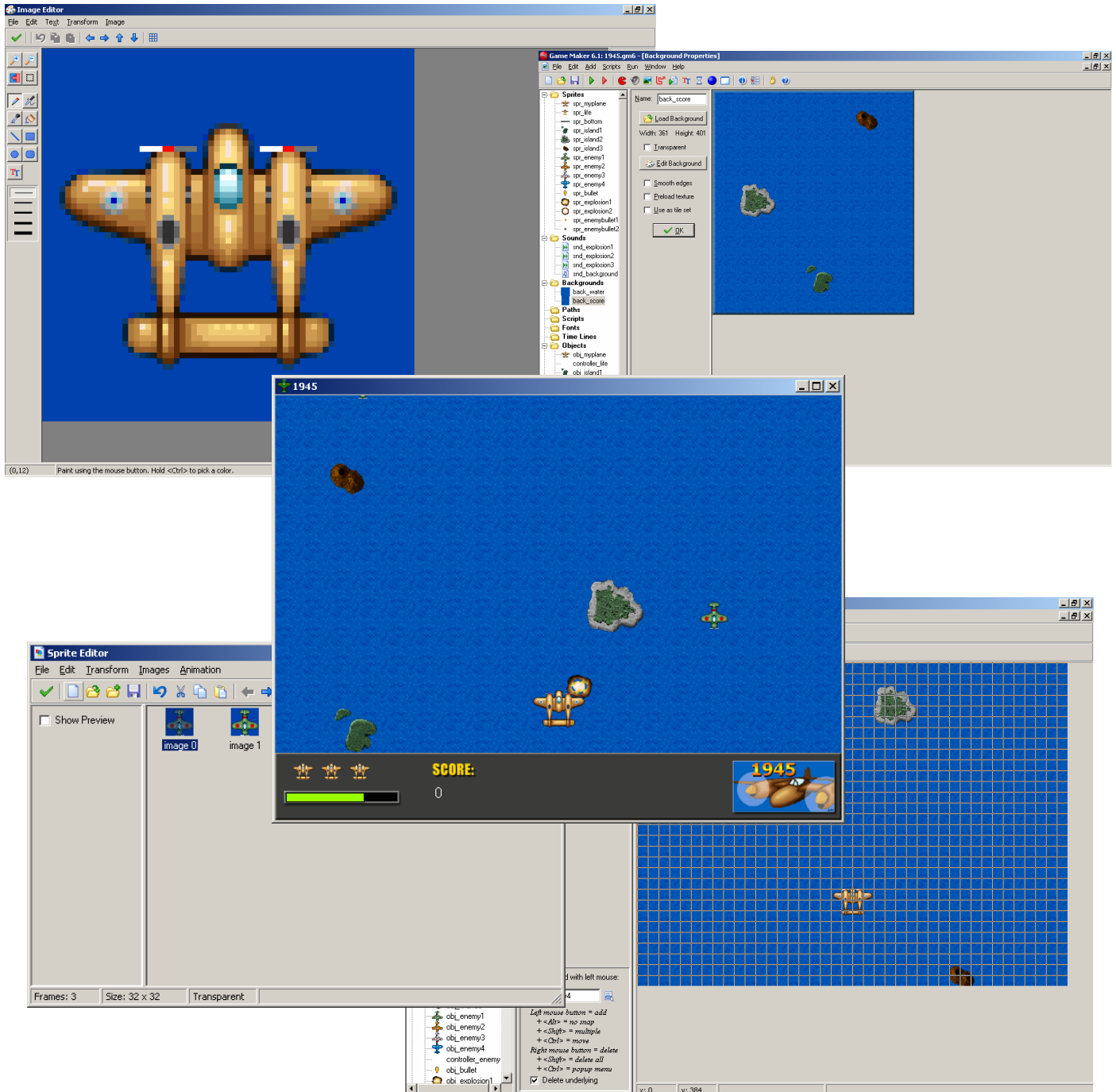


# Game Maker 6.1

## Scrolling shooter

### Hoofdstuk 6.1



## Scrolling Shooter

Een spelgenre is shooter. Je kent ze wel: Halo, de bondgames, Metroid Prime ook. Dit zijn First of Third Person Shooters. Daarnaast heb je ook nog de Scrolling Shooters. Een voorbeeld hiervan is 1945, geleverd bij Game Maker.

Hoe maak je nou zo'n Scrolling Shooter? In deze opdracht leer je zelf een scrolling shooter te maken. Je leert een vliegtuigje programmeren dat kan schieten en bewegen. Je leert hoe je vijanden willekeurig (random) op het veld kan laten verschijnen en nog meer.

Ik ga ervan uit dat je sprites, geluiden, objecten, achtergronden en rooms kan toevoegen. Zo niet, maak dan een van de andere opdrachten.

