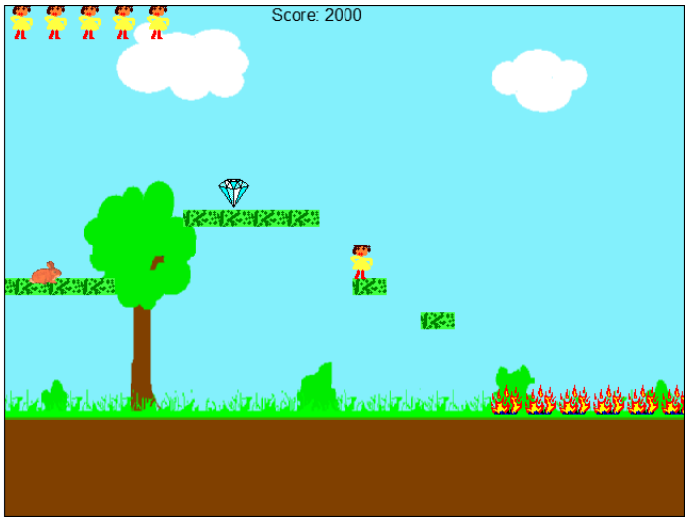
Deel 3



Het project Gamemaker Kids is mede mogelijk gemaakt door:

VSBfonds

Auteur: Klasien Postma

Eindredactie: Sacha de Ruiter, Technika 10 Nederland

Copyright: Creative Commons

www.gamemakerkids.nl/index.pl/licentie

Uitgave: September 2007

Inhoudsopgave

Hoofdstuk 1: Inleiding 4
Hoofdstuk 2: Belangrijk 5
Hoofdstuk 3: Je eerste level 6
De basis van je spel: ik-figuur, bodem, doel 6
Room: achtergrond beweegt met de hoofdfiguur mee 8
Hoofdstuk 4: De ik-figuur programmeren 10
Gebruik pijltjestoetsen programmeren 10
De ik-figuur laten springen 12
Obj_doel programmeren 16
Hoofdstuk 5: Beloningen en hindernissen 17
Het object bereikt de rand van het speelveld en dan verdwijnt het object 17
Het object verdwijnt van het speelveld en komt aan de andere kant weer op 18
Met een bom een muurtje opblazen 19
Levens en punten bewegen met het scherm mee 26
Hoofdstuk 6: Je spel afmaken 29
Hoofdstuk 7: Tenslotte 30
Overzicht van cursusboeken Game Maker 30
Hoofdstuk 8: Bijlage - testschema voor de eindtest 31

Hoofdstuk 1: Inleiding

Game Maker is een programma waarmee je games kunt maken, zonder dat je daarvoor een programmeertaal hoeft te leren.
Op een heel simpele manier kun je toch goede games maken.
Het programma is gemaakt door de Utrechtse professor informatica Mark Overmars.

Het cursusmateriaal voor de cursus **Game Maker voor kids** bestaat uit 3 delen.
Zorg dat je eerst deel 1 en 2 hebt doorgewerkt voor je aan deel 3 begint.
In deel 1 van de cursus Game Maker leer je het programma Game Maker een spel kunt maken. De nadruk ligt vooral op hoe het programma Game Maker werkt. In deel 2 ontwerp en maak je zelf een doolhofspel.

De opzet van het cursusboek deel 3 is anders dan die van de delen 1 en 2.
Cursusboek deel 3 is bedoeld als een naslagwerk voor het maken van platformgames. Alles wat je in deel 2 geleerd wordt in dit deel niet herhaald, maar kun je wel toepassen bij het maken van een platformgame.
Op **www.gamemakerkids.nl** kun je een voorbeeld van een **platformgame** downloaden en spelen.

We gaan er van uit dat op jouw computer Game Maker versie 7.0 al is geïnstalleerd.
Verder is er een map GameMaker geïnstalleerd. Daarin staan de mappen GamemakerBestanden, MijnGames en Voorbeeldgames. Is dat niet zo, download dan de installatiehandleiding en lees daar hoe dat moet. Kom je er niet uit, vraag om hulp.
Veel plezier!!

Hoofdstuk 2: Belangrijk

Vergeet niet steeds bij elke verandering het volgende stappenplan:

1. **Ontwerp** maken
2. Spel **onder een nieuwe naam** opslaan
3. Je ontwerp **programmeren**
4. Je bestand **opslaan**
5. Je **testschema** bijwerken met nieuwe testregels
6. Aan de had van je testschema **testen** of je aanpassing goed werkt
7. Pas als je tevreden bent over je aanpassing, dit lijstje weer voor de volgende aanpassing doorlopen.

Hoofdstuk 3: Je eerste level

De basis van je spel: ik-figuur, bodem, doel

Net als bij het doolhofspel in cursusboek deel 2 begin je met het ontwerpen en bouwen van twee levels in de meest eenvoudige vorm. Later ga je het spel leuker en mooier maken met een geluiden, een leuk startscherm, meer uitdagingen, bewegende sprites enzovoorts.

In dit boek zijn de plaatjes overgenomen uit het voorbeeldspel in de map Voorbeeldgames: voorbeeld-avonturenspel.

Wat heb je minimaal nodig voor een platformspel?
Dat lijkt erg op dat wat je minimaal nodig hebt voor een doolhofspel.
Voor een platformspel zijn dat:

- de **ik-figuur**
- **'bodems'** om op te springen
- **doel**

1. Maak **sprites** voor deze 3 objecten
2. Maak **objecten** met de 3 sprites

Hoewel het voor het spel niet per se noodzakelijk is, maakt een achtergrond bij een avonturenspel het spel wel veel mooier.

3. Maak in **Paint** een hele simpele **achtergrond** van **800** bij **600**. Een paar gekleurde vlakken is vaak al genoeg. De vulling doe je met objecten. Lees de paragrafen hieronder om te weten waar je op moet letten.

Ontwerp een tekening alsof je behang ontwerpt

De linkerkant en de rechterkant moeten precies aan elkaar passen. Als de achtergrond van links naar rechts beweegt wordt het plaatje namelijk steeds herhaald, net als bij behang.
Maak je een platformgame waarbij de achtergrond van boven naar onderen beweegt of andersom, dan moet je er voor zorgen dat de bovenkant en de onderkant aan elkaar passen.

Zichtbare of onzichtbare bodems gebruiken

Je kunt in Paint een mooie maar simpele achtergrondtekening maken en dan de 'bodem-objecten' onzichtbaar maken door bij het object het vinkje bij visible weg te halen.
Je kunt ook meer aandacht besteden aan de sprites voor de bodem en die wel zichtbaar laten. Voordeel van zichtbare sprites is dat je dan precies ziet waar de bodem is, in een tekening zit je dan vaak op halve hokjes wat problemen kan geven.

