




WORKSHOP 4

In deze workshop leer je hoe

-  we je hoofdrolspeler sterker kunnen maken zodat hij bijvoorbeeld rotsblokken kan voortduwen
-  hoe we je hoofdrolspeler bommen kunnen laten plaatsen
-  hoe we je hoofdrolspeler kunnen laten schieten



Net als in vorige workshops is het ook hier weer de bedoeling dat je de voorbeelden meebouwt.

In de voorbeeld levels wordt ook gebruik gemaakt van allerlei zaken die in vorige workshops aan de orde zijn gekomen. Die ga ik dan ook niet nogmaals bespreken. Bekijk hiervoor ook het voorbeeldspel dat bij deze workshops hoort.



Voorwerpen voortduwen

Het is mogelijk om je hoofdrolspelertje voorwerpen te laten voortduwen.

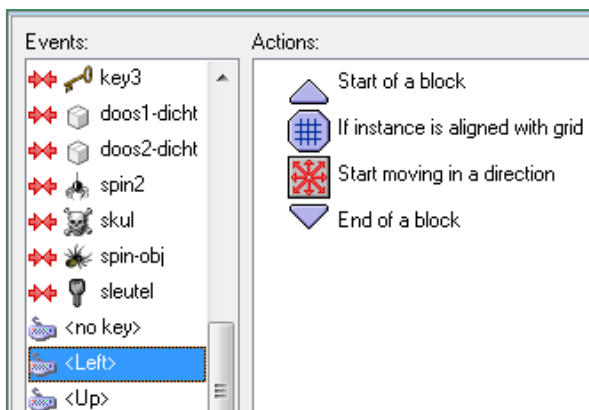
Denk daarbij aan het plaatsen van bommen, monsters insluiten met behulp van rotsblokken, water naar brandjes brengen, sleutels naar sloten brengen, enzovoorts. Laat je fantasie maar werken.



Bedenk zelf ook iets origineels en verzamel daarbij de juiste sprites en/of achtergronden.

Eerst Uitlijnen

Bij het voortduwen van bepaalde objecten is het zeer belangrijk dat de hoofdrolspeler, in dit geval Birdy, zich altijd precies binnen de rasterhokjes bevindt. Als eerste gaan we daar dus voor zorgen bij de eigenschappen van Birdy (of jouw poppetje).



Voeg bij **alle** richtingen de actie 'Check Grid' toe. Deze actie vind je onder tabblad 'control'.

Vergeet daarbij niet te melden dat het om een grid van 32 x 32 gaat.

Doe dit ook voor <no key>.

Test deze acties en merk op dat, als je het goed hebt gedaan, de hoofdrolspeler zich mooi en soepel over het veld beweegt.



Zie hier een deel van een level. Als achtergrond is een afbeelding van een spinnenweb gebruikt. In het midden een grote spin die er met behulp van tiles is ingezet. Allemaal zaken die in vorige workshops aan de orde zijn geweest.

Zie ook het voorbeeldspel.