### Design Document:

#### **Description:**

Je gaat een Scrolling Shooter maken waarin je een vliegtuigje bestuurt, dat kan schieten en heen en weer bewegen. Als vijanden komen er andere vliegtuigjes op je af en aan het einde van het level kom je een eindbaas tegen.

#### **Game Objects**

Er zullen 3 objecten in het begin zijn. Deze objecten zijn: je eigen vliegtuig, tegenstander, je kogels. Je eigen vliegtuig heeft de afbeelding van een vliegtuig uit het extra resource pakket. (te vinden via <http://www.gamemaker.nl/resource.html>, weet alleen niet meer welke precies). Dit vliegtuigje bestuurt jij. Hij kan heen en weer bewegen en schieten. Bij een botsing met de vijanden gaat er een deel van zijn leven af.

Je tegenstander heeft ook de afbeelding van een vliegtuigje. Deze tegenstander komt op een willekeurige plek in het scherm en zal dan naar de andere kant vliegen. Als hij botst met een kogel van jouw gaat hij kapot.

Je kogels hebben de afbeelding van een balletje. Deze kogels worden gemaakt als je op de schietknop drukt en gaan kapot als ze in aanraking komen met de vijand.

#### **Sounds**

Ik voeg geen geluiden toe, maar dat zal je ondertussen zelf wel kunnen.

#### **Controls**

Bewegen doe je met de pijltjes toetsen, schieten met de spatiebalk.

#### **Gameflow**

Als het spel op begint staat de score op 0. De speler begint met 3 levens. Het spel eindigt als deze op zijn of als de speler op <Esc> drukt.

#### **Levels**

Er is 1 level

## Hoofdstuk 6.1

Ok, je kunt beginnen. Zet Game Maker 6.0 eerst in *advance mode*.

- Start Gamemaker op.
- Kies File.
- Kies *advance mode*.

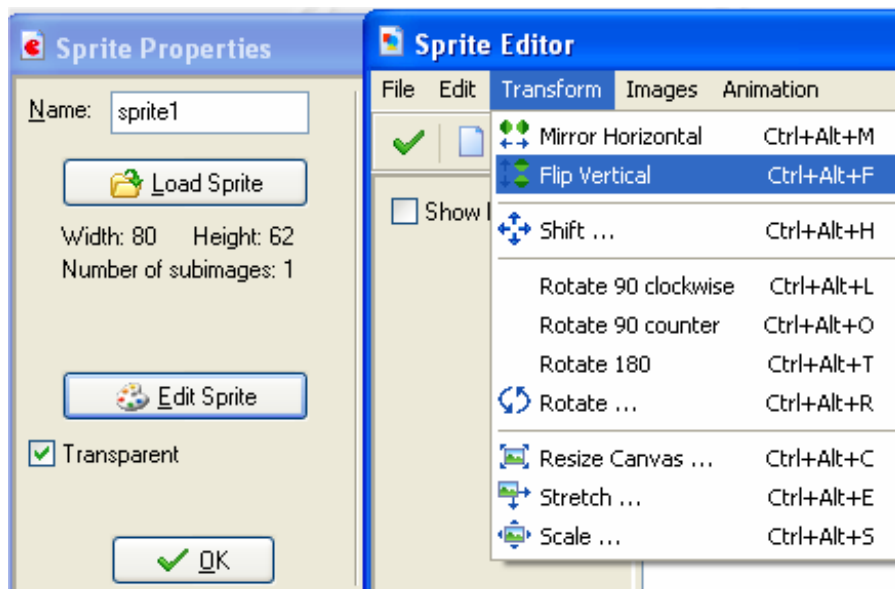
Je zult nu merken dat je in de menu's nu meer opties heb dan in de *simple mode*.

- Voeg de sprites, objecten en eventueel geluiden toe voor de speler en de vijanden.

Vergeet niet om duidelijke namen te gebruiken en voor de sprites de afkorting *sprt\_* te gebruiken en voor de objecten *obj\_* (dus *sprt\_speler* en *obj\_speler*, *sprt\_kogel\_speler* en *obj\_kogel\_speler*, *sprt\_vijand* en *obj\_vijand*)

De sprite van de speler moet omhoog gericht zijn en die van de vijanden omlaag.

- Als dit niet het geval is, klik dan op *edit sprite*
- Klik op *Transform*
- Kies *flip vertical*




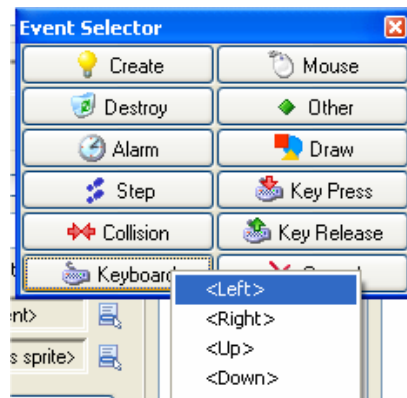
## Hoofdstuk 6.1

Je begint met het programmeren van de speler.