4. Maak via **Create a background** een achtergrond en **kies** de door jouw gemaakte **tekening**.

Room: achtergrond beweegt met de hoofdfiguur mee

Bij een platformspel, avonturenspel maar ook met andere spellen wil je meestal dat, wanneer de hoofdpersoon loopt, de achtergrond meebeweegt. Je hoofdpersoon kan dan niet het scherm uit lopen. Dit werkt alleen als je Game Maker in de Advanced mode(= *gevorderde modus)* zet. Hoe je dat doet zie je hieronder.

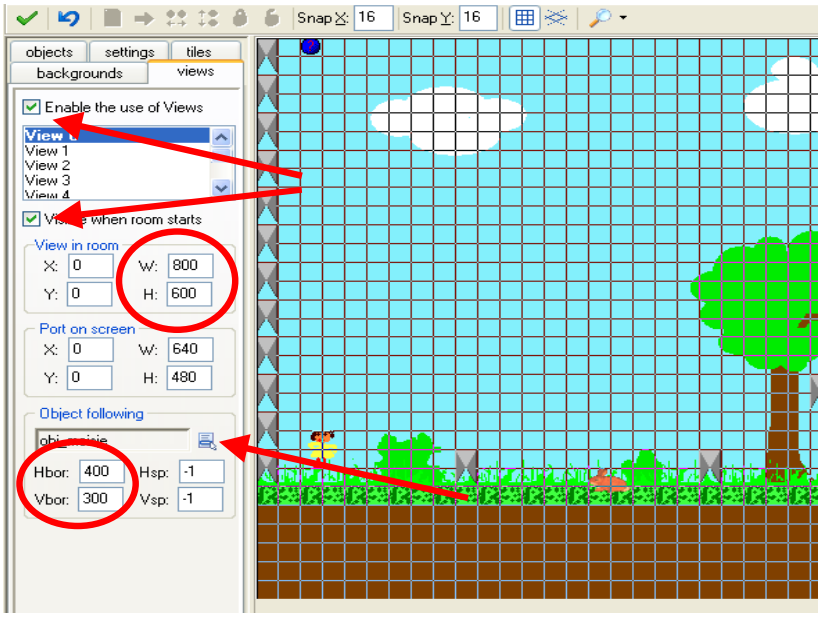
1. Zet in het menu **File** een vinkje voor **Advanced mode** (= *gevorderde modus).* Als je nog een spel had geopend wordt dat nu gesloten, dan pas wordt Game Maker in de Advanced mode gezet. Er verschijnen allerlei extra opties, laat je daardoor niet in de war brengen.
2. **Open opnieuw je spel**.
3. Maak met **Create a room** een nieuw level.
4. Je mag bij **Snap X** en **snap Y** nu ook **16** kiezen.
5. Klik op **tab Views**.

Met de instellingen bij Views bepaal je welk stuk van de Room je op je scherm ziet als je het spel speelt.

6. Zet een **vinkje** bij **'Enable the use of Views'**
7. Zet een **vinkje** bij **'Visible when the room starts'**
8. Vul bij **'View in room'** waarden in.
    a. Met de waarden **X** en **Y** bepaal je de **linkerbovenhoek** van je View wanneer je het spel start. Als beiden nul zijn, start je het spel met de view linkerbovenhoek.
    b. Met de waarden **W (=width) en H (=height),** bepaal je de breedte en hoogte van het scherm. Kies hier voor **800 en 600**.
9. Kies bij **'Object following'** je **ik-figuur**.

Als je besturingsobject nu naar rechts loopt en bijna van het scherm afloopt beweegt de achtergrond naar links, zodat je altijd je besturingsobject op het scherm ziet.

10. Vul bij **Hbor. 400 en bij Vbor. 300** in. Je besturingsobject blijft nu altijd midden in het scherm.



11. Kies **tab settings**. Vul je bij **Width** een **groot getal** in, dan wordt het landschap heel breed. Vul je bij **Height** een heel **groot getal** in, dan wordt je landschap heel hoog. Beide kun je natuurlijk ook heel groot maken.
12. Kies in tabblad **Background** de **achtergrond** die je hebt gemaakt.
13. Maak nu een **eerste level** met **bodemdelen**, je **ik-figuur** en het **doel**.

Hoofdstuk 4: De ik-figuur programmeren

Gebruik pijltjestoetsen programmeren

Aan de ik-figuur moeten Events en Actions worden toegevoegd, zodat deze iets 'doet'. In de vorige cursusboeken heb je steeds gebruik gemaakt van de acties Move Fixed.
Bij een platformspel, waar de ik-figuur ook schuin moet kunnen springen kun je beter gebruik maken van de actie **Jump to position** (=*spring naar een positie)* .
Het is nu ook wel nuttig te leren over het verschil tussen relative wel of niet aanvinken. Daarover komt eerst een stukje uitleg.

Verschil tussen relatief en niet-relatief

Bij veel acties moet je kiezen of je wilt dat een verplaatsing of snelheid relatief is of niet. Het verschil is dat als je **relative**(=*relatief)* aanvinkt de verplaatsing of snelheid **erbij** komt.
Stel je vult bij de actie jump to a position (=*spring naar een positie)* bij x=4 en y=0 in.
Vink je **relatief niet** aan dan springt het object naar positie x=4, y=1, dat is een plek ergens linksboven in het speelveld.
Vink je **relatief wel** aan, dan komt er bij de x 4 bij en bij de y komt er 1 bij. Het object springt dus 4 naar rechts en 1 omlaag.

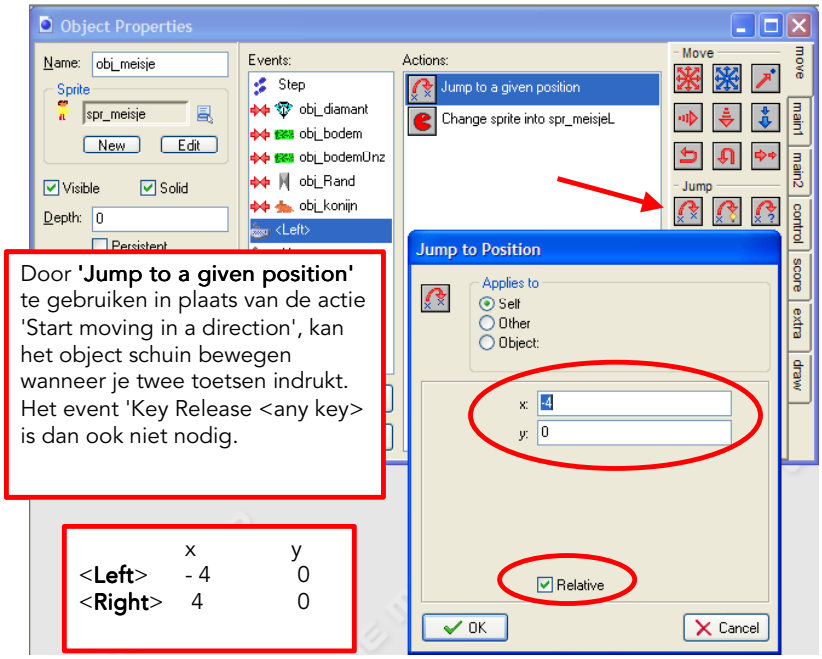
Hetzelfde gebeurt bij snelheid. Stel je hebt al een snelheid 10 en je kiest nu bij een gebeurtenis snelheid 1.
Vink je **relatief niet** aan, dan wordt je snelheid 1.
Vink je **relatief wel** aan, dan wordt je snelheid 1 *meer*, dus 11.