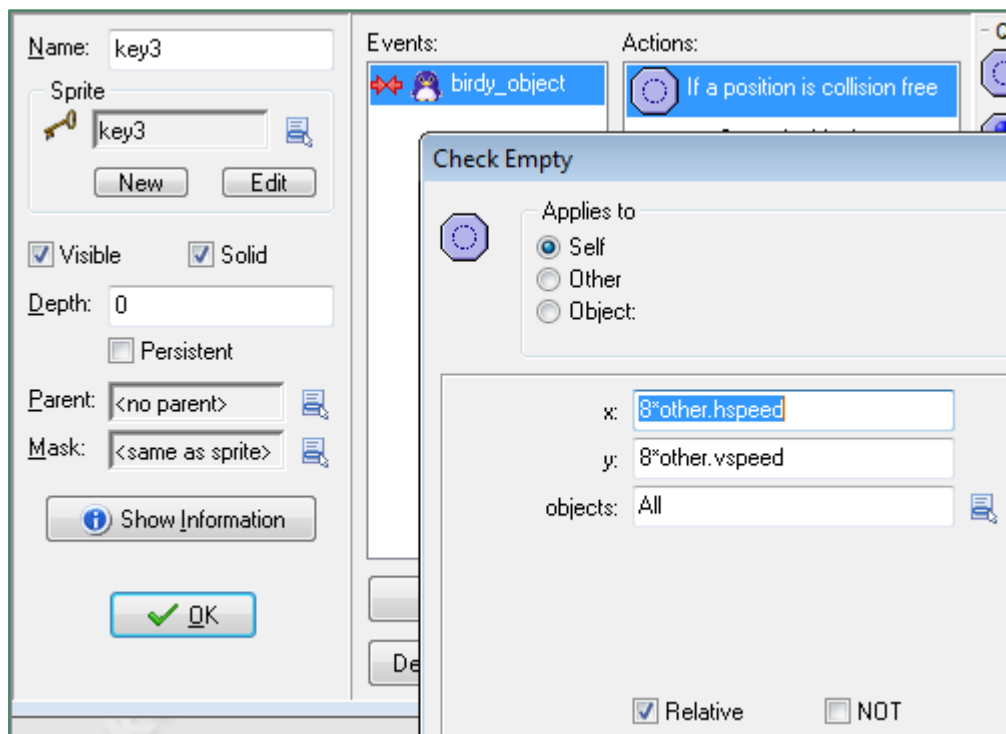


Waar het hier om gaat is dat achter ieder hekje een vogeltje zit opgesloten. Als Birdy nu een sleutel pakt en deze naar een hekje brengt kan hij een vogeltje bevrijden.



Om te beginnen vertellen we tegen Birdy dat als hij in aanraking komt met het sleutel-object hij niets mag doen.

Het is namelijk de sleutel die om zich heen moet kijken om de boel in de gaten moet houden. Open het eigenschapsvenster van het duw-object.



Als het duw-object in aanraking komt met Birdy moet er gekeken worden of de gewenste positie botsingvrij is. En dat is hier een beetje tricky.

Wij denken bij het Engelse woord Speed aan snelheid. Maar in Gamemaker heeft dat woord meerdere betekenissen. Snelheid, richting en afstand komen hier in één woord samen.











hspeed horizontale bewegingsrichting, **vspeed** verticale bewegingsrichting.

Birdy heeft een snelheid van 4, en het raster heeft een grid van 32x32.

Het duw-object moet dus kijken in de richting waarin Birdy ook kijkt. Omdat het duw-object een hokje van 32 pixels vooruit wil, moet hij ook zover vooruit kijken.

En dat is dan 8 maal de bewegingsrichting van Birdy.








Vandaar dus $8 * \text{other.hspeed}$.

 If a position is collision free
 Start of a block
 Play sound key
 Jump to position (8*other.hspeed,8*other.vspeed)
 End of a block
 Else
 Start of a block
 If there is an object at a position
 Jump to position (8*other.hspeed,8*other.vspeed)
 End of a block

Als de positie, in de kijkrichting van Birdy, 32 pixels vrij is dan mag het duw-object naar deze positie springen. En anders blijft het automatisch staan.

Anders:

Als op die positie zich een hek-object bevindt mag het duw-object ook vooruit.

 birdy_object  hek-obj	 Start of a block  Destroy the instance  Destroy the instance  Play sound bird  End of a block
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

En als het duw-object, in dit geval dus de sleutel, tegen het hek aankomt moeten beide verwijderd worden. Het vogeltje is blij dat het vrij is en laat dat even horen.



En natuurlijk moet er ook iets gebeuren als alle hekjes zijn verdwenen. Bijvoorbeeld een deur naar het volgende level. Dus moet hier een controller voor komen.



Maak een level waarin je je poppetje een voorwerp laat duwen. Verder even niets. Test of het allemaal soepel en naar wens gaat. Breidt dan je level verder uit.

Bommen en explosies



Met één of meerdere bommen kun je nog meer spektakel in je spel brengen. Je kunt bijvoorbeeld muurtjes opblazen of monsters doodmaken.