- Open het object *obj\_speler*.

Allereerst wil je dat hij kan bewegen.

- Add event: keyboard <left> toe.
- Kies nu de action  *jump to a given position* en vul bij  $x = -4$  in. Vergeet niet **relative** aan te vinken.
- Add event keyboard <right>.
- Kies nu de action  *jump to a given position* en vul bij  $x = 4$  in.

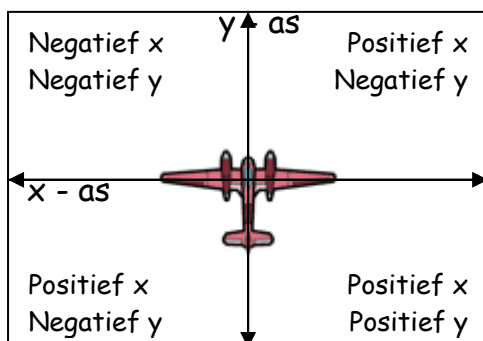


Als je wilt dat je vliegtuigje ook nog naar boven en beneden kan, moet je nog twee events toevoegen.



- Add event keyboard <up> en dan weer  gebruiken, maar dan bij  $y = -4$  invullen.
- Voor omlaag moet je keyboard <down> gebruiken en dan voor   $y = 4$  gebruiken.

Nu kan je vliegtuigje bewegen.

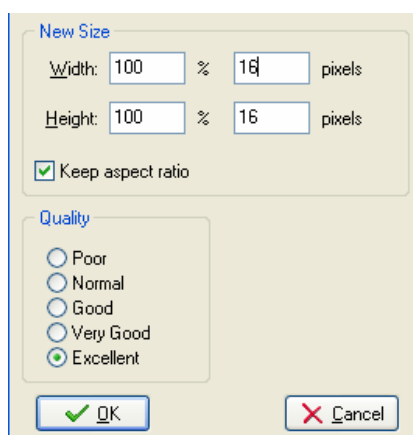
Bij deze methode verplaats je het vliegtuigje 4 pixels als je een toets indrukt. In het onderstaande schema kun je zien dat het midden van het vliegtuig het 0-punt is. Van uit het midden berekend Game Maker welke kant je op wil. Als de waarde van  $x$  negatief is en van  $y$  positief, dan beweegt je object naar linksonder. Zijn zowel  $x$  als  $y$  positief, dan beweegt het object naar rechtsonder.



## Hoofdstuk 6.1

Misschien ben je gewend een 2<sup>e</sup> actie te gebruiken om te bewegen? Je zal dan  gebruiken. Het verschil tussen beide is dat je Game Maker met de 1<sup>e</sup> manier het object een aantal pixels verplaatst en met de 2<sup>e</sup> manier je het object een snelheid naar een richting geeft en vervolgens een extra event toevoegt (key-release) zodat het object stopt met bewegen. Ik geef de voorkeur aan de  methode, maar je mag veranderen als je wilt.

Als je nu een room (kamer/level) toevoegt en daar je speler inzet kan je je spel uitproberen. Je zult nu zien dat het werkt, maar dat je speler het level uit kan. Daarvoor maak je een onzichtbare muur.

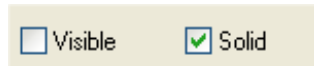



➤ Voeg een sprite toe voor een muur.  
*Sprt\_muur*

De kans is groot dat je vrij grote sprites hebt van de muur. Mooier is om deze te verkleinen zodat je niet al te snel op de muur zit. Dit doe je weer bij *edit sprite*.

➤ Nu ga je weer naar transform, maar daarna naar stretch.

- Kies voor een grootte van 16 bij 16 (niet de procenten gaan veranderen. Dit gebeurt automatisch)
- Vink ook nog even Excellent aan voor het beste resultaat.
- Voeg vervolgens een object toe en geef hem een naam en de sprite die je net hebt toegevoegd.
- Vink nu visible uit en solid aan.
- Voeg deze in het level toe (helemaal rondom de rand).



Nu moet je nog tegen het vliegtuigje zeggen dat als hij tegen de muur botst, hij niet verder mag. Dit doe je in de *collision* event van het vliegtuigje en de muur. Zet bij deze *event* de *action*  en dan de het blokje in het middel aanvinken.

Nu dit is gedaan ga je het vliegtuigje laten schieten. Heb je het object voor de kogel al toegevoegd? Zo niet, doe dat dan nu.