Bij event pijltjes toets de actie Jump to a position toevoegen

1. Programmeer event **<left>** en event **<right>.** Voeg de actie **Jump to Position** toe. In de figuur hieronder zie je bij welke event je welke waarden voor x en y moet invullen. Let op dat je **Relative** aanvinkt.

   Voordeel Jump to position is ook dat de actie stopt als je de toets loslaat, dus je hoeft niet een extra event 'Key Release <any key>' te programmeren.

   Wil je dat het object sneller beweegt, kies dan bij x of y bijvoorbeeld 6 in plaats van 4.



De ik-figuur laten springen

In een platformgame moet de ik-figuur ook ergens óp kunnen springen en mooi landen. Het 'mooi laten landen' doe je door de zwaartekracht te gebruiken. (Gravity) Het is namelijk in het echt ook zo, dat wanneer je springt, je op een gegeven moment vanzelf omlaag gaat. Dat komt door de zwaartekracht.

Dit is een moeilijke paragraaf. Lees heel goed en doe precies wat er staat.

Er moeten 3 events worden geprogrammeerd:

- Keyboard event **<up>**
- Event **Step**
- Event **Collission met een bodem**

Keyboard event <Up>

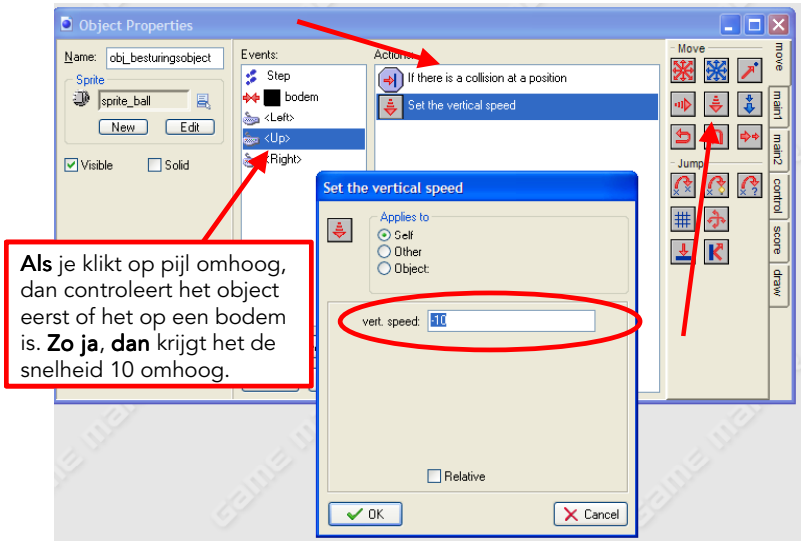
Als het object nog op de bodem ligt moet hij bij het indrukken van de pijltjestoets omhoog een snelheid 10 omhoog krijgen.

Ofwel als er een botsing is met een voorwerp *onder* het object, dan krijgt het object bij drukken op <up> een verticale snelheid 10 naar boven.

Dit programmeer je als volgt.

2. Voeg bij je besturingsobject het **Keyboard event** <**Up**> toe.
3. Sleep de volgende acties naar het tabblad Actions. En vul de bijbehorende waarden in. De acties vind je op de tabbladen Control en Move.
   - **'If there is a collision at position'** *(=wanneer er een botsing is op positie)*
     - **x=0**,
     - **y=1**, (y=1 *dus onder het object)*
     - **Objects**: **only solid**, (=*objecten: alleen massieve objecten)*
     - **Relative**: **aangevinkt**

- 'Set de vertical speed' =(*Stel de verticale snelheid op)*
    - **vert. speed: -10** (dus naar boven)
    - **Relative**: **niet** aangevinkt.



Step event *(= stap event, dus als het object iets doet)*

Als het object niet meer op de bodem ligt treedt de zwaartekracht in werking. Dus er moet worden gecontroleerd of onder het object geen bodem is.

Zo niet,

dan is de zwaartekracht omlaag 0,5, dan gaat het object steeds minder snel omhoog en zal uiteindelijk weer naar beneden vallen.

Is dat wel het geval,

dan is de zwaartekracht 0 en staat het object stil. Het moet namelijk niet door de bodem vallen.

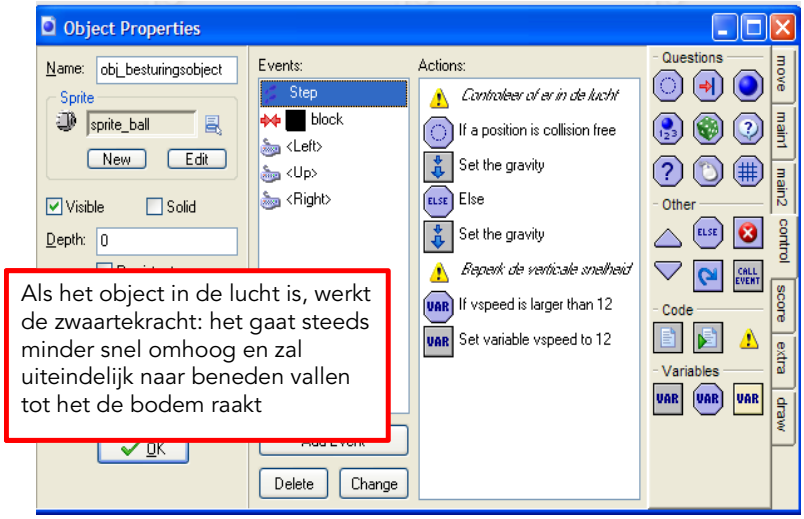
Dat programmeer je als volgt.

4. Voeg bij je **ik-figuur** het event **Step** toe.
5. Sleep de volgende acties naar het tabblad Actions. En vul de bijbehorende waarden in. De acties staan allemaal op het **tabblad Control**.