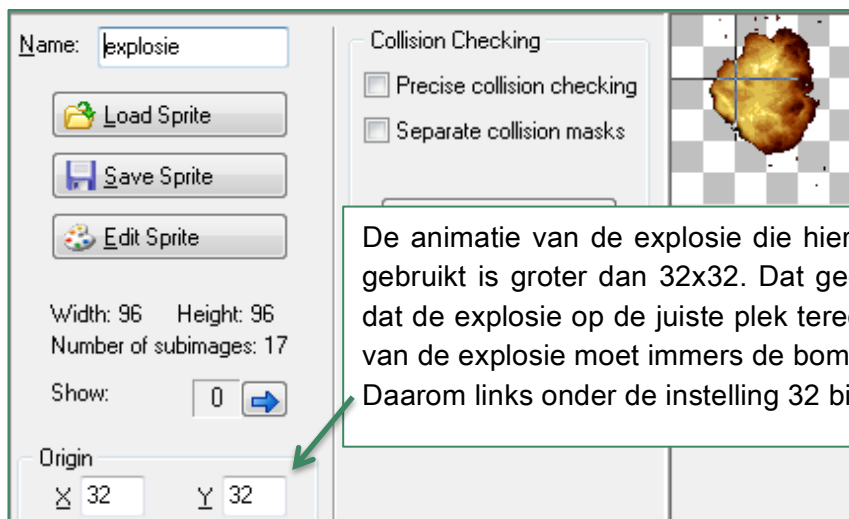
Voor het realiseren van zo'n bomexplosie zijn minsten vier dingen nodig:

1. een plaatje van een bom
2. een plaatje van een trigger
3. een plaatje (animatie) van de explosie
4. geluid van een explosie
5. eventueel een plaatje van een ingedrukte trigger



Verzamel een aantal van deze sprites en koppel ze aan objecten.

Maak een level waarin je een tegelmuurtje maakt en een bom en een trigger neerzet. Dan kun je zo dadelijk testen of het allemaal naar wens gaat.



NB

Je kunt één of meerdere bommen in het speelveld neerleggen. Bedenk echter wel dat als je poppetje in aanraking met de trigger komt alle bommen tegelijk afgaan. Als je dat niet wilt dan moet je meerdere triggers en meerdere bommen nemen.

Bijvoorbeeld trigger1 en bom1, trigger2 en bom2, enzovoorts. Daar kun je uiteraard wel steeds dezelfde sprite voor gebruiken.



Links zie je een deel van een voorbeeldlevel. Zie ook het voorbeeldspel.

Daar zweven hele enge grote wespen van links naar rechts en kruipen er torren van boven naar beneden. Daar kom je niet zomaar doorheen. Derhalve zijn daar een aantal bommen geplaatst, grijze en zwarte. Verder ook twee triggers. Eén voor de grijze bommen en één voor de zwarte bommen.

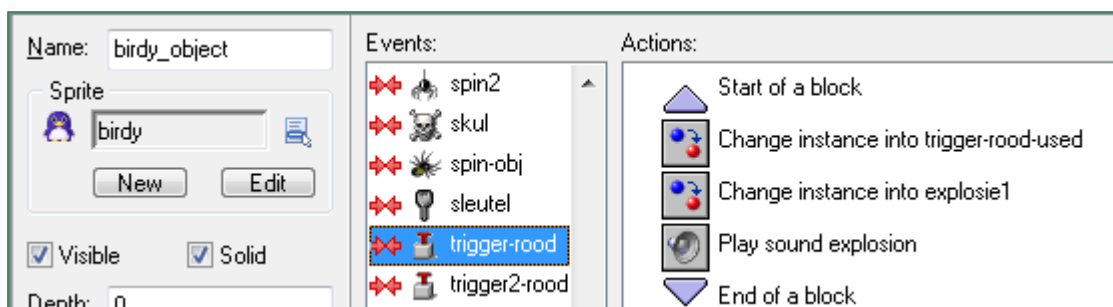
Als Birdy met de eerste trigger in aanraking komt gaan de grijze bommen af. Dat wil zeggen, als Birdy aan deze trigger komt :

- worden de instanties van het grijze bom-object verwisseld voor een explosie-animatie
- wordt een explosie geluid afgespeeld
- worden instanties van bepaalde objecten vernietigd.
- Wordt de trigger verwisseld met een afbeelding van ingedrukte trigger. (niet noodzakelijk, alleen maar leuk)



Om dit allemaal voor elkaar te krijgen hebben we behoorlijk wat instellingen nodig en een stukje code.