Je gaat eerst zeggen wat de kogel moet gaan doen als hij gemaakt wordt. Wat wil je? Je wilt dat hij naar boven gaat bewegen. Dit doe je met 

## Hoofdstuk 6.1

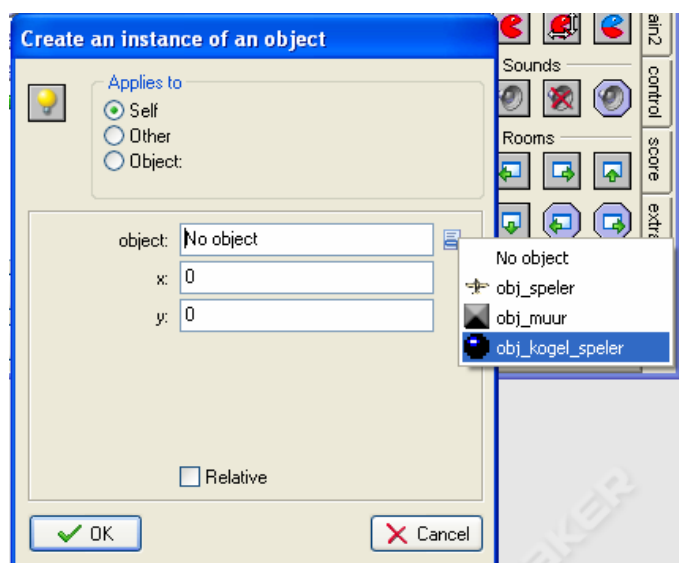
en dan pijltje naar boven aanvinken. Dit zet je in de *create event*. Gebruik 6 voor de snelheid.

Dus:

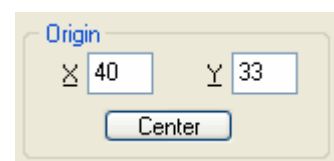
- ☛ Klik op *Add event*.
- ☛ Kies  *create*.
- ☛ Kies action  *move*.
- ☛ Kies *speed* = 6.

Nu moet je programmeren dat als je op de spatiebalk drukt, je vliegtuigje een kogel afschiet.

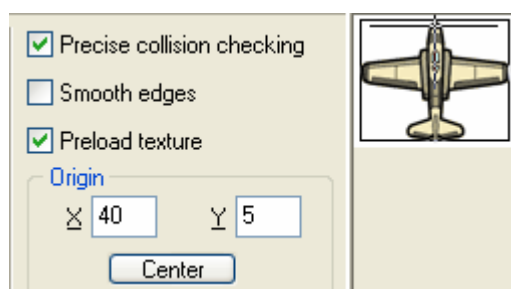
- ☛ Ga weer naar *obj\_speler*.
- ☛ Voeg event *keyboard* > *space* toe (space=spatiebalk).
- ☛ Gebruik dan de action  *create* die je een object in het scherm laat toevoegen. Je krijgt dan dit scherm te zien:
- ☛ Kies voor object *obj\_kogel\_speler*.



Voor x en y mag je 0 laten staan maar dan wel *relative* aanvinken. Hiermee zeg je dus dat Game Maker het object moet aanmaken op dezelfde x- en y-waarde als het vliegtuig. Maar welke zijn dat nou precies? Want het vliegtuigje is hoger en breder dan 1 pixel.



Hier krijg je te maken met de **origin** van sprites. De origin van een sprite is die ene pixel waar Game Maker begint met het opbouwen van je sprite. Het opbouwen gaat zo snel dat je dat niet met het blote oog kunt zien.



Maar Game Maker moet wel weten waar het moet beginnen. Kijk ook nog maar een keer naar het diagram van het vliegtuig eerder in deze opdracht, waarin de x- en y-waarden uitgelegd zijn. De origin van een sprite kan je aanwijzen door weer naar de sprite

## Hoofdstuk 6.1

van het vliegtuigje te gaan. Verander dan de x- en de y-waarde zoals op het plaatje hierboven.



Dit wordt nu dus 1 punt en daar word je kogel gemaakt.

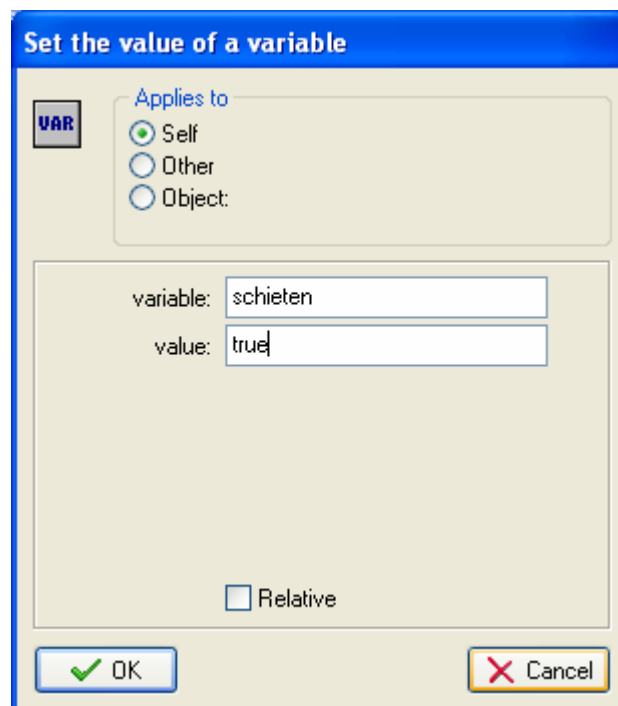
- Doe nu hetzelfde bij de *sprt\_kogel* van de speler. Dan wordt ook deze mooi in het midden gezet.