- **'Put some comment'** (*=commentaar, uitleg van wat er moet gebeuren*)
    - **comment: Controleer of er in de lucht iets is**
- **'If a position is collision free'** *(=wanneer een positie botsingsvrij is)*
    - vul bij **x 0** in,
    - vul bij **y 1** in *(dus onder het object)*
    - **Objects**: **only solid**, (*=objecten: alleen massieve objecten)*
    - **Relative aangevinkt**
- 'Set the gravity' *(=stel de zwaartekracht op)*
    - **direction: 270**
    - **gravity: 0.5**
    - **Relative**: **niet** aangevinkt
- 'Else' *(= anders)*
- **'Set the gravity'** *(=stel de zwaartekracht op)*
    - **direction: 270**
    - **gravity: 0**

Verder mag de snelheid niet groter worden dan 12. Dat programmeer je zo.

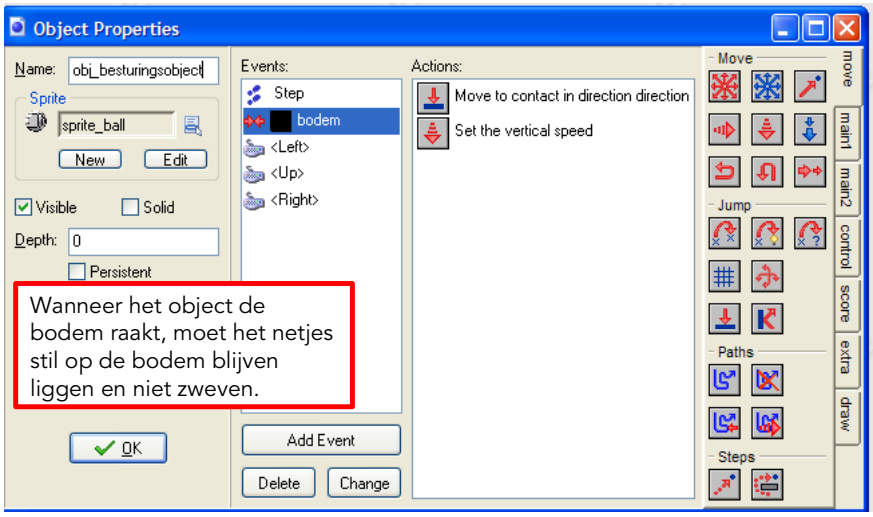
- 'Put some comment'
    - **comment: Beperk de verticale snelheid**
- **'If a variable has a value'** *(=Als een variabele een waarde heeft)*
    - **variable: vspeed** (*= variabele: v.snelheid (verticale snelheid))*
    - **value: 12** *(= waarde 12)*
    - **operation: larger then** *(= operatie: groter dan)*
- **'Set the value of a variable'** (*= Stel de waarde in van een variabele)*
    - **variable: vspeed** (*= variabele: v.snelheid (verticale snelheid))*
    - **value: 12** *(= waarde12)*



Collision event met een bodem

In het event Step heb je geprogrammeerd dat het object net 1 boven de bodem blijft hangen. Immers zodra er binnen afstand 1 een bodem was, moest de zwaartekracht 0 worden. Dus staat het object 1 boven de bodem stil. Nu moet je nog zorgen dat hij wel netjes op de bodem landt en niet blijft zweven.
Dat programmeer je als volgt.

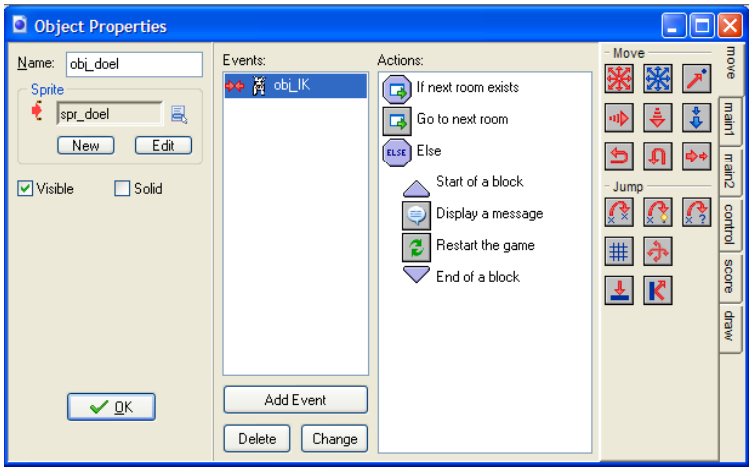
6. Voeg bij je besturingsobject het event **Collission** met **obj_bodem** toe.
7. Sleep de volgende acties naar het tabblad Actions. Ze staan allemaal op het tabblad move. En vul de bijbehorende waarden in.
   - **Move to contact position**
     - **direction: direction**
     - **maximum: 12**
     - **against: solid objects**
   - **Set the vertical speed**
     - **0**
     - **relative**: **niet** aangevinkt



Obj_doel programmeren

Het obj_doel werkt op dezelfde manier als obj_doel in een doolhofspel.

Als obj_ik botst met **obj_doel**, **dan** wordt eerst **gekeken of** er nog een **volgend level** is. Is dat het geval, **dan** ga je naar het **volgende level**. **Zoniet**, dan hebt je het gehaald en **dan start** bijvoorbeeld het spel **opnieuw**.



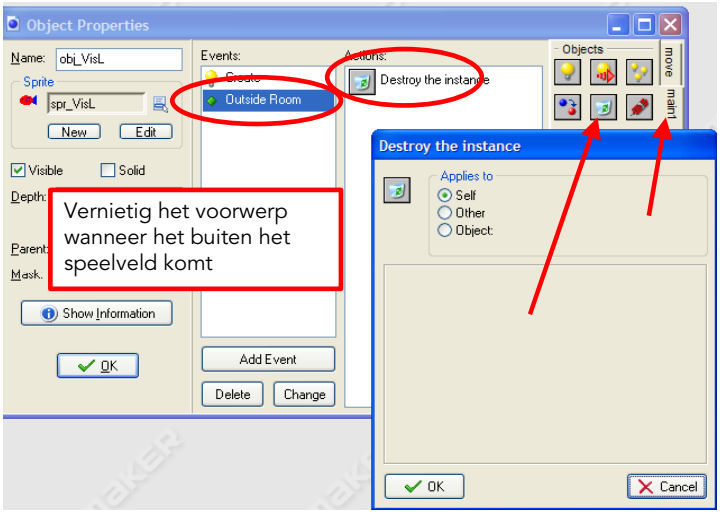
Hoofdstuk 5: Beloningen en hindernissen

Heb je een eerste level met bodems, de ik-figuur en een doel af en getest of alles goed werkt, dan kun je beloningen en hindernissen gaan programmeren. Vergeet dan ook niet om een controller-object te maken.
Kijk in deel 1 of 2 hoe je dat allemaal doet. Hieronder volgen nog een paar tips.