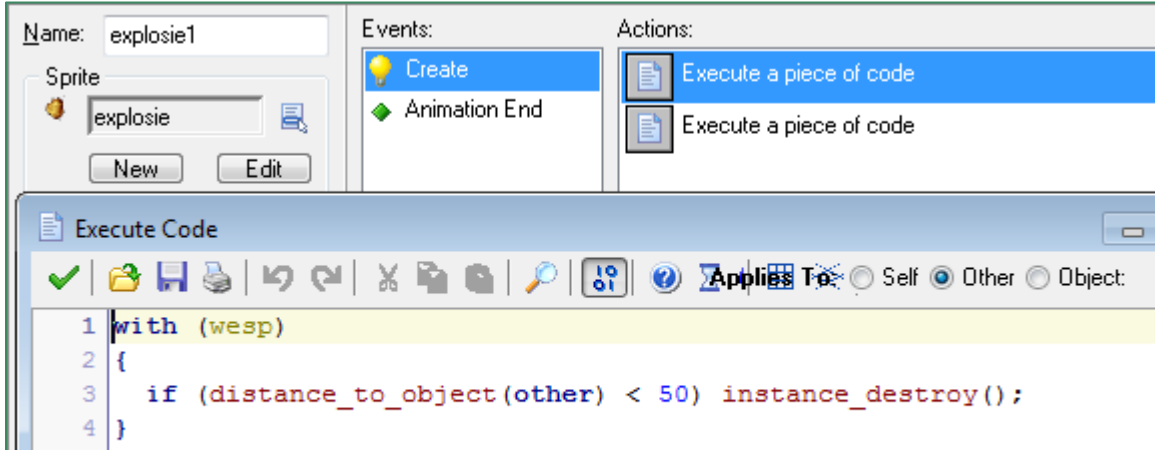
Om te beginnen gaan we bij de eigenschappen van Birdy instellen wat er moet gebeuren als Birdy in aanraking komt met de trigger.



Dan moet de trigger in een ingedrukte trigger veranderen, de bom moet veranderen in explosie1 en moet er ook een explosie te horen zijn. Tevens moet bij **'perform events'** *yes* worden ingevuld. Er moet namelijk nog iets gebeuren, er moeten object-instanties worden vernietigd.

Er is ook een trigger2 en een explosie2, die horen bij de zwarte bommen. Maar krijgen natuurlijk dezelfde instellingen.

Vervolgens gaan we bij `explosie1` instellen wat er verder gebeuren moet. Als de explosie wordt afgespeeld moeten er zoals boven is gezegd dingen vernietigd worden. En aan het eind van de animatie moet deze zichzelf vernietigen. (prullenbakje)



The screenshot shows the GameMaker 8.0 interface. The 'Name' field is set to 'explosie1'. The 'Sprite' is 'explosie'. The 'Events' list includes 'Create' and 'Animation End'. The 'Actions' list includes two 'Execute a piece of code' actions. The 'Execute Code' window is open, showing the following code:

```
1 with (wesp)
2 {
3   if (distance_to_object(other) < 50) instance_destroy();
4 }
```

`with (all)` In de code kunnen we een keuze maken voor 'with all'. Dat wil zeggen dat alles dan wordt vernietigd in een straal van 50 pixels ten opzichte van de animatie.