Je kan center gebruiken, maar als het plaatje niet in het midden zit, zullen de x en y waarde ook niet het midden van de sprite geven. In dat geval moet je even de x en y waarde proberen tot ze goed uitkomen.

Het kan nu zijn dat de kogels vanuit het midden van het vliegtuigje komen en niet vanuit de voorkant van het vliegtuig. Dit zal er niet mooi uitzien en dan moet je de waarde van de y-origin van het vliegtuigje veranderen. Deze moet je waarschijnlijk kleiner zetten zodat deze op de top van het vliegtuigje komt te zitten.

Als je nu het spel speelt kan je vliegtuigje schieten, maar als je de spatiebalk ingedrukt houdt, komt er een onophoudelijke stroom van kogels. Om dit tegen te gaan moet je met *alarms* werken. Hiermee kan je ervoor zorgen je steeds een tijdje moet wachten voor je weer kan schieten. Dit gaat als volgt te werk.

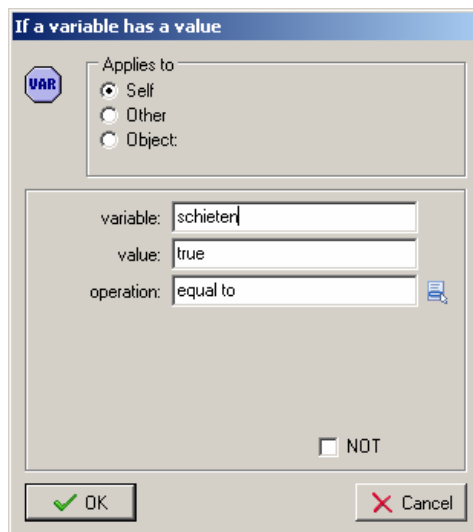
- Add event *create* bij *obj\_speler*.
- Kies action , te vinden onder het tabblad **control**.
- Noem deze  schieten en geef hem de waarde "true" (dus het is waar dat je mag schieten).
- Add event *keyboard→space*






Daar moet je zeggen dat als de variabele schieten "true" is, je kan schieten en dat er dan een alarm gaat lopen en schieten "false" wordt.






## Hoofdstuk 6.1



- Kies action .
- Kies variable 'schieten'
- Kies value 'true'
- Kies operation 'equal to' (engels voor gelijk aan)

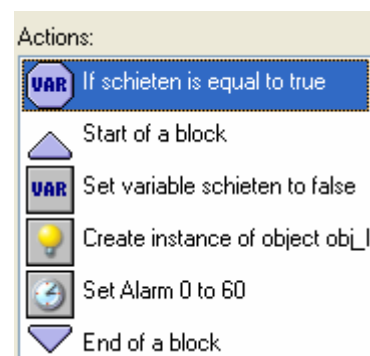
Als dat zo is komen er meerdere dingen die je moet doen dus moet je een die acties afbakenen met een blok  .

- Kies na de  variable,  en dan .

- Tussen die 2 moet je de volgende acties zetten:
-  je moet schieten naar false zetten (de speler mag niet meer schieten)
-  je moet de kogel toevoegen. Dit is de actie die er al stond.
-  je moet een alarm aangeven, die als hij afloopt ervoor zorgt dat je weer kan schieten.

Bij "number of steps" moet je zeggen hoelang het duurt voor het alarm afloopt. Steps is hier een tijdsaanduiding. Het aantal steps per seconde kun je vinden bij de settings van de room. Standaard duren 30 steps 1 seconde. 15 steps een halve sec etc.

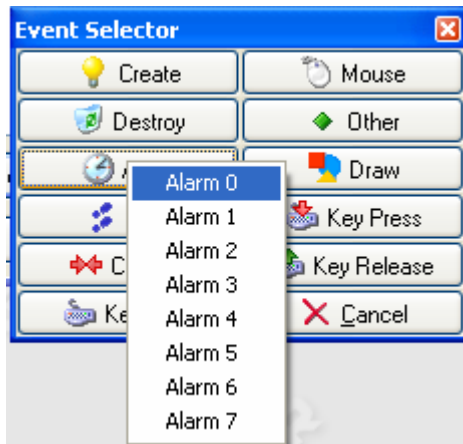
- Een halve seconde wachten voor je kan schieten is wel goed, dus vul maar 15 steps in.
- "In alarm no." is welk alarm je gaat gebruiken. Je hebben nog geen alarm gebruikt dus kunnen je gewoon "alarm 0" gebruiken.




De acties zouden er zo uit moeten zien als hierboven. Alleen moet alarm 0 15 stappen zijn en geen 60.