Het object bereikt de rand van het speelveld en dan verdwijnt het object

Wanneer objecten over het speelveld bewegen kunnen ze van het speelveld af raken. Het is belangrijk om die objecten wel te laten verdwijnen, anders blijven ze namelijk eigenlijk gewoon in het spel. Bij een schietspel heb je dan bijvoorbeeld misschien wel 100 kogels die je niet meer ziet. Maar het spel wordt daardoor wel steeds trager. In het spel Beertje in deel 1 en in Snuf in deel 2 kon een object niet van het speelveld af omdat er een muur omheen stond. Bij een platformspel is het vaak handiger het te programmeren.
Soms wil je, dat een object, dat van het speelveld afraakt gewoon verdwijnt.
Dat programmeer je als volgt:

1. Kies bij Events voor **Other** (=*overig*) en dan **Outside room** (=*buiten de room*).
2. Sleep van tab main1 de actie **'Destroy the instance'** (= *vernietig het voorwerp*) naar het vak Actions.



Het object verdwijnt van het speelveld en komt aan de andere kant weer op

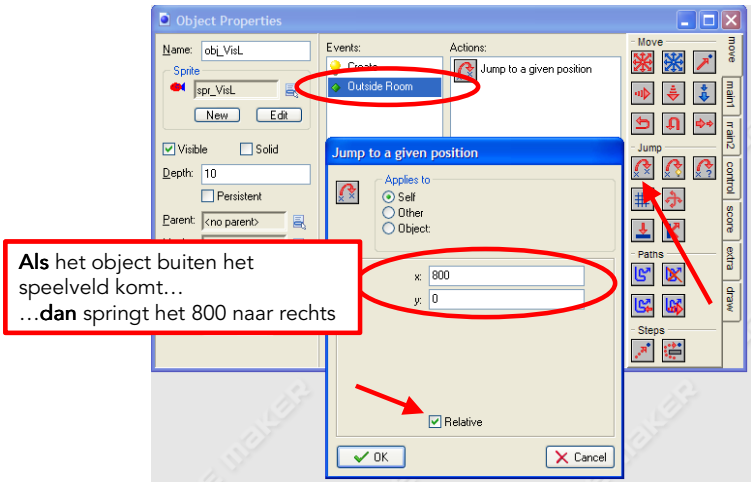
Wanneer je een platformspel met bewegende objecten maakt, dan kun je er ook voor kiezen, dat wanneer een object links over het randje gaat, het object rechts weer opkomt. Dit kunnen we programmeren met Other Event 'Outside room'. (= *Overige gebeurtenissen 'Buiten het speelveld'*.)

We gaan ervan uit dat je speelveld 800 bij 600 is en dat het object naar links beweegt.

1. Kies bij Events voor **Other** en dan **Outside room**.
2. Sleep de actie **Jump to a position** (=*spring naar een positie)* naar het vak Actions.
3. Vul bij **x 800** in en bij **y 0**.

Let op: Als de breedte anders is dan moet je bij x een ander getal invullen. En beweegt het object naar rechts, dan vul je bij x bijvoorbeeld **-**800 in.

4. Vink **relative aan**.
   Omdat relative is aangevinkt, wordt x 800 groter en y 0, dus het object springt 800 naar rechts.



5. Wat moet je voor x en y invullen wanneer je wilt dat een object dat aan de bovenkant het speelveld verlaat, weer aan de onderkant opkomt?

   ……..

Met een bom een muurtje opblazen

In cursusboek deel 2 heb je geleerd hoe je een muurtje op kunt blazen, zonder bom. De muur blies zichzelf op. Nu leer je hoe je een bom ergens kunt neerleggen en daarmee de muurtjes kunt opblazen.

De bom wordt op afstand tot ontsteking gebracht met een trigger, .

Je moet daarvoor het volgende programmeren:

Als de ik-figuur tegen de trigger botst, **dan** moeten alle bommen veranderen in explosies. De trigger moet **dan** verdwijnen. Die explosies vernietigen op hun beurt alles in hun omgeving.

1. Je hebt de volgende sprites nodig:
   a. **spr_trigger**
   b. **spr_bom**
   c. **spr_explosie**
   d. **spr_muur**
2. Maak met elke sprite een **object**

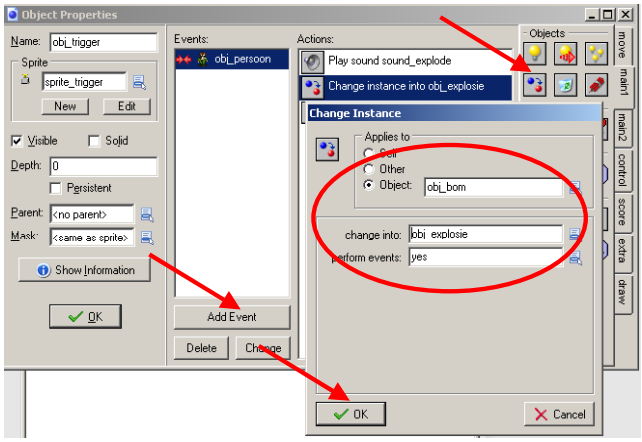
Dan voeg je aan de objecten de events toe:

3. Ga naar **obj_trigger**
   a. Kies **Add Event** -> **Collision** -> **obj_IK**

      Voeg bij deze event Collision de volgende acties toe:

      i. **Change instance** (= verander het voorwerp)
         1. Kies bij **'Applies to'** (= toepassen op) Object : **obj_bom**
         2. Kies bij **'Change into'**(= verander in) **obj_explosie**

3. Kies bij **'Perform events'** (=events ook uitvoeren) **yes**
4. Klik op **OK**, om deze actie op te slaan



ii. Voeg bij event Collision de actie **Destroy the instance** toe,
   1. Kies bij **Applies to: Self**
   2. Je kunt ook nog een explosie-geluid toevoegen met **Play a sound**. Je moet dan wel eerst een Sound hebben aangemaakt (Add a sound)
   3. Klik op **OK**, om deze actie op te slaan

4. Ga naar **obj_explosie**

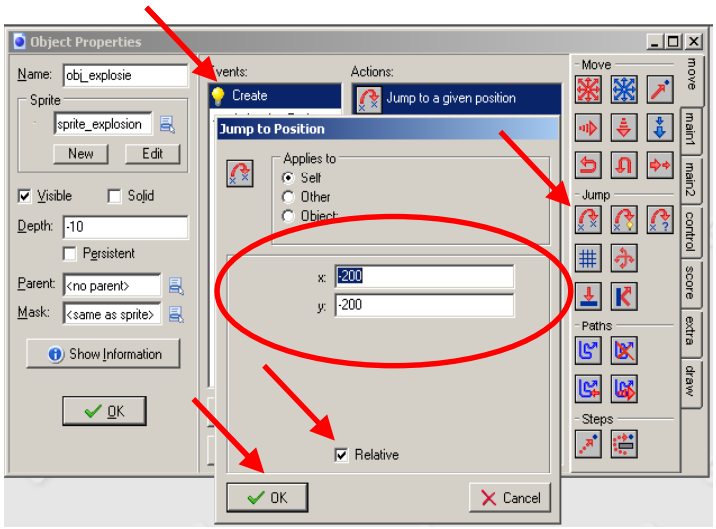
   Je moet voorkomen dat de explosie zichzelf opblaast. Daarom springt de explosie, zodra hij is gemaakt (namelijk toen de bom in een explosie veranderde) even opzij. Raar maar waar.