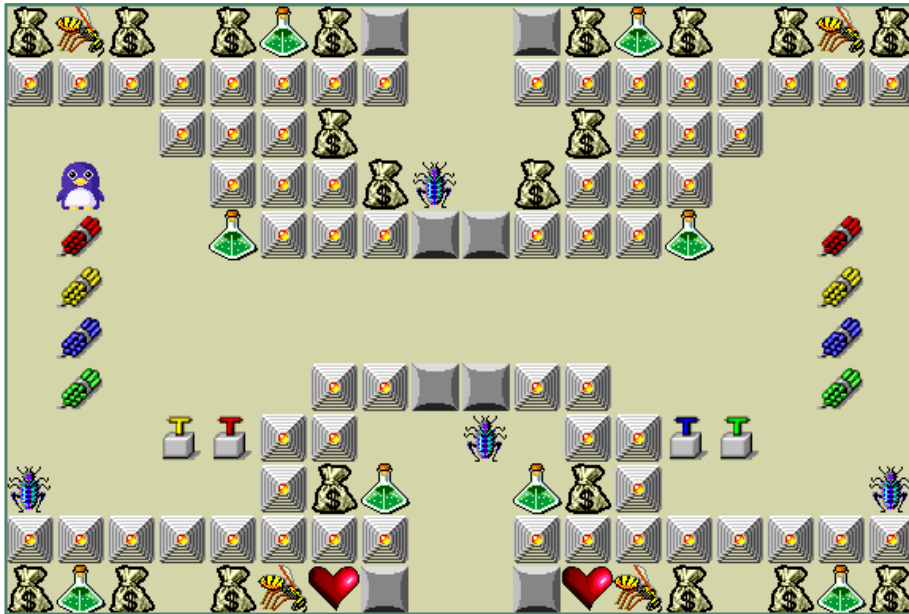
Ik wil echter dat de tegels blijven en Birdy zelf ook. Dus geef ik aan dat de wespen in nabijheid van de explosie moeten verdwijnen. Het zelfde stukje code doe ik dan ook voor de tor.

Hoe groot de straal moet zijn waarbinnen dingen worden vernietigd is geheel afhankelijk van de situatie. Daar moet je gewoon een beetje mee experimenteren.



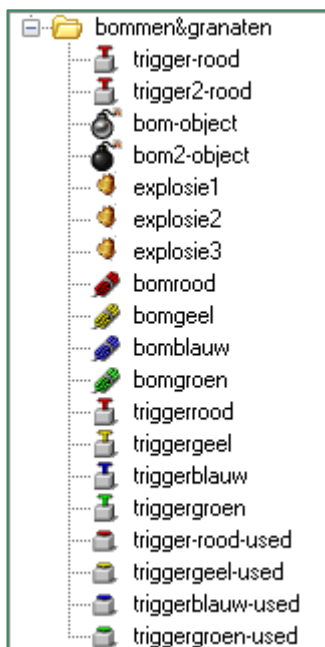
De speler laten bepalen waar de bommen gezet moeten worden.

Dat wil zeggen dat de bommen verschoven moeten worden. Zie voorbeeld onder en het voorbeeldspel.



Een voorbeeld van een level waarin de speler van het spel bepaald waar de bommen worden neergelegd.

Je ziet hier acht bommen in vier verschillende kleuren. Bij iedere kleur bom hoort een passende trigger. Dat kun je makkelijk in de editor zelf doen.

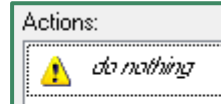


Zoals je links ziet zijn er voor iedere kleur passende objecten gemaakt. Ook is er een object explosie3 gemaakt omdat ik nu wel wil dat de tegels verdwijnen maar niet de zakjes met geld. Dus moet deze explosie net iets anders doen dan nr1 en nr2.