## Hoofdstuk 6.1



Nu moeten je nog zeggen dat als het alarm afgaat hij de variabele schieten naar "true" zet. Hier gebruiken je de event alarm voor.

- ☛ Selecteer alarm 0, want dat is het alarm dat je in had gesteld.
- ☛ Kies action . Variabele is dus schieten en value is true.




Sla je spel op.

- ☛ Kijk of het werkt.

Werkt het? Mooi. Het is wel een beetje saai zo, of niet? Daarom ga je nu de vijanden maken.

Heb je al een object voor de vijand? Zo niet, maak een object aan voor de vijand. Noem hem maar `obj_vijand`.

Wat wil je van deze vijand? Hij moet in ieder geval van boven naar beneden vliegen.

- ☛ Dus zet bij het create event de actie  neer.
- ☛ Klik het pijltje naar beneden aan.

Wil je dat elke vijand met dezelfde snelheid naar beneden gaat? Nee. Dat is niet zo leuk.

- ☛ Zet daarom de volgende code neer bij de snelheid: `random(3)+2`.

Met deze code, zeg je dat hij als snelheid een willekeurig getal van 0-3 moet nemen (inclusief kommagetallen) omdat de waarde nu 0 kan zijn heb ik er +2 bijgezet. Nu heeft de vijand een minimale snelheid van 2 en een maximale snelheid van  $3+2=5$ .

- ☛ Voor het maken van vijanden maak je een apart object.
- ☛ Noem dit object `obj_controller`. Een sprite is niet nodig, omdat je dit object toch niet wilt zien.
- ☛ In zijn create-event zet je .

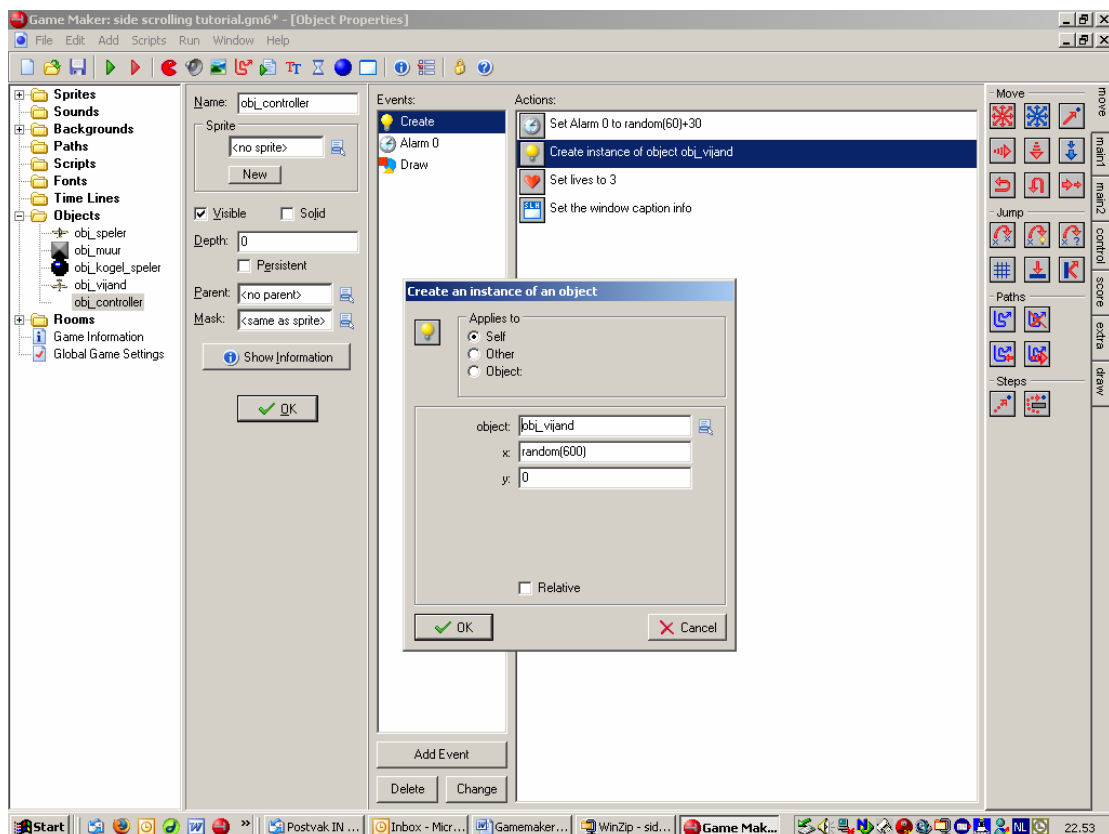
## Hoofdstuk 6.1

☛ "object" wordt *obj\_vijand*.

Voor de y wil je gewoon bovenaan het scherm, dus 0 is goed. Voor de x wil je echter dat hij over de hele breedte van het veld kan komen. Hier kunnen je weer de *random* functie gebruiken.