   a. Kies **Add event -> Create**

      Voeg bij dit event Create de volgende acties toe:

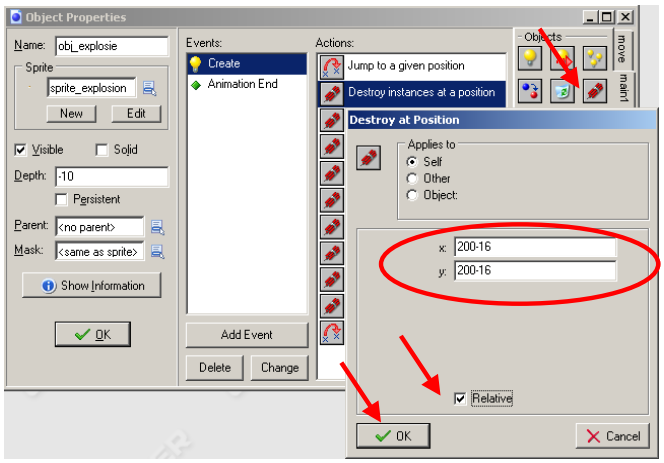
      i. **Jump to a given position** (=spring naar een positie)
         1. Kies bij **Applies to -> Self**
         2. Vul bij **x in: -200**, vul bij **y in: -200**
         3. Vink **Relative aan**.

4. Klik op **OK** om de actie op te slaan.



Nu moet alles in de buurt van de oorspronkelijk plek van de explosie alles exploderen. Alles in de buurt van de explosie wordt gerekend vanaf de linkerbovenhoek van het hokje waar de explosie in stond. We gaan ervan uit dat de hokjes in de room 32 bij 32 zijn.

5. Voeg bij **obj_explosie** bij event **Create**
    a. de actie **'Destroy at Position'** (=vernietig op een positie) toe
        i. Kies bij **Applies to -> self**
        ii. Vul bij **x in: 200-16** (16 links van de beginpositie van de explosie)
        iii. Vul bij **y in: 200-16** (16 boven de beginpositie van de explosie)
        iv. Vink **Relative** aan
        v. Klik op **OK** om de actie op te slaan



Dus een voorwerp in het hokje **linksboven (1)** van de explosie verdwijnt.

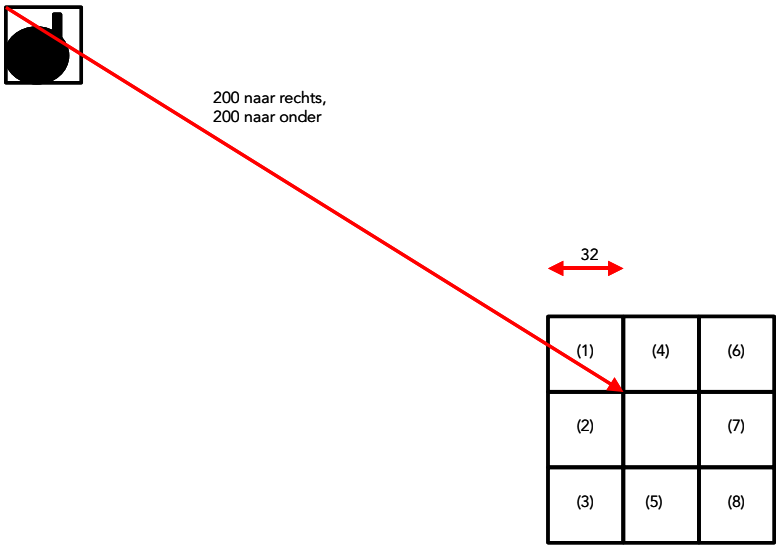
b. Voeg bij obj_explosie bij event Create nog een actie 'Destroy at Position' toe
   i. Vul weer bij **x: 200 - 16** in
   ii. Vul nu bij **y: 200 + 16** in
   iii. Vink **relatieve** weer **aan**
   iv. Klik op **OK** om de actie op te sluiten

Nu verdwijnt een voorwerp in het **hokje links (2)**.

En zo voeg je bij event Create ook acties toe die de voorwerpen in de andere hokjes laten verdwijnen.

Je hebt nog 6 andere acties 'Destroy at Position' nodig met de volgende x en y:

	x	y
3) hokje linksonder	**200 - 16**	**200 + 48**
4) hokje boven	**200 + 16**	**200 - 16**
5) hokje onder	**200 + 16**	**200 + 48**
6) hokje rechtsboven	**200 + 48**	**200 - 16**
7) hokje rechts	**200 + 48**	**200 + 16**
8) hokje rechtsonder	**200 + 16**	**200 + 48**



Nu moet obj_explosie weer terugspringen naar zijn plaats.

c. Voeg bij **obj_explosie** bij event **Create** actie

Jump to a given position (=spring naar een positie) toe.

i. Kies bij **Applies to -> Self**

ii. Vul bij **x** in: **200**, vul bij **y** in: **200**

iii. Vink **Relative aan**.

iv. Klik op **OK** om de actie op te slaan.

6. Als de explosie is afgelopen moet hij weer worden vernietigd.
   Voeg daarom bij obj_explosion event Animation End toe.

   a. Kies **Add event -> Animation End**

   b. Voeg bij Animation End de actie **'Destroy the instance'** toe

Tip 1:

Je wil niet een trigger gebruiken maar de hoofdpersoon bommen laten leggen

In sommige spellen gebruik je geen trigger, maar legt je hoofdpersoon bijvoorbeeld een bom neer met drukken op <spatie>.

Je hebt dan het object obj_trigger niet nodig.