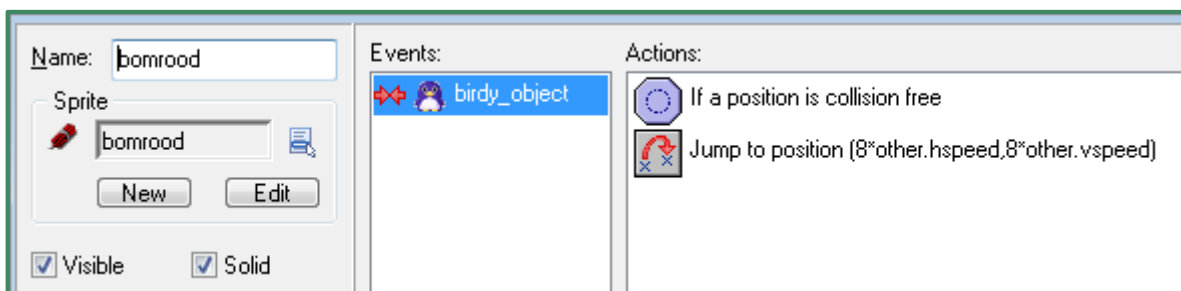


In dit voorbeeld worden de gekleurde bommen voortgeduwd. Dat gaat nagenoeg op dezelfde manier als het voortduwen van de sleutel of het rotsblok.

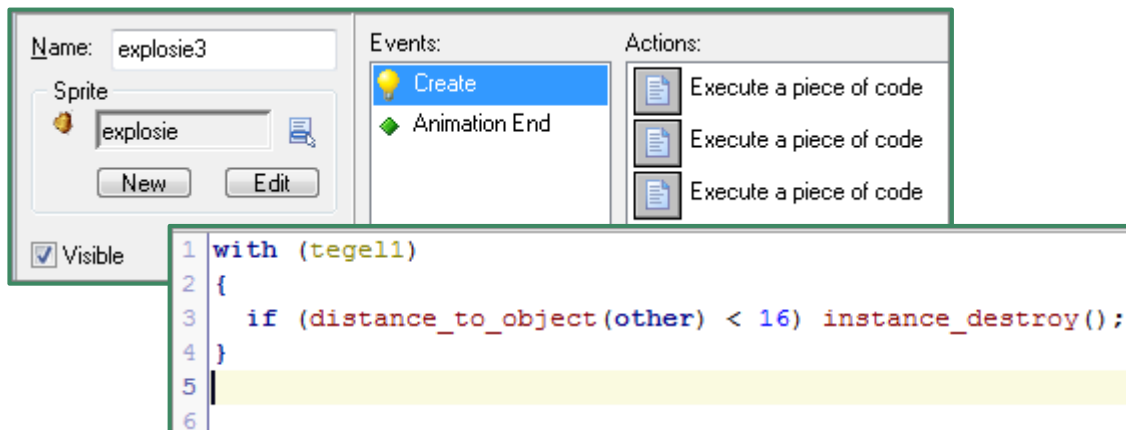
We vertellen dus eerst tegen Birdy dat hij niets moet doen als hij in aanraking komt met zo'n bom. Het is immers de gekleurde bom die om zich heen moet kijken waar hij naar toe wordt geduwd.



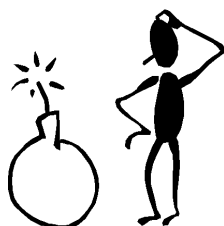
We vertellen dus tegen de bom dat als Birdy tegen hem aanbotst hij moet kijken of het plekje waar hij naar toe wordt geduwd wel vrij is. Als dat het geval is dan mag hij daar dus naar toe springen. Let op dat er nu *geen* 'else' komt. Er hoeft namelijk verder niets te gebeuren dan alleen maar de bom naar een bepaald plekje te duwen.



Tot slot vertellen we aan de explosie wat we van hem verwachten.



Omdat ik wil dat na de explosie de torren, de wespen en de tegels in de omgeving van de bom verdwijnen heb ik nu drie stukjes code. De afstand vrij klein omdat de explosie ook al 96x96 is. Zoals al eerder vermeld moet je daar een beetje mee experimenteren.



Schieten



Natuurlijk kan er in een beetje spel geschoten worden. Een spel waarin monsters en spoken rondwalen zonder dat er geschoten wordt is ondenkbaar.

Je kunt zelfs de munitie kiezen waarmee je hoofdpersoonje wil laten schieten. Dat kunnen kogels zijn, maar ook bijvoorbeeld eieren, tomaten of watjes.

Willen we de hoofdpersoon in alle vier de richtingen laten schieten moeten we wel zo'n persoonje hebben dat zich ook visueel in vier richtingen beweegt. Anders schiet hij uit z'n oren of uit z'n kontje.