☛ Kies dus  $\text{random}(640)$  = is het aantal pixels in de breedte van je speelveld.

Maar nu kan het vliegtuig van de vijand ook op het randje van het veld komen. Dit is niet mooi. Als de origin op 0,0 (x=0 en y=0) staat kan je dit verhelpen door  $\text{random}(640 - \text{breedte van de vijand})$  te doen. Bij mij is de breedte 40 dus krijg ik  $\text{random}(640 - 40) = \text{random}(600)$ . Nu komt het vliegtuigje ergens willekeurig van 0 (links in het scherm) tot 600 rechts in het scherm.



Ook moet er na en poosje weer een nieuwe vijand gemaakt worden, dus moet je ook een alarm plaatsen na het maken van de vijand. Leuk is als ook dit een willekeurig tijdje duurt.

☛ Zet voor het aantal steps dan  $\text{random}(60) + 30$ , dan duurt het tussen de 1 en 3 secondes. Immers 30 steps is 1 seconde.


## Hoofdstuk 6.1

- In "alarm no" kun je weer alarm 0 gebruiken omdat je in een ander object zit. Dan hinderen de 2 alarms elkaar niet.

Als het alarm afgaat moet hetzelfde gebeuren als in het creation-event,

- Plak en kopieer deze twee events.
- Vergeet niet *obj\_controller* in je room te zetten.

Nu moet je nog programmeren dat als je kogel een vijand raakt, deze kapot gaat.

- Open *obj\_vijand*.
- Maak een collision event met het object van je kogels.
- Kies als action dat beide dingen kapot gaan met . Je kan dan kiezen voor self, other en object.
- Je moet 1x self kiezen (dan gaat de vijand kapot) en 1x other (dan gaat de kogel kapot)

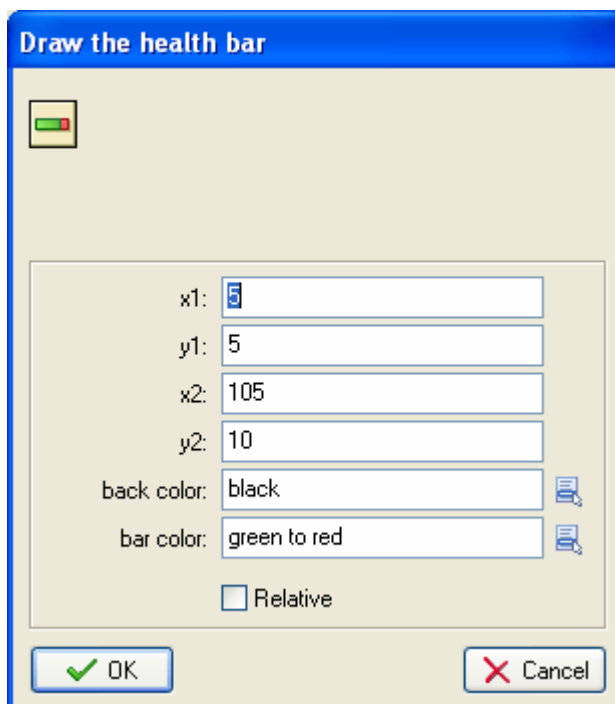
Als laatste wil je nog punten geven.

- Kies action .
- Als score zet je dan bijvoorbeeld 100.
- Vergeet niet relative aan te vinken, anders komt er niks bij.

Nu nog maken dat de vijanden ook jou kunnen raken. Voor je dit kunt doen, moet je een **healtbar** tekenen. Een healtbar neemt langzaam in waarde af als je bijvoorbeeld geraakt wordt door een kogel.

- Add event  Draw aan *obj\_controler*
- Zet in de draw event van *obj\_controler* de volgende action .

Hiermee teken je de healtbar. Dit moet in de draw event van een object. Dit object hoeft



Draw the health bar

x1: 5  
y1: 5  
x2: 105  
y2: 10  
back color: black  
bar color: green to red  
☐ Relative




OK Cancel

## Hoofdstuk 6.1


geen sprite te hebben, omdat als een object een draw event heeft, hij de sprite toch niet tekent.

- Vul de waardes in zoals op het plaatje.


Bij de create-event van *obj\_speler* kun je de waarde aangeven van je health.

- Klik op de  create event van *obj\_speler*.
- Kies action .
- Maak van de health 100.
- Maak nu een collision event met *obj\_speler* aan in *obj\_vijand*.
- Zet hier als actie  en dan -20 als score en relatief aanvinken. (dus als je geraakt wordt gaat er 20 van je leven af).

Ook willen je dat de vijand kapot gaat, dus voegen je er een destroy-actie bij.

- Kies action .
- Vink nu *other* aan, want je zit bij de speler.



Nog een laatste action voor de healthbar goed werkt. Je wil dat als de health gelijk of kleiner dan 0 is, de speler een leven verliest.