7. Je kunt dit programmeren door bij je **ik-figuur** het event **'Key Press'** -> **<Space>** toe te voegen.

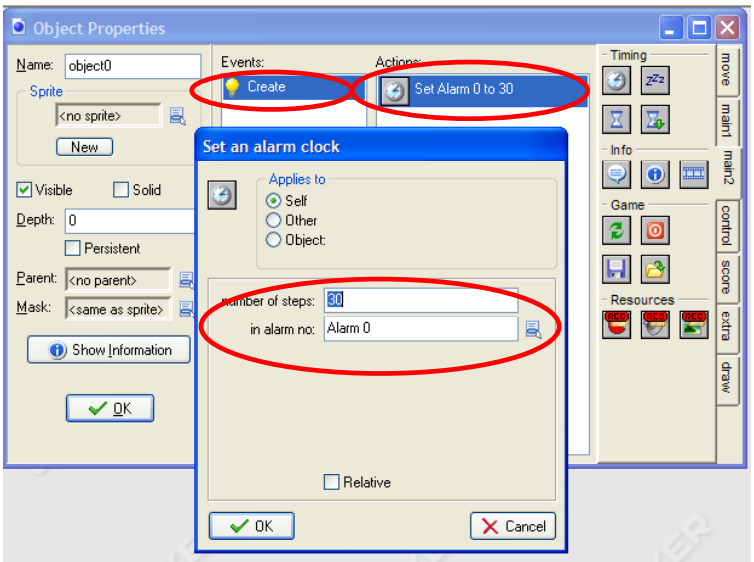
   a. Bij event Key Press->space voeg je dan actie **'Create an instance of an object'** toe.

      i. Kies bij **object obj_bom**

      ii. Kies bij **perform events yes**

Om te voorkomen dat de hoofdpersoon wordt opgeblazen, geef je hem/haar even de tijd om weg te komen.

8. Begin daarom bij het **obj_bom** bij event **Create** met de actie **Set an alarm clock** (= *zet een wekker*).
    a. Kies bij **number of steps** bijvoorbeeld **30** en kies bij **'in alarm no' Alarm 0**



9. Als de wekker na 30 stappen afloopt moet de bom ontploffen.
   Voeg daarom bij events event Alarm 0 toe. (Klik op **Add event** --> kies **Alarm** ---> kies **Alarm 0**.)
    a. Vervolgens voeg je de actie **'Change instance'** toe, waarbij je:
        i. bij **object** kiest: **obj_explosie**,
        ii. en bij **'Perform events'** kies je **'yes'**.
        iii. Voeg bij **obj_explosie alle events en acties zoals hierboven** toe.

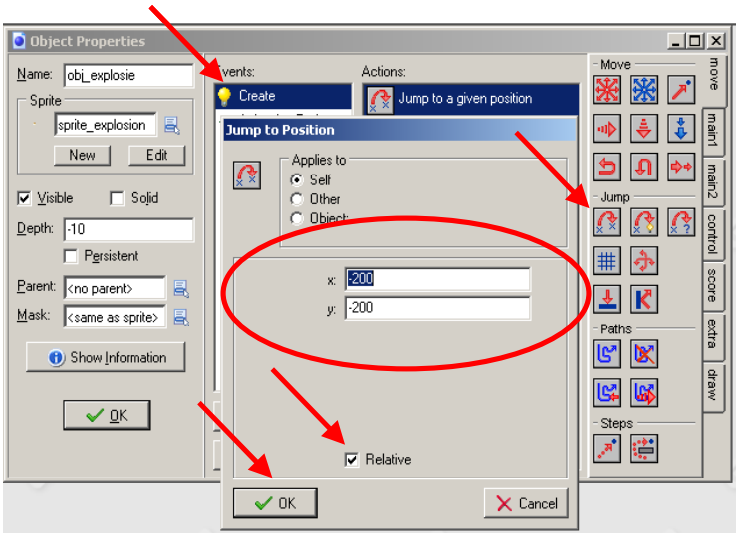
Tip 2: Niet alle objecten in de buurt moeten exploderen

Wil je niet dat alle objecten ontploffen in de buurt van de bom, dan moet je die ook tijdelijk even buiten het speelveld plaatsen. Dat gaat op dezelfde manier als je deed met de bom.

10. Kies bij object **obj_Explosie Add event** -> **Create**
    a. Voeg bij dit event **Create** de volgende acties toe:

i. **Jump to a given position** (=spring naar een positie)

1. Kies bij **Applies to** -> **Object** en kies het object dat je niet wilt laten ontploffen
2. Vul bij **x** in: **-200**, vul bij y in: **-300**
3. Vink **Relative aan**.
4. Klik op **OK** om de actie op te slaan.
5. Zorg dat deze **actie bovenaan** in de lijst komt.



Dan moet het object ook weer terugspringen naar zijn plaats.

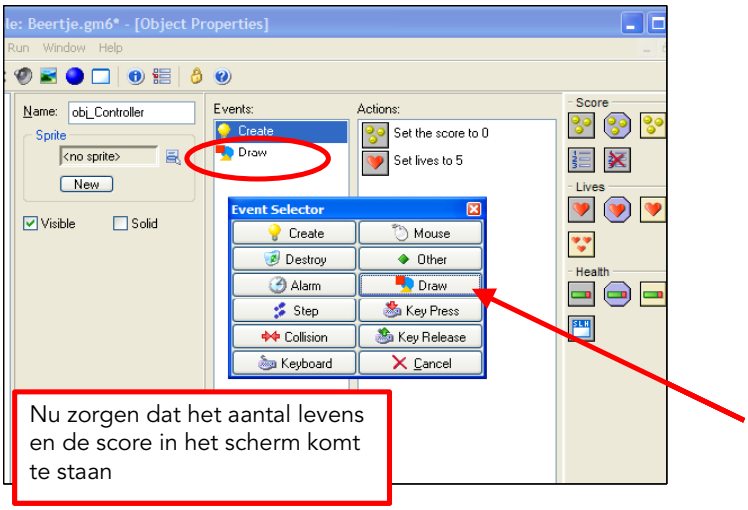
b. Voeg bij event **Create** actie **Jump to a given position** (=spring naar een positie) toe.

1. Kies bij **Applies to** -> **Other** en kies het object dat nu terug moet springen
2. Vul bij **x** in: 200, vul bij y in: **300**
3. Vink **Relative** aan.
4. Klik op **OK** om de actie op te slaan.

Levens en punten bewegen met het scherm mee

Als de achtergrond meebeweegt met de ik-figuur dan zie je op een gegeven moment de levens en de score niet meer. Die zijn in een hoekje blijven staan. Met de event Draw kun je ervoor zorgen dat ook de levens en score altijd zichtbaar blijven.

11. Klik bij **obj_Controller** op **Add Event** en daarna op **Draw**.

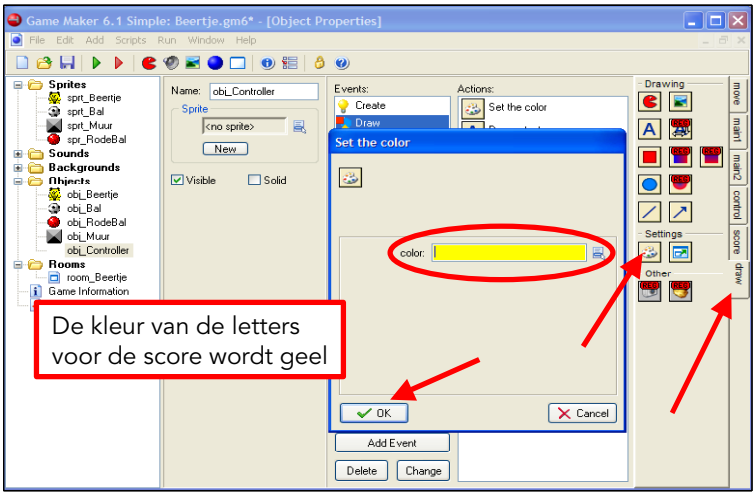


Eerst gaan we de kleur van de tekst bepalen.

12. Sleep vanuit het **tabblad Draw** de actie **Set the color** naar Actions.

    Kies een **kleur** die goed opvalt tegen de achtergrond van je spel .

    Klik op **OK**.

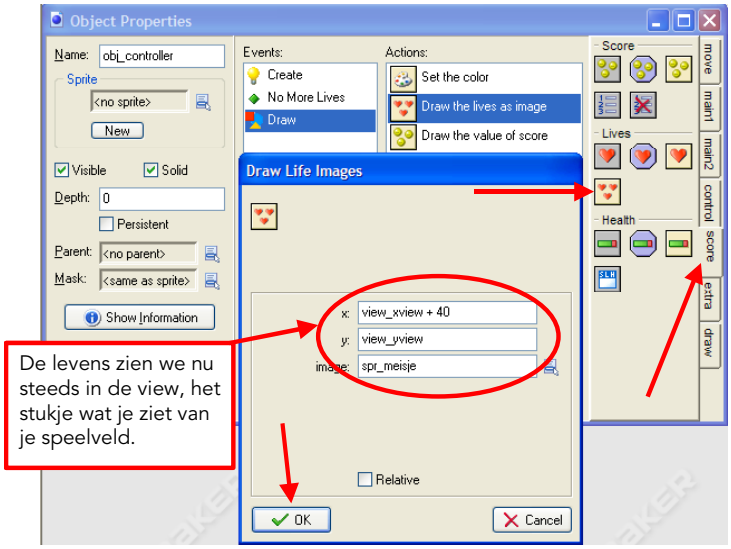


Vervolgens gaan we de levens als plaatjes weergeven.

13. Sleep vanuit het **tabblad score** de actie **'Draw the lives as image'**(= *teken het* aantal *levens als plaatjes*) naar Actions.

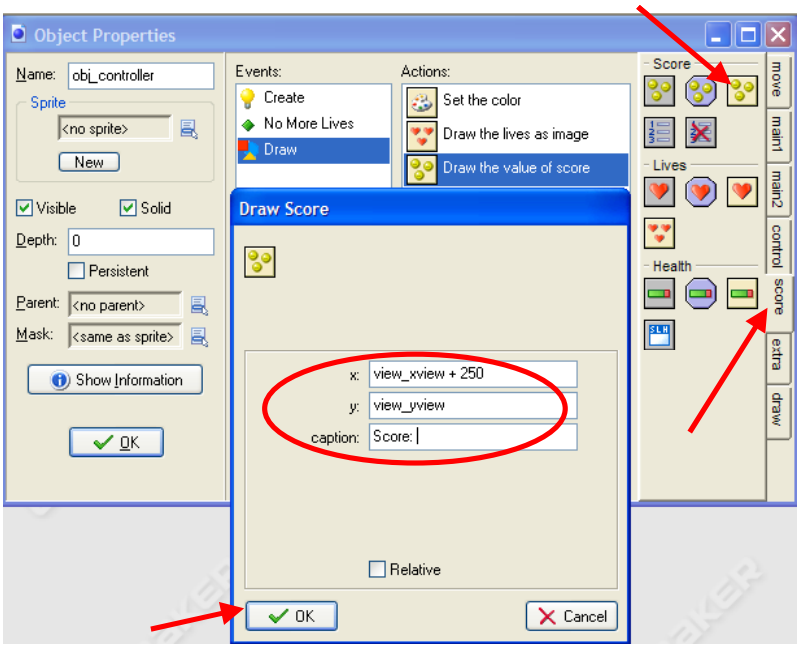
De levens en de score moet altijd in de view, het stukje wat je ziet van het speelveld, worden getoond. Van de linkerbovenhoek van de view is de x-waarde view_xview en de y-waarde view_yview. Als je dit niet snapt is niet erg, typ precies wat hieronder wordt aangegeven.

14. Vul bij **x view_xview+40** in en bij **y view_yview**.
    Het plaatje komt zo 40 pixels, is een klein stukje, van de linker rand en tegen de bovenrand van de view.
15. Kies bij **image** het **plaatje** dat je als leven wil gebruiken.
16. Klik op **OK**



En dan willen we ook nog de score in het scherm van het spel.

17. Sleep vanuit het **tabblad 'score'** de actie **'Draw score'** naar Actions.
18. Vul bij **x view_xview + 250** in en bij **y view_yview + 8**. (Hiermee bepaal je waar de tekst in de view komt)
19. Typ bij **caption Score:** Dit is de tekst die in het scherm komt te staan.
20. Klik op **OK.**



Hoofdstuk 6: Je spel afmaken

Zoals je hebt gemerkt hebben zijn alleen nieuwe onderwerpen behandeld. Aan jou om helemaal zelf te bedenken wat je allemaal nog meer wil inbouwen. Meer levels, geluiden, opdrachten, scores enzovoorts, enzovoorts. Laat je fantasie spreken en wie weet maak jij een spel waar straks miljoenen kinderen mee spelen.
Veel succes!

Hoofdstuk 7: Tenslotte

Dit was het einde van het derde deel van de cursus. Wil je nog verder, dan kun je op internet zoeken naar meer informatie hoe je iets aan moet pakken.
Op **www.gamemakerkids.nl** vind je links die je kunnen helpen bij je zoektocht.
We hopen dat je het een leuke cursus hebt gevonden en dat je er veel van hebt geleerd.
Heb je tips voor verbeteringen, laat het ons weten.

Overzicht van cursusboeken Game Maker

Game Maker-Introductie

Stap voor stap maak je een spel, waar al een begin mee is gemaakt, af.

Game Maker-Deel 1

Met dit cursusboek maak je helemaal zelf een compleet spel met veel mogelijkheden.

Game Maker-Deel 2

Je leert hoe je een spel ontwerpt. Waar moet je op letten. Wat maakt een spel tot een goed spel. Daarna ga je het zelf ontworpen spel maken.

Game Maker-Deel 3

In dit deel leer je nog meer trucs, zodat je ook andere spellen kunt maken, bijvoorbeeld een avonturenspel (platformgame).

Alle cursusboeken kunnen worden gedownload van **www.gamemakerkids.nl**.

Hoofdstuk 8: Bijlage - testschema voor de eindtest

Event	Actie	OK?

Event	Actie	OK?