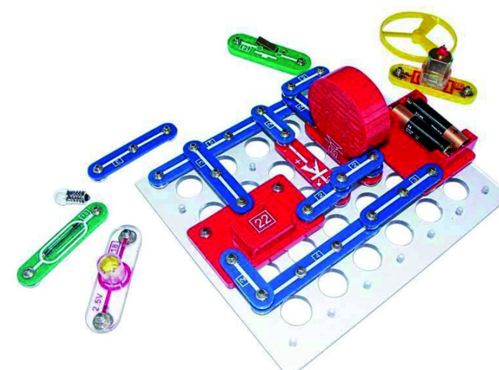
Figuur 1 - Scratch startscherm

Klik dus maar op Speelveld (1). Op dit moment is er nog maar één (lege) achtergrond en daar gaan we wat mee doen.



Figuur 2 - Achtergrond

Zoals op veel plaatsen in Scratch heb je bij (1) drie tabbladen. We kiezen nu voor 'Achtergronden', want die gaan we maken. De 'Code' en 'Geluiden' komen later pas aan de orde. Vooralsnog dus de achtergrond. Bij (2) kan je zien hoe het er nu uitziet. Dat is helemaal leeg, want we hebben nog niets gedaan, maar dat gaat zo veranderen. Bij (3) kan je het speelveld in- en uitzoemen. Bij (4) kan je je tekengereedschappen selecteren. Daar gaan we zo mee stoeien. Bij (5) kunnen we de achtergrond een naam geven. Nu heet die nog 'achtergrond1', maar maak hier maar 'Basisplaat' van, want die gaan we tekenen



Bij (6) kan je trouwens je project nog een naam geven, ik stel voor om hier 'Bouwdoos Elektronica' in te vullen.

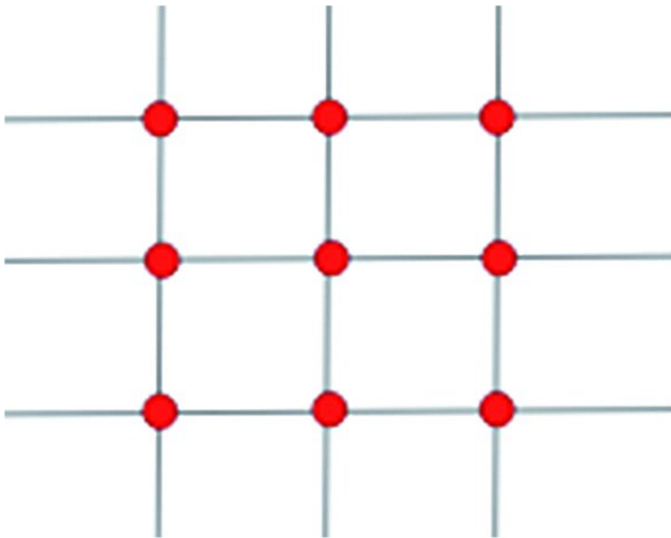
Figuur 3 - Elektronica bouwdoos

Nog even in de herinnering, of misschien wel voor het eerst als u net pas aanhaakt, de bedoeling is dat we dit type bouwdoos gaan nabouwen:

Teken nu met rechte lijnen die het veld bestrijken een raster, zodanig dat de onderlinge afstanden even groot zijn. De bedoeling is, dat we op de knooppunten van het raster de contactpunten gaan aanleggen. Als we dat gedaan hebben mogen de (hulp)lijnen weer weg, maar ik wil dat het raster er een beetje netjes uitziet, zodat we de componenten steeds passend kunnen plaatsen. Je kunt de hulplijnen natuurlijk ook netjes laten zitten.

In de bouwdozen zitten ook extra losse contactpuntjes, om eventuele hoogteverschillen te compenseren, omdat de componenten een fysieke dikte hebben. Dat voegt een extra dimensie toe, die we in onze simulatie ook kunnen gebruiken, maar op dit moment hebben we al genoeg problemen op te lossen, dus dit is voor later zorg.

Als je tevreden bent met je raster kan je de achtergrond editor verlaten. Voor dit doel heb ik me even beperkt tot een klein stukje, waarmee we wel wat zaken kunnen testen, maar de volledige opdracht niet kunnen maken.