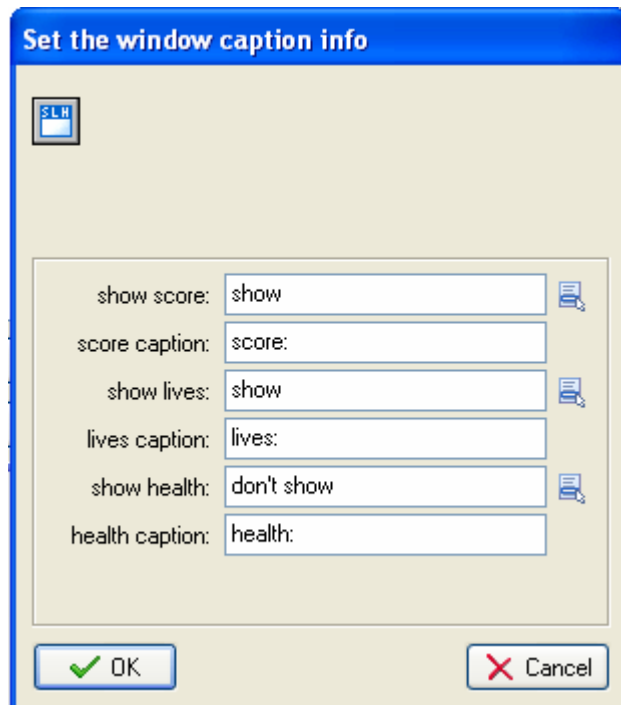
- Voeg daarom aan de *collision with obj\_speler* de action  if *health is a value* toe.
- Maak van de "value" 1 en van de operation "smaller then".

Nu zeg je dat als de health kleiner dan 1 is (gelijk of kleiner dan 0 dus) hij iets moet doen.


- Kies action  *destroy* en dan *self*.


Nu wil je nog dat er een leven van afgaat. Daarvoor moet je eerst zeggen hoeveel levens je hebben.

- Klik op  create-event van *obj\_controller*
- Kies  als action.





- Kies als aantal 3.

Je moet nog even aangeven dat Gamemaker het aantal levens bovenin de balk van het spel moet laten zien. Dit doe je met , te vinden onder score.

- Kies action .
- Vul de waarden in, zoals in het plaatje.



Dan zetten je bij het *destroy* event van *obj\_speler* dat er een leven afgaat. Deze moet je nog wel even maken.

- Klik op *obj\_speler*.
- Voeg het event  *Destroy* toe.
- Kies action  *set lives*.
- Kies dan als waarde -1 invullen en vink *relative* aan.

Als het aantal levens 0 is, moet je de highscore tabel laten zien en daarna het spel laten afsluiten.

- Zet na ,  om te kijken hoeveel levens er zijn.

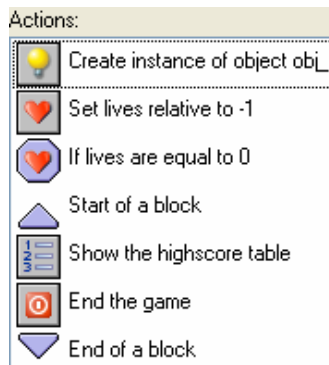
Als waarde 0 en operation "equals" (dus als het aantal levens gelijk aan 0 is) dan wil je weer meerdere dingen doen (namelijk highscore tabel laten zien en het spel afsluiten) dus gebruik je weer een blok.


- Gebruik  dan .
- Daar tussen in zet je  om de highscore tabel te laten zien (de waardes zijn standaard goed, dus je kan gewoon op ok drukken)
- Kies daarna  om het spel af te sluiten.



Als laatste moet je weer een nieuw object van de speler maken.

## Hoofdstuk 6.1



- Gebruik  als object dus *obj\_speler*.
- Hij moet in het midden van de kamer komen. Dit doe je door `room_width/2` in te vullen bij `x` (hiermee neemt hij de breedte van de kamer en deelt dat door 2, zo kom je mooi in het midden).
- Als `y` moet je 400 nemen.

De acties zouden er dan zoals in het plaatje uit moeten zien.

Nu is het tijd om een room te maken.

- Add background.
- Kies een mooie achtergrond voor je spel.
- Add room.
- Kies de juiste background.



Sla je spel op.



- Maak een executable.



Zo, hiermee heb je je eigen scrolling shooter. Ik hoop dat het allemaal gelukt is.

## Hoofdstuk 6.1

### Vragen hoofdstuk 6.1

1. Wat is de *origin* van een sprite?
2. Welke kant ga je op als je vanuit het 0-punt in de richting van  $y=-4$  en  $x=4$  beweegt?
  - A. Links boven
  - B. Links onder
  - C. Rechts boven
  - D. Rechts onder
3. Wat betekend *random*?
4. Wat betekend deze  action?
5. Wat doet deze  action?