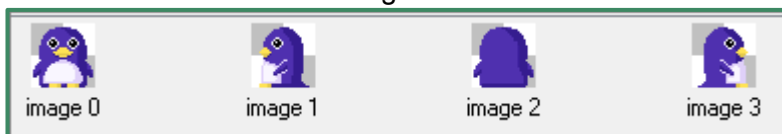
Waarschijnlijk heb je zo'n poppetje, maar als je dat niet hebt creëer er dan een. In workshop2 heb je geleerd hoe dat moet. Er waren twee mogelijkheden:

1. een sprite-animatie bestaande uit vier plaatjes die met behulp van een stukje code werden gewisseld van richting.
2. Vier aparte sprites die al naar gelang de gewenste richting gewisseld werden.

Om het poppetje te laten schieten moet je rekening houden welke mogelijkheid je hebt gekozen om je poppetje van richting te laten wisselen. Er zit namelijk een klein verschil wat betreft de schietinstellingen.

We beginnen met **mogelijkheid 1**

Om de richting te bepalen waarin geschoten wordt moet je eerst even kijken bij welk plaatje van de animatie welke richting hoort.



Controleer dit ook bij jouw poppetje en noteer dat. Je hebt deze gegevens nodig.

Rechts : de richtingen van het speelveld.

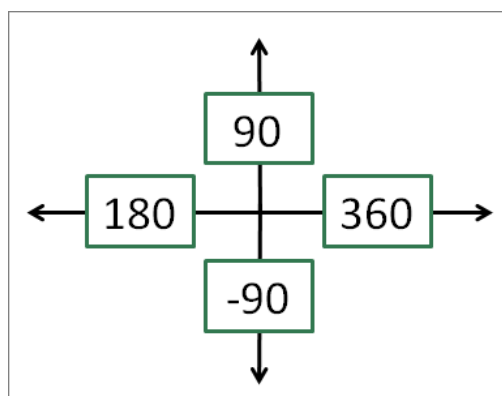
Maak een keypress-event en koppel deze aan een toets waarmee je kunt schieten. Bijvoorbeeld de spatiebalk. Voor elke richting moet je twee acties maken:



Nummer van het single-plaatje



Richting van de kogel en snelheid.



The screenshot displays the Game Maker 8.0 interface. On the left, a list of actions is shown under the heading 'Actions:'. Each action is preceded by a 'VAR' icon and a condition: 'If image_single is equal to 0', '1', '2', and '3'. Each condition is followed by a block of actions: 'Start of a block', 'Create moving instance of kogel', 'Play sound shoot', and 'End of a block'. On the right, two configuration panels are visible. The top panel, titled 'VAR', shows the 'Applies to' section with 'Self' selected, and the 'variable' field set to 'image_single', 'value' set to '3', and 'operation' set to 'equal to'. The bottom panel, titled 'Create Moving', shows the 'Applies to' section with 'Self' selected, and the 'object' field set to 'kogel', 'x' set to '0', 'y' set to '0', 'speed' set to '15', and 'direction' set to '360'. A 'Relative' checkbox is checked at the bottom of this panel.

Boven links:

De acties die horen bij het keypress-event.

Boven rechts:

- als er sprake is van image_single met waarde 3 (in het voorbeeld boven rechts)
- dan moet er een instantie van het kogel-object gecreeerd worden. Vanaf de plaats van het persoonje dat vuurt is dat relatief $x=0$ en $y=0$
- snelheid is 15
- richting is in dit geval 360

Dan nu **mogelijkheid 2**

Dat wil zeggen, je hebt voor je hoofdpersootje vier aparte sprites, namelijk: sprite-left, sprite-right, sprite-up en sprite-down

of iets van gelijksoortige benamingen

Dan moet de action 'Test Variable' net iets anders ingesteld worden.

Je maakt dan als variabele dan een sprite_index en de waarde wordt dan bijvoorbeeld sprite-left

De instellingen voor 'Create Moving' blijven gelijk als boven.