Figuur 4 - Raster

Vervolgens moeten we onze componenten gaan tekenen. Deze zijn geen onderdeel van de achtergrond, maar moeten er overheen kunnen bewegen, het worden dus 'sprites' in Scratch-terminen. Ze moeten straks wel ergens bij het basisblok liggen om gepakt te worden. Hierbij kan je het voordeel van Scratch uitnutten door elk component oneindig vaak te laten gebruiken, maar dat raad ik zeker in het begin nog niet aan. Want als straks alles geplaatst is, moet er gerekend gaan worden of de stroom al dan niet gaat lopen. In eerste instantie zou ik dat nog niet al te ingewikkeld willen maken. Zoals in de opdracht weergegeven, de volgende componenten moet gesimuleerd kunnen worden:

- Doorverbinding in verschillende lengtes
- Batterij pakket, met plus- en minpool
- Weerstand
- Schakelaar (aan/uit)
- Drukknop (aan/uit)
- Diode/LED
- Lampje
- Motor

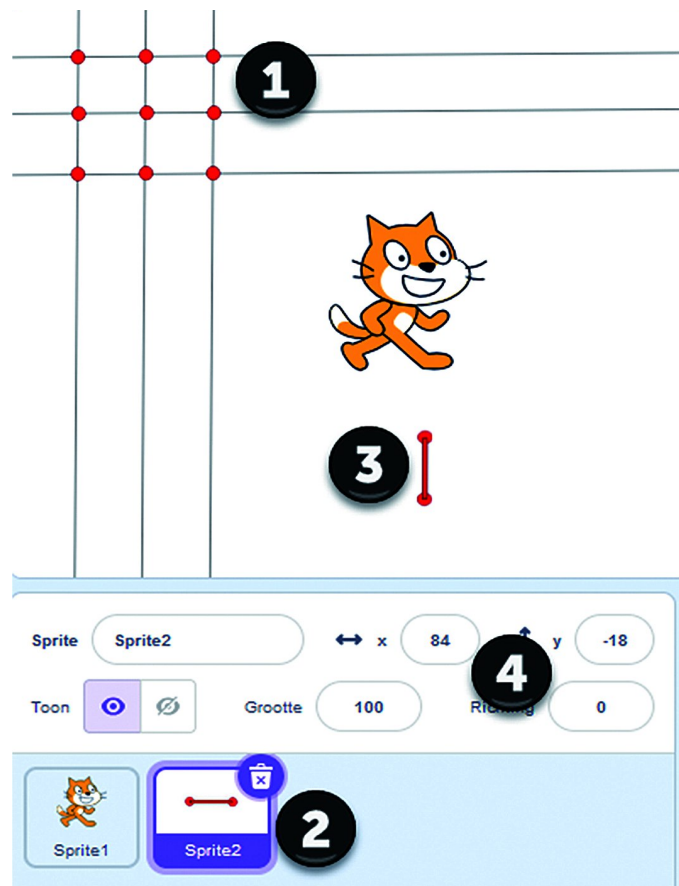
De volgende uitdagingen staan ons hierbij te wachten:

1. De 'artiest' moet de componenten tekenen;
2. De componenten moeten door de gebruiker gepakt kunnen worden en op hun plaats in het schema komen;
3. De bedieningselementen moeten op een bepaalde manier bediend kunnen worden;
4. De stroom moet gaan lopen, als er een correct elektrisch circuit is;

Dat is de basis, maar je kunt natuurlijk e.e.a. zo uitgebreid maken als je wilt.

Componenten tekenen

Het is de bedoeling dat de componenten contactpunten hebben, die aansluiten met de contactpunten op de basisplaat. En als ze aansluiten, dan veronderstellen we contact. Het is dan ook de bedoeling, dat de componenten hetzij horizontaal, hetzij verticaal contact maken met 2 of meer contactpunten. De afstand tussen de contactpunten op de componenten moet dus overeenkomen met de afstanden op de basisplaat. Dat is een uitdaging, vooral als je niet zo'n geweldige tekenaar bent. Maar ik ga het proberen. Je klikt dus op het symbool rechts onderin voor een nieuwe sprite en dan kies je voor 'teken zelf'. Ik probeerde dus eerst de contactpunten te kopiëren van de basisplaat en deze op de juiste afstand te zetten. Ik ben geen artiest en zette dus gewoon een rechthoek tussen de twee punten en keek of ik ze als sprite zowel horizontaal als verticaal kon laten aansluiten en dat lukt:



Figuur 5 - Doorverbinding

Bij (1) zie dus het raster, bij (2) de nieuwe sprite. Je ziet deze in het speelveld bij (3) en dan kan je kijken of hij past. Als je 'm beweegt en je raakt 'm kwijt dan kan je bij (4) weer de x- en y-positie instellen en ook de richting. Om het allemaal blijvend te laten passen moet je niet aan de grootte komen en eigenlijk alleen maar in veelvoud van 90 graden kunnen draaien (voor een doorverbinding zijn maar twee richtingen nodig, maar komen er straks complexere elementen, dan heb je misschien meer mogelijkheden nodig, zoals voor een diode). Gelukkig kan je tijdens het spelen niet aan de grootte en draairichting veranderen, behalve door specifieke opdrachten die je als bouwer volledig in de hand hebt. Omdat ik ook nog even aan de andere opdracht wil snuffelen wil ik het hier op dit moment wel een beetje bij laten. Maar voor huiswerk voor een korte periode was het misschien wat te veel gevraagd; voor een soort studieproject is het denk ik wel een fraaie uitdaging. Ik ben er dus toch wel blij mee, maar niet heel verbaasd dat niemand een oplossing instuurde. Dan toch alvast wat huiswerk over dit onderwerp.

Opgave 23.1

Bedenk een manier waarop tijdens het spelen de gebruiker het object op kan pakken, kan draaien en neer kan leggen. Een 'auto-snap' om 'm automatisch met de contactpunten door te verbinden zou helemaal te gek zijn.

Opgave 23.2

Bedenk alvast hoe je zou willen vaststellen of er een correct elektrisch circuit is gevormd

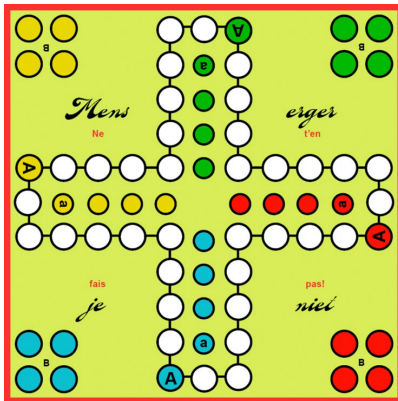
Mens erger je niet

Toen ik dat bij mijn oma speelde stond er nog 'Mensch erger je niet' op de doos. Dat was al niet meer echt gebruikelijk, maar ze hadden de doos al wat langer. In die tijd waren er ook nog niet zo veel mooie spellen als nu het geval is. Ik weet dat ik bij opa en oma de keuze had tussen 'Mensch erger je niet', 'Ganzenbord' en 'Dammen'. En kaarten of dobbelen natuurlijk, en ze hadden ook 'Halma' maar niemand wist hoe je dat moest spelen en er was ook nog geen Internet om de spelregels op te zoeken. Intussen hebben wij alleen al zo'n 200 bordspellen thuis en zijn we echt nog geen topscorers, verre van. Af en toe doen collega's er zo eens 100 weg, omdat ze die niet meer spelen. Enfin, er is dus een enorm aanbod. Maar het voorliggend spel is in zijn simplicitéit leuk voor vooral kleinere kinderen en biedt al genoeg uitdaging als we dit in Scratch willen nabootsen.

Hoe begin je?

Wat hebben we nodig:

1. Een speelbord
 2. Een dobbelsteen
 3. Een aantal pionnen, 4 per speler
- Het speelbord halen we van Internet:

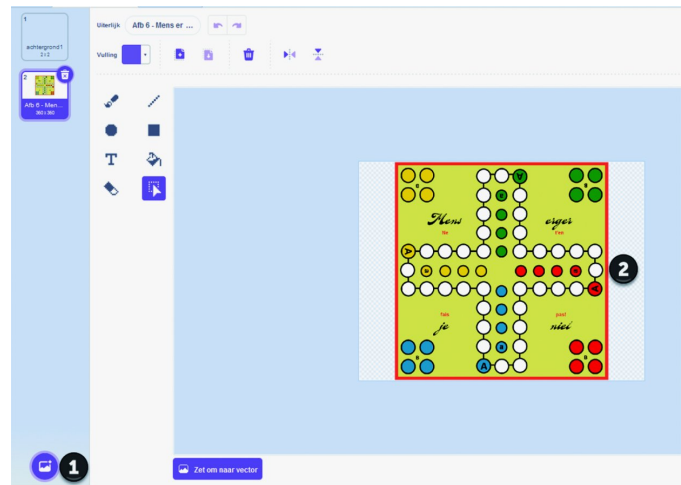


Figuur 6 - Mens erger je niet

Dit is de vierpersoonszijde van het bord. Er is ook een variant voor zes spelers en in principe ben je natuurlijk met een computerspel niet echt gelimiteerd, je kunt het gegeven ook uitvoeren voor 120 spelers. Of het leuk is mag je zelf uitmaken, ik stel gewoon voor met de vier personen te beginnen.

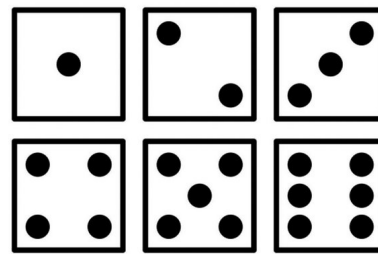
Sla dus even je vorige project op en ga weer naar de startpagina van Scratch en klik weer op 'Maak'. We kunnen nu een nieuw spel gaan maken en dat is precies wat we nu gaan doen. De vorige keer liet ik een plaatje zien met perspectief, maar bedenk dat Scratch toch echt wel een tweedimensionale wereld kent, dus ik zou het ook nu weer niet te ingewikkeld maken. We gaan het speelbord van figuur 6 opnemen in onze achtergrond.

We kiezen bij (1) de mogelijkheid van uploaden van een achtergrond en vervolgens zie je in (2) dat de achtergrond geplaatst is. Op dit moment is hij beeldvullend. Je kunt hem dus iets kleiner maken, zodat je er wat ruimte naast hebt voor bijvoorbeeld de dobbelstenen, maar je kunt er natuurlijk ook voor kiezen om de resultaten van de worp weer te geven bij de actieve speler. Ik laat het even zo. We hebben een speelbord.



Figuur 7 - Achtergrond inladen

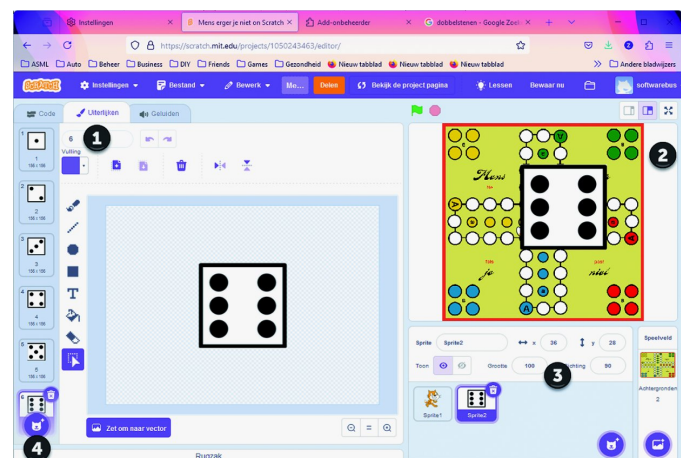
Het spel wordt gespeeld met één dobbelsteen. Het spel is puur tweedimensionaal. Ik kan dus volstaan met eenvoudige dobbelstenen, bijvoorbeeld zoals dit:



Figuur 8 - Dobbelstenen

Met een tekenprogramma knip je hier zes uiterlijken van voor één sprite 'Dobbelsteen'. Als je een beetje op de maat let kan je deze ook nog in het midden van het bord kwijt. Je zet 'm als sprite daar neer en laat 'm vervolgens verdwijnen. Als je dan moet werpen, laat je 'm verschijnen, wisselt een aantal keren van uiterlijk en weet dan welk uiterlijk boven ligt en dat is de waarde van de worp.

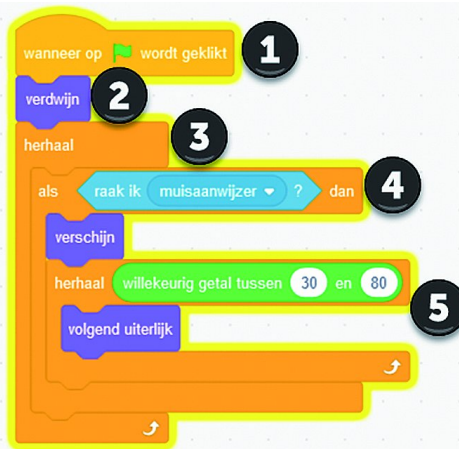
We maken dus even een sprite 'Dobbelsteen' met 6 uiterlijken.



Figuur 9 - Sprite 'dobbelsteen'

We moeten de sprite nog even 'dobbelsteen' noemen, maar je ziet één van de uiterlijken voorliggen. Bij (1) zie je dat dit uiterlijk gewoon '6' heet, want zo heb ik de diverse files ook genoemd, gewoon de waarde als naam. Je ziet bij (2) dat de nieuwe sprite op het speelbord is gepositioneerd. Je ziet ook dat we bij (3) iets aan de grootte moeten doen, want dit is niet echt fraai. Het moet echt flink kleiner, maar daar kan je mee stoeien. Gelukkig voer je hier de waarde voor de sprite in, niet voor een specifiek uiterlijk. In mijn bestanden zijn alle uiterlijken 156 bij 156 pixels. Als ik 'm naar 25% verklein dan is hij nog heel duidelijk, maar niet storend in beeld.

Dan hebben we een stukje code nodig om te dobbelen:



Figuur 10 -
Dobbelen

Deze code is opgenomen bij de sprite 'dobbelsteen' (intussen heb ik die naam wel aangepast). Dit stukje code is nu een test, dus bij (1) heb ik 'm laten initië-

ren door de groene vlag, maar als we eenmaal het spel af hebben, dan weten jullie dat ik doorgaans maar één stuk code op de vlag laat reageren en de rest bestuur met behulp van signalen. Dit om ervoor te zorgen, dat de volgorde duidelijk is en niet te veel van toeval afhankelijk.

Het verdwijnen (2) is eigenlijk niet nodig, dus kan je net zo goed weglaten. Maar het was even een testje dat ik tussendoor uitvoerde om even zeker te weten dat het systeem deed wat ik verwachtte: als de sprite niet zichtbaar is, dan kan de sprite nog steeds wel signaleren dat de muisaanwijzer erbij is.

Bij (3) begin ik de eeuwige lus. Hier moeten we straks iets subtieler mee omgaan, want je wordt geacht pas te dobbelen als de vorige beurt is afgewikkeld, maar dat terzijde. Bij (4) kijk ik als dobbelsteen of de muisaanwijzer bij me staat, hetgeen het signaal is om te dobbelen. Als we verder zijn in het spel, dan staat dit dus niet altijd aan, maar is deze trigger op zich wel goed. Je beweegt met de cursor naar de dobbelsteen om te dobbelen.

Bij 5 staat de 'worp' zelf. We veranderen een willekeurig aantal keren van uiterlijk, dus je ziet de dobbelsteen niet rollen maar je ziet wel de waarde continu veranderen, totdat hij stilstaat. Dat is dan de worp. En die worp kan je in een variabele plaatsen zonder te moeten bijhouden hoeveel keren het uiterlijk is veranderd. Scratch weet welk uiterlijk voorligt en dat is de worp:



Figuur 11 - Dobbelen
verbeterd

Ik heb een variabele 'worp' gedefinieerd, die algemeen beschikbaar is, want je moet hem straks bij de pionnen ook kunnen gebruiken. Als de dobbelstenen zijn uitgedraaid, dus buiten de middelste lus, maar binnen de eeuwige lus, maak je deze variabele gelijk aan het uiterlijk nummer. En deze worp kan je in de verdere afhandeling gebruiken.

Denk nu eens na over hoe je verder zou willen gaan, want mijn ruimte is weer op.

Opgave 23.3

Laat pionnen over het bord schuiven. Welke problemen ervaar je